Received June 1, 2022, accepted June 14, 2022, date of publication June 17, 2022, date of current version June 24, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3183902

Chaos Elite Harris Hawk Optimization Algorithm to Solve Chemical Dynamic Optimization Problems

LILA HONG[1], YUANBIN MO[1,2], DONGXUE BAO[1], AND RONG GONG[1]

[1]Institute of Artificial Intelligence, Guangxi Minzu University, Nanning 530006, China
[2]Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis, Guangxi Minzu University, Nanning 530006, China

Corresponding author: Yuanbin Mo (moyuanbin2020@gxun.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 21466008, in part by the Guangxi Natural Science Foundation under Grant 2019GXNSFAA185017, and in part by the School Level Scientific Research Projects under Grant 2021M DKJ004.

ABSTRACT Dynamic optimization problems in chemical engineering are often described by complex differential equations, In general, Due to the complexity of the model established in practice, it is not easy to solve it with an accurate algorithm, so the study of numerical methods to solve such problems has received much attention. As a new computing model, intelligent optimization algorithm has attracted more attention in solving dynamic optimization problems because of their easy operation. Based on the analysis of the Harris Hawk Optimization algorithm, this paper proposes the Chaos Elite Harris Hawk Optimization algorithm (CEHHO), which is used to improve the performance of CEHHO using control vector parameterization to solve dynamic optimization problems of the chemical industry. First of all, when the population is initialized, the population is initialized by Opposition-based learning Logistic chaos, which improves the diversity of the population and the quality of the solution. Second, the linear decreasing escape energy factor is changed to a nonlinear decreasing escape energy factor to balance the exploration and exploitation capabilities of the algorithm. Finally, through the mutation strategy guided by elite individuals, the algorithm can jump out of the local optimum. We use 8 test functions and 5 classical chemical optimization problems to evaluate the feasibility of the algorithm and compare and analyze the research results with other solving methods, showing the superiority of the CEHHO algorithm.

INDEX TERMS Chaos Elite Harris Hawk, opposition-based learning, elite individual, mutation strategy, chemical optimization, control vector parameterization.

I. INTRODUCTION

The goal of chemical dynamic optimization is to optimize predefined performance indicators, such as profitability or productivity, and optimize control variables to make performance indicators optimal, meet safety or environmental specifications [1], improve chemical production efficiency, and reduce energy consumption. The dynamic optimization control of the chemical industry has received extensive attention from domestic and foreign academic circles, and scholars have done a lot of research on the development of dynamic optimization problems. Generally speaking, these methods basically consist of two parts, control discrete and parameter optimization [2]. Control discretization transforms an infinite-dimensional problem into a finite-dimensional nonlinear dynamic programming problem (NLP). Parameter optimization methods are mainly divided into three categories: (1) dynamic programming; (2) mathematical programming algorithms; (3) intelligent optimization algorithms.

Dynamic programming discretizes both time and control variables, Dadebo S A and McAuley K B added a motion suppression criterion to the dynamic optimization algorithm to punish excessive control motion, and proposed a dynamic optimization for chemical engineering problems based on dynamic programming [3], Bayón *et al.* based on Pontryagin's Minimum principle, obtained the initial guess of the solution, and put forward the initial guess of the

The associate editor coordinating the review of this manuscript and approving it for publication was Ioannis Schizas.

dynamic optimization solution of the chemical process [4], However, when the dimensions of the control variables and state variables are large, the calculation process becomes very cumbersome. To solve this shortcoming, an improved dynamic optimization method, iterative dynamic programming (IDP), is proposed. Sundaralingam R proposed a two-step method for solving dynamic optimization problems with state inequality constraints by iterative dynamic programming [5]. Although IDP reduces the computational time caused by the increase of dimensions, IDP is time-consuming and inefficient [6], especially It is more prominent when solving dynamic optimization problems of chemical industry.

Mathematical programming algorithms include Newton's method, sequential quadratic method, control variable parameterization (CVP), etc. [7], Huang and Luo used the control vector parameterization method to transform the dynamic optimization problem into a mixed-integer nonlinear programming problem, and used it to deal with the fluctuating ethylene tower [8], Huang *et al.* proposed a dynamic optimization algorithm based on control vector parameterization and state transition algorithm [9], This method has a strong dependence on the initial value and needs gradient information. However, many differential equations, in reality, are non-continuous and non-derivable. For these problems, this method is no longer applicable.

The solution of the control variable parameterization method is too dependent on the initial value, and the setting of the initial trajectory value has a strong sensitivity to the solution accuracy and affects the convergence speed of the algorithm. To solve the above problems, a strategy of combining an intelligent optimization algorithm with the control variable parameterization method is proposed. The intelligent optimization algorithm has few parameters, simple principles, easy implementation, and does not require the problem to be solved to be derivable and continuous, so it is widely used to solve optimization problems in various disciplines of engineering [10]. Zhang proposed a dynamic optimization of chemical processes based on a modified sailfish optimizer combined with an equal division method [11], Mattia Vallerio *et al.* discusses a computationally efficient robust dynamic optimization approach based on the Sigma Point method [12], Angira and Santosh proposed a trigonometric differential evolution method for solving dynamical system optimization [13], Fan proposed a hybrid improved genetic algorithm and its application in the dynamic optimization problem of a chemical process [14], Chen *et al.* proposed a sorting-based differential evolution algorithm to solve the chemical dynamic optimization problem to improve the performance of the differential evolution algorithm using the control vector parameterization [15]. Chen *et al.* proposed a chemical dynamic system optimization based on quadratic interpolation for teaching and learning, and introduced diversity-enhancing teaching strategies and quadratic interpolation in basic teaching and learning to enhance the exploration ability of the algorithm [16], Fan *et al.* proposed a control vector parameterization method based on a differential evolution algorithm and its application in dynamic optimization of a chemical process [17]. The above-improved algorithm can improve the accuracy and convergence speed of chemical dynamic optimization to a certain extent. In this paper, a chaotic elite Harris hawk optimization algorithm is proposed to solve the chemical dynamic optimization problem. Firstly, the infinite-dimensional problem is transformed into a finite-dimensional problem by using the control variable parameterization method. Then use the improved algorithm to optimize the problem.

Although the method proposed above can improve the accuracy and convergence speed of solving chemical dynamic optimization problems to a certain extent, it can be seen from the literature that the trajectory of the solved control variables has certain fluctuations, which cannot better simulate the changing process of chemical problems. Harris Harks optimization algorithm is a new intelligent algorithm with good solving ability. Based on the NFL (No Free Lunch) principle, this paper firstly initializes the population through logistic reverse learning, dynamically adjusts the global and local search, and better balances the relationship between exploration and exploitation. Secondly, using the elite individual guidance strategy and the piecewise constant method to optimize the dynamic optimization problem of the chemical industry, the trajectory of the control variable tends to be smooth, Optimal performance index. In addition, recent optimization algorithms [18]–[22] provide other ideas for solving dynamic optimization problems in chemical engineering.

The rest of this paper is organized as follows: Section 2 introduces the chemical dynamic optimization problem, section 3 describes the Harris Hawks optimization algorithm, section 4 describes the improved Harris Hawks optimization algorithm, and section 5 tests the performance of the improved algorithm through five chemical dynamic optimization problems. Finally, section 6 summarizes the work done.

II. PROBLEM DESCRIPTION

A. DYNAMIC OPTIMIZATION PROBLEM

The solution of the function extreme value problem is a static problem. Compared with the static problem, Real-world optimization problems are mostly time or space-dependent. This kind of problem is called dynamic optimization problem. It is usually described by complex differential equations. Dynamic optimization is to optimize a certain performance index by controlling operating variables [23]. Such as profitability, product quality, productivity, etc. [24]. Typical dynamic optimization problems are usually described by the following mathematical models.

$$\min J(u(t)) = \varphi(x(t_0, t_f)) + \int_{t_0}^{t_f} \phi(t, x(t), u(t))dt$$

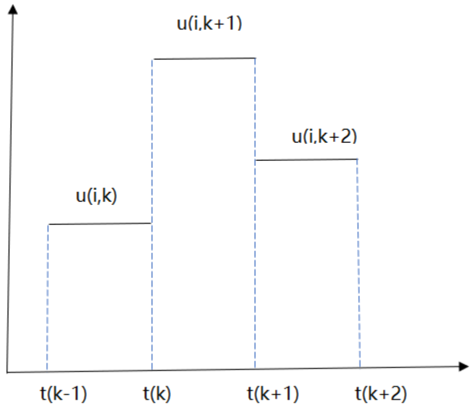


FIGURE 1. Piecewise constant method.

$$s.t \begin{cases} x(t) = f(t, x(t), u(t)) \\ x(t_0) = x_0 \\ u^l \leq u(t) \leq u^u \\ t \in [t_0, t_f] \end{cases} \tag{1}$$

In the formula, J is the performance index, namely the objective function, The solution to this problem is to find a control trajectory in the time interval $[t_0, t_f]$ to minimize the performance index. x and u represent state variables and control variables, respectively. $x(t_0)$ and $x(t_f)$ represent the initial and final state values, respectively. f, φ Represents the constraints of the system on differential and algebraic equations. u^l, u^u represents the lower and upper bounds of the control variable.

B. CONTROL VECTOR PARAMETERIZATION

In order to better approximate the control trajectory, the control vector parameterization transforms the control vector from an infinite dimension into a finite dimension through discretization, and each segment passes through the basis functions (piecewise constant [25], [26], [27], piecewise linear functions [28], [29], piecewise quadratic polynomial functions [30], etc.).

The most common way to parameterize the control vector is piecewise constant, as shown in Figure 1. Divide the control time domain $[t_0, t_f]$ into a finite number of subintervals, $[t_{i-1}, t_i](i = 1, 2, \ldots, N)$. as shown in Figure 2. The ith component *u(ti)* of the control variable $u(t)$ in the control time domain can be represented by an approximation of the ith subinterval [31].

$$u_i(t) \approx \sum_{j=1}^{N} u_j^i(t)\tau[t_{i-1}, t_i] \quad \text{i} = 1, 2, \ldots, \text{N} \tag{2}$$

The CVP method flow is as follows:

(1) Divide the control time domain into several finite subintervals from infinite dimensions.

(2) Use piecewise constants to approximate the control variables in each subinterval.

(3) Optimize the parameters by using the chaotic elite Harris hawk optimization algorithm.

(4) When the conditions are met or the number of optimizations is reached, the optimal solution is output.

III. HARRIS HAWK OPTIMIZATION ALGORITHM

Harris hawk Optimization (HHO) is a population-based, un-gradient meta-heuristic optimization algorithm [32]. By simulating the behavior of a Harris hawk to catch its prey (rabbit). Harris hawk uses the "seven kills" strategy to capture rabbits and changes different strategies to besiege according to different escape methods. In the process of attacking, there is a certain relationship between the rabbit's escape ability and energy. Therefore, the escape energy factor is used to express the escape ability of rabbits. The algorithm has a strong local search ability and requires fewer parameters to adjust. High search accuracy and easy operation. HHO is mainly composed of two parts: exploration and development. The detailed process is as follows:

A. EXPLORATION PHASE

In the HHO algorithm, each Harris hawk individual is a candidate solution, and they all have a chance to approach the target value. During the exploration phase, all Harris hawks will randomly perch on electric poles or large trees to supervise and observe the prey. Monitoring is carried out in two equal-probability ways. The mathematical model is as follows:

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1|X_{rand}(t) - 2r_2X(t)| & \text{q} \geq 0.5 \\ X_{rabbit}(t) - X_m(t) - r_3(lb + r_4(ub - lb)) & \text{q} < 0.5 \end{cases} \tag{3}$$

$$X_m(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t) \tag{4}$$

Among them, $X(t)$ represents the current iteration individual, $X(t+1)$ represents the next iteration individual, $X_{rand}(t)$ represents an individual randomly selected from the population, and $X_{rabbit}(t)$ represents the position of the prey, that is, the position of the rabbit, $X_m(t)$ represents the average value of the current population position, ub, lb represent the upper and lower boundary values of the eagle position, $q, r_1, r_2, r_3, r_4 \in (0, 1)$, and N represents the number of populations.

B. TRANSITION FROM EXPLORATION TO EXPLOITATION

Harris hawk has to adopt different strategies to besiege and capture rabbits according to their escape methods. This complex and changeable escape strategy is determined by the escape energy factor. Therefore, the escape energy factor determines whether the HHO algorithm performs a global search or a local search.

$$E = 2E_0(1 - \frac{t}{T}) \tag{5}$$

$$E_0 = 2r_5 - 1 \tag{6}$$

In the formula, E is the escape energy factor. $E0$ represents the initial energy of the prey, the value range is $(-1,1)$. t is the current iteration number, T is the maximum number of iterations. $r_5 \in (0, 1)$.

C. EXPLOITATION PHASE

In the exploration stage, Harris hawks are distributed in space, observing and monitoring the behavior of prey, waiting for opportunities to catch prey. Harris hawk attacking prey is carried out in the development stage, and the prey is attacked and captured by adopting four methods. When $|E| \geq 1$, the algorithm is in the exploration phase, when $|E| < 1$, the algorithm is in the development phase, A random number r to represent the probability of the prey escaping. $r \in (0, 1)$.

1) USING SOFT SIEGE TO SIEGE THE PREY

When $r \geq 0.5\&\&|E| \geq 0.5$, The rabbit had enough energy to escape and tried to escape by jumping, but it failed in the end. In the process of escaping the rabbit, the eagle gently surrounded it, exhausting the rabbit, and then caught it, the behavior The simulation is carried out by the following mathematical model.

$$\mathrm{X}(t+1) = \Delta X(t) - E|JX_{rabbit}(t) - X(t)| \tag{7}$$

$$\Delta X(t) = X_{rabbit}(t) - X(t) \tag{8}$$

$$J = 2(1 - r_6) \tag{9}$$

In the formula. $r_6 \in (0, 1)$, $\Delta X(t)$ represents the position difference between the current position of the rabbit and the eagle. J represents the jumping intensity of the rabbit, and the escape behavior of the rabbit is simulated by different values.

2) USING HARD SIEGE TO SIEGE THE PREY

When $r \geq 0.5\&\&|E| < 0.5$, At this time, the rabbit does not have enough energy to escape and is in exhaustion, so the eagle directly captures the rabbit. Simulation is performed by formula (11).

$$X(t+1) = X_{rabbit}(t) - E|\Delta X(t)| \tag{10}$$

3) PROGRESSIVE DIVE SOFT SIEGE

When $r < 0.5\&\&|E| \geq 0.5$, Rabbits have the energy to run away, but eagles will adopt a more nimble strategy than before. First, the concept of levy flight was introduced to imitate the deceptive movements of the rabbit before escaping. The eagle would make several irregulars, fast dives around the rabbit in an attempt to correct its position and direction. Second, a greedy selection mechanism is added at this stage to compare the previous motion with the results of the levy flight motion and choose the better one. The mathematical model is as follows:

$$Y = X_{rabbit}(t) - E|JX_{rabbit}(t) - X(t)| \tag{11}$$

$$Z = Y + S \cdot LF(D) \tag{12}$$

where D is the dimension to solve the problem, S is a random vector of $1 \cdot D$, LF is the levy flight function.

$$LF(x) = 0.01\frac{\mu\sigma}{|\nu|^{\frac{1}{\beta}}}, \sigma = (\frac{\Gamma(1+\beta)\sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2})\beta \cdot 2^{(\frac{\beta-1}{2})}})^{\frac{1}{\beta}} \tag{13}$$

in the formula. $\mu, \nu \in (0, 1)$, β is a constant of 1.5.

$$X(t+1) = \begin{cases} Y & \text{if F(Y)<F(X(t))} \\ Z & \text{if F(Z)<F(X(t))} \end{cases} \tag{14}$$

Y, Z are calculated by equations (12) and (13).

4) HARD SIEGE TO SIEGE OF PREY WITH PROGRESSIVE RAPID DIVING

When $r < 0.5\&\&|E| < 0.5$, Rabbits don't have enough energy to escape. The hawk has formed a circle around the rabbit. the siege is similar to the progressive dive soft siege, During this process, by calculating the average position of the rabbit and the eagle, it is better to evaluate the best position from the rabbit. The model is as follows:

$$Y = X_{rabbit}(t) - E|JX_{rabbit}(t) - X_m(t)| \tag{15}$$

$$Z = Y + S \cdot LF(D) \tag{16}$$

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

The pseudocode of the HHO algorithm is shown in Algorithm 1.

IV. CHAOS ELITE HARRIS HAWK OPTIMIZATION ALGORITHM

HHO algorithm is a competitive meta-heuristic optimization algorithm, which shows good flexibility and robustness in solving optimization problems, so it has received extensive attention. Based on the free lunch theorem, the HHO algorithm, like all algorithms, is not suitable for all optimization problems. For complex problems, it also has slow convergence speed, low solution accuracy, and is easy to fall into local optimum. The main reason is. First, the HHO algorithm adopts random initialization, resulting in low diversity of the population.

Secondly, when the HHO algorithm transitions from the exploration stage to the exploitation stage, the escape energy factor decreases linearly. This method can find the solution of the problem for solving simple problems, but for complex problems, the algorithm cannot solve the global optimal value. Finally, in the exploitation phase of hard siege and soft siege strategies, the individuals of the population are not fully communicated.

A. OPPOSITION-BASED LEARNING LOGISTIC CHAOS INITIALIZATION POPULATION

For the meta-heuristic optimization algorithm, the distribution of the initial solution of the population in the solution space has a great influence on the algorithm, affecting the solution accuracy and convergence speed of the algorithm.

Algorithm 1 : Pseudo-Codes of the HHO

```
Inputs: The Population size $N$, maximum number of iterations $T$.
Outputs: Solution of the problem
While $(t < T)$
Calculate the fitness values, Find the best individual(Rabbit position)
Check whether it is out of bounds
for each hawk($X_i$)
    Update the escape energy factor with equation (5)
    Update the initial energy of prey with equation (6)
    Calculate the jumping degree of the rabbit with equation (9)
    if $(|E| \geq 1)$
        Use equation (3) to calculate the current hawks position
    End if
    else if (|E| <1)
        if $(r \geq 0.5\&\&|E| \geq 0.5)$
            Use equation (7) to update the position of the hawks
        End if
        if $r \geq 0.5\&\&|E| < 0.5$
            Use equation (10) to update the position of the hawks
        End if
        if $r < 0.5\&\&|E| \geq 0.5$
            Use equation (11)-(14) to update the position of the hawks
        End if
        if $(r < 0.5\&\&|E| < 0.5)$
            Use equation (17)-(19) to update the position of the hawks
        End if
    End if
end for
t=t+1
end while
```

A population with a uniform distribution is beneficial to improve the solution performance of the algorithm. However, the standard HHO algorithm is initialized in a random way, which is not evenly distributed.

Chaos is a bounded dynamic behavior that occurs in deterministic nonlinear systems, with ergodicity, randomness, and sensitivity. Chaos theory is usually described as the ''butterfly effect'' [33]. In recent years, in the meta-heuristic algorithm, the application of chaos theory to the initialization population has become a hot spot. Chaos sequence replaces random sequence to enhance population diversity, which is called chaotic optimization algorithm. Due to the non-repetitive characteristics of chaos, a chaotic optimization algorithm is faster than a random search [34]. Commonly used chaotic maps include a logistic map, tent map, Chebyshev map, Gauss map, Circle map, and Sine map [35]. Compared with other chaotic maps, a logistic map has the advantages of better ergodicity and convergence. Therefore, this paper uses the logistic map to initialize the population. The logistic formula is shown in (21).

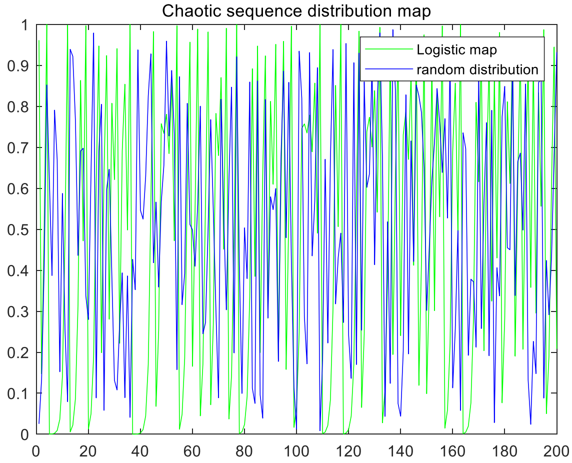


FIGURE 2. Comparison of chaotic sequence distribution.

$$z_{n+1} = \lambda z_n(1 - z_n) \tag{18}$$

where $\chi(\chi \in (0, 4))$ is the logistic control parameter. Figure 2 shows the distribution of logistic chaos and random distribution sequences with $\lambda = 4$ and 200 iterations. $z_n \in (0, 1)\&\&z_n \neq 0, z_n \neq 0.5, z_n \neq 0.7, z_n \neq 1$.

Opposition-based learning (OBL) strategies can be used to improve the search efficiency of metaheuristics [36]. The probability that OBLs find a solution is related to the opposite solution they find in the opposite region of the solution space, compared to random directions, adversarial learning provides a higher chance of finding unknown optimal solutions to problems in a random direction, and they are closer to the global optimum than random solutions. The strategy of simultaneously examining candidate solutions and their inverse solutions is adopted to speed up the convergence of the algorithm to the global optimal solution [37], [38], [39]. On this basis, this paper proposes the logistic chaotic reverse learning. Based on the logistic chaotic initialization population, adding the opposition-based learning mechanism can not only enhance the diversity of the population but also help to get closer to the solution of the problem faster. Speed up the convergence of the algorithm. opposition-based learning is defined as follows:

Let $x(x \in [lb, ub])$ be a number in one-dimensional space, Then its opposite position x' can be expressed by the following formula:

$$x' = lu + ub - x \tag{19}$$

Generalizing from one-dimensional space to n-dimensional space, let $x_i \in [lb_i, ub_i], \forall i \in 1, 2, \cdots, n$, Then its opposite position is expressed as follows:

$$x_i^0 = lb_i + ub_i - x_i \tag{20}$$

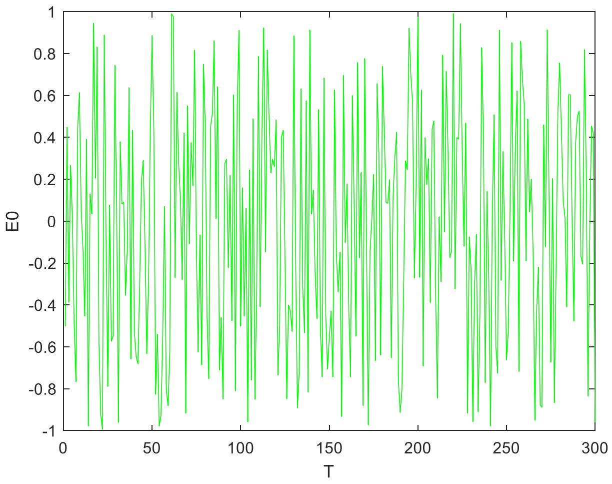


FIGURE 3. E_0 **iteration diagram.**

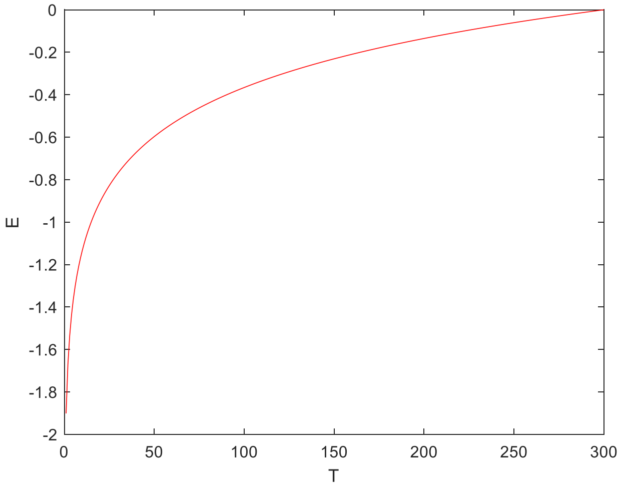


FIGURE 4. **E iteration diagram.**

TABLE 1. **Algorithm parameters.**

Algorithm	Parameter settings
HHO	T=1000， popsize=30, β =1.5
PSO	T=1000， popsize=30,w=0.99,$c1$=2,$c2$=2
GA	T=1000， popsize=30,pc=0.6,pm=0.2
WOA	T=1000， popsize=30,b=1
SSA	T=1000， popsize=30,ST=0.6， PD=0.7， SD=0.2
MFO	T=1000， popsize=30,b=1
CEHHO	T=1000， popsize=30, β =1.5

B. DYNAMICALLY ADJUST THE EXPLORATION AND EXPLOITATION PHASE

The HHO algorithm uses a linear decreasing method to change the escape energy factor. In the initial stage of iteration, the population only conducts a global search. When an individual falls into a local optimum, the strategy in the exploration stage is not easy to make the population jump out of the local optimum. Therefore, this paper proposes a nonlinear decreasing method to explore the dynamic adjustment algorithm. and the development stage,

In the exploration stage, the population can have the opportunity to use the local exploration strategy to search, and the current optimal position can be searched more accurately, which is conducive to speeding up the convergence speed of the algorithm and making the algorithm more harmoniously combined the exploration stage and the development stage. The proposed nonlinear decreasing escape energy factor update formula is as follows:

$$E = \ln(\frac{t}{T})^{\frac{1}{3}} \tag{21}$$

C. ELITE MUTATION STRATEGY

In the exploration stage, the HHO algorithm only uses the optimal position of the population and does not fully utilize the individual information resources of the population, which makes the population too close to the current optimal solution. When solving complex problems, the HHO algorithm is easy to fall into local optimality. Therefore, in the hard siege and soft siege strategies in the development stage, according to the average position among the population individuals, this paper proposes a new update strategy guided by elite individuals. In this strategy, the population can not only use the information of the current optimal individual, and fully excavate and use all individual information, Finally, through the comparison of fitness values, if the fitness value of the new update strategy is better, keep it, otherwise keep the previous position, which can make full use of the individual's position information and guide other individuals to move closer to the optimal value more accurately move closer. The newly proposed update formula is as follows:

$$X(t+1) = X_{rabbit}(t) + r_7 * (X_m(t) - X(t)) \tag{22}$$

where r_7 is a random number of (0, 1).

D. CEHHO ALGORITHM FLOW CHART

E. NUMERICAL EXPERIMENT SIMULATION ANALYSIS

To test the performance of the CEHHO algorithm, this section will conduct a numerical experimental analysis. This paper selects five algorithms (HHO, CEHHO, The whale optimization algorithm (WOA) [40], Sparrow Search Algorithm (SSA) [41], Moth-flame optimization algorithm (MFO) [42]) for a comparative study. In the experiment, the population size was uniformly set to 30, the maximum number of iterations T = 1000, and all algorithms were run independently 30 times. In this experiment, the processor is Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz 2.40 GHz, and the experimental software is MATLAB 2018b. In addition, this paper lists the various parameters of different algorithms, as shown in Table 1.

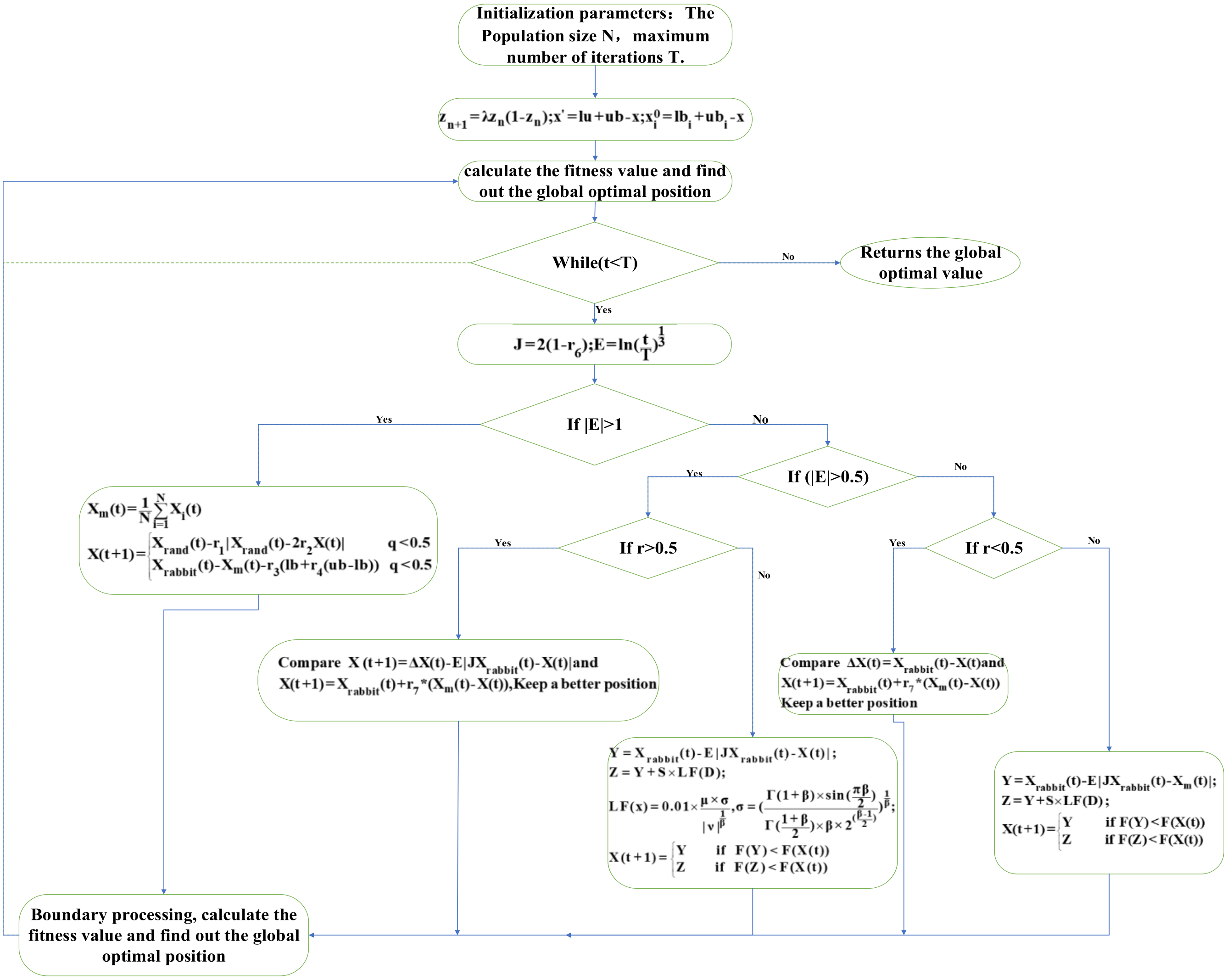

Initialization parameters: The Population size N, maximum number of iterations T.
$z_{n+1}=\lambda z_n(1-z_n);x'=lu+ub-x;x_i^0=lb_i+ub_i-x$
calculate the fitness value and find out the global optimal position
While(t<T)
No
Returns the global optimal value
Yes
$J=2(1-r_6);E=\ln(\frac{t}{T})^{\frac{1}{3}}$
If |E|>1
Yes
No
$X_m(t)=\frac{1}{N}\sum_{i=1}^{N}X_i(t)$
$X(t+1)=\begin{cases}X_{rand}(t)-r_1|X_{rand}(t)-2r_2X(t)| & q<0.5\\ X_{rabbit}(t)-X_m(t)-r_3(lb+r_4(ub-lb)) & q<0.5\end{cases}$
If (|E|>0.5)
Yes
No
If r>0.5
Yes
No
If r<0.5
Yes
No
Compare $X(t+1)=\Delta X(t)-E|JX_{rabbit}(t)-X(t)|$and $X(t+1)=X_{rabbit}(t)+r_7*(X_m(t)-X(t))$,Keep a better position
Compare $\Delta X(t)=X_{rabbit}(t)-X(t)$and $X(t+1)=X_{rabbit}(t)+r_7*(X_m(t)-X(t))$ Keep a better position
$Y=X_{rabbit}(t)-E|JX_{rabbit}(t)-X(t)|;$
$Z=Y+S\times LF(D);$
$LF(x)=0.01\times\frac{\mu\times\sigma}{|v|^{\frac{1}{\beta}}},\sigma=(\frac{\Gamma(1+\beta)\times\sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2})\times\beta\times 2^{(\frac{\beta-1}{2})}})^{\frac{1}{\beta}};$
$X(t+1)=\begin{cases}Y & \text{if } F(Y)<F(X(t))\\ Z & \text{if } F(Z)<F(X(t))\end{cases}$
$Y=X_{rabbit}(t)-E|JX_{rabbit}(t)-X_m(t)|;$
$Z=Y+S\times LF(D);$
$X(t+1)=\begin{cases}Y & \text{if } F(Y)<F(X(t))\\ Z & \text{if } F(Z)<F(X(t))\end{cases}$
Boundary processing, calculate the fitness value and find out the global optimal position

FIGURE 5. CEHHO algorithm flow chart.

1) BENCHMARK FUNCTION

This paper selects eight benchmark functions to test the algorithm, including unimodal functions (F1-F4) and multimodal functions (F4-F8). The unimodal function has only one solution, which is mainly used to test the convergence ability of the algorithm. The multimodal function contains multiple optimal solutions, of which there is only one global optimal solution. For most algorithms, it is easy to fall into the local optimal solution. Therefore, the multimodal function can be better used to evaluate the development ability of the algorithm. Table 2 gives the basic information of these eight functions.

2) NUMERICAL ANALYSIS

In the experiment, the optimal value, worst value, average value, and variance are analyzed. The experimental results are shown in Table 3, and the optimal data is bolded. It can be seen from the experimental results that the CEHHO algorithm proposed in this paper has better solution accuracy than the other four algorithms in optimizing the unimodal function (F1-F4). and is at least 100 orders of magnitude higher. For multimodal functions (F5-F8), when optimizing function F5, HHO, WOA, SSA, and CEHHO algorithms can all solve the theoretical optimal value, and when optimizing function F6, HHO and CEHHO algorithms can solve the optimal value. For functions F7 and F8, although none of the five algorithms can solve the theoretical value, the CEHHO algorithm shows a good solution accuracy compared with the other four algorithms. From the perspective of variance, the CEHHO algorithm shows good stability.

3) CONVERGENCE ANALYSIS

In order to more intuitively observe whether the algorithm can jump out of the local optimum, the convergence speed of the algorithm and other performances in the whole experiment process. Figure 4 lists the convergence diagrams of the 8 test

TABLE 2. Benchmark functions.

Test function	Dim	Range of search space	Fmin
Unimodal benchmark			
$f_1(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	30	[-100,100]	0
$f_2(x)=\sum_{i=1}^{n}(\sum_{j-1}^{i}x_i)^2$	30	[-100,100]	0
$f_3(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	30	[-30，30]	0
$f_4(x)=\sum_{i=1}^{n}([x_i+0.5])^2$	30	[-100,100]	0
Multimodal benchmark			
$f_5(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$	30	[-5.12,5.12]	0
$f_6(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$	30	[-600,600]	0
$f_7(x)=\frac{\pi}{n}\{10\sin^2(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2[1+10\sin^2(\pi y_i+1)]+$ $(y_n-1)^2+\sum_{i=1}^{n}u(x_i,10,100,4)\}$ $u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m & x_i>a\\ 0, & \text{-}a\le x_i\le a\\ k(-x_i-a)^m, & x_i<-a\end{cases}$	30	[-50,50]	0
$f_8(x)=0.1\{\sin^2(3\pi x_1)+\sum_{i=1}^{n}(x_i-1)^2[1+\sin^2(3\pi x_i+1)]+$ $(x_n-1)^2[1+\sin^2(2\pi x_n)]\}+\sum_{i=1}^{n}u(x_i,5,100,4)$	30	[-50,50]	0

functions. For functions F1-F4, the convergence speed and solution accuracy of the CEHHO algorithm is much higher than those of the other four algorithms. When solving the function F3, the CEHHO algorithm is probably iterating to The global optimal solution can be solved 380 times, and the global optimal solution of the function F4 can be solved when the number of iterations is about 100 times. For the multimodal functions F5 and F6, it can be seen from the figure that although the HHO, WOA, SSA, and CEHHO algorithms can solve the global optimum, the convergence speed of CEHHO is faster than other algorithms. It can be seen from the convergence diagrams of F7 and F8 that the CEHHO algorithm can jump out of the local optimum. In general, the CEHHO algorithm proposed in this paper has good advantages in terms of solution accuracy and convergence speed.

4) BOX PLOT ANALYSIS OF TEST RESULTS

Figure 7 draws the boxplots of the five algorithms running independently 30 times. Table 4 shows the Inter Quartile Range (IQR) values of the eight functions required. Combining with the chart, it can be seen that when solving F3, F4, and F7, F8, the IQR value is the smallest, indicating that the result distribution of the CEHHO solution is more concentrated. When solving F1, although the IQR value of CEHHO is not the most concentrated relative to MFO, it can be seen from Figure 5 that the fitness value of MFO is in - 10, the solution accuracy is far inferior to the CEHHO algorithm,

TABLE 3. Experimental data.

Functions	Result	HHO	PSO	GA	WOA	SSA	MFO	CEHHO
F1	Best	3.75E-109	5.36E-02	2.23E-06	8.19E-113	-5.65E-38	-1.00E+01	**9.85E-210**
	Worst	1.45E-92	8.00E+01	1.39E-03	2.85E-102	1.19E-247	-3.19E-06	**1.64E-197**
	Mean	7.35E-94	3.59E+01	3.60E-04	2.08E-103	-2.83E-39	-6.50E+00	**1.41E-198**
	Std	3.24E-93	1.93E+01	3.62E-04	6.49E-103	1.26E-38	4.89E+00	**0.00E+00**
F2	Best	1.48E-185	1.30E+04	2.31E-12	4.73E+03	-3.34E-39	-1.00E+02	**0.00E+00**
	Worst	3.15E-148	5.62E+04	1.28E-05	4.39E+04	0.00E+00	-5.13E+01	**1.57E-291**
	Mean	1.57E-149	3.02E+04	1.26E-06	2.06E+04	-1.67E-40	-9.53E+01	**8.07E-293**
	Std	7.04E-149	1.06E+04	2.64E-06	1.14E+04	7.47E-40	1.46E+01	**0.00E+00**
F3	Best	4.61E-05	7.72E+01	**0.00E+00**	2.66E+01	9.99E-01	-3.00E+01	**0.00E+00**
	Worst	2.33E-02	9.01E+04	**0.00E+00**	2.85E+01	1.00E+00	7.96E-01	**0.00E+00**
	Mean	4.21E-03	3.03E+04	**0.00E+00**	2.72E+01	1.00E+00	-7.17E+00	**0.00E+00**
	Std	5.82E-03	4.30E+04	**0.00E+00**	4.89E-01	1.58E-04	1.19E+01	**0.00E+00**
F4	Best	1.69E-07	1.29E-14	9.16E-12	9.55E-03	-5.00E-01	-1.00E+02	**0.00E+00**
	Worst	3.38E-04	1.01E+04	4.91E-05	3.26E-01	-5.00E-01	-5.01E-01	**0.00E+00**
	Mean	5.27E-05	2.34E+03	5.69E-06	5.98E-02	-5.00E-01	-5.48E+00	**0.00E+00**
	Std	7.91E-05	4.31E+03	1.13E-05	8.80E-02	9.21E-05	2.22E+01	**0.00E+00**
F5	Best	**0.00E+00**	6.02E+01	3.91E-14	**0.00E+00**	**0.00E+00**	-5.12E+00	**0.00E+00**
	Worst	**0.00E+00**	1.86E+02	2.20E-05	**0.00E+00**	**0.00E+00**	-1.99E+00	**0.00E+00**
	Mean	**0.00E+00**	1.24E+02	1.98E-06	**0.00E+00**	**0.00E+00**	-4.65E+00	**0.00E+00**
	Std	**0.00E+00**	3.54E+01	4.33E-06	**0.00E+00**	**0.00E+00**	7.93E-01	**0.00E+00**
F6	Best	**0.00E+00**	3.32E-14	2.94E-09	**0.00E+00**	**0.00E+00**	-6.00E+02	**0.00E+00**
	Worst	**0.00E+00**	1.80E+02	8.34E-04	7.94E-02	3.47E-11	-8.04E-03	**0.00E+00**
	Mean	**0.00E+00**	3.98E+01	7.73E-05	3.97E-03	-1.12E-08	-3.31E+01	**0.00E+00**
	Std	**0.00E+00**	4.83E+01	1.66E-04	1.78E-02	1.04E-08	1.34E+02	**0.00E+00**
F7	Best	1.63E-09	6.83E-09	1.64E-10	1.42E-03	-1.00E+00	-8.99E+00	**1.57E-32**
	Worst	3.93E-06	1.17E+01	1.33E-04	2.10E-02	-1.00E+00	-1.05E+00	**3.92E-21**
	Mean	1.12E-06	1.70E+00	1.42E-05	7.53E-03	-1.00E+00	-4.28E+00	**1.96E-22**
	Std	1.32E-06	2.39E+00	3.06E-05	6.41E-03	6.97E-04	3.21E+00	**8.77E-22**
F8	Best	6.90E-07	2.75E-05	6.22E-13	2.66E-02	9.99E-01	-7.39E+00	**1.35E-32**
	Worst	9.28E-05	2.96E+01	1.34E-04	5.70E-01	1.00E+00	9.86E-01	**1.64E-21**
	Mean	2.11E-05	4.35E+00	1.51E-05	1.92E-01	1.00E+00	-1.48E-01	**8.18E-23**
	Std	2.68E-05	6.59E+00	3.36E-05	1.50E-01	3.20E-04	2.22E+00	**3.66E-22**
Total		8	0	4	5	0	0	32

which verifies that the algorithm proposed in this paper has better robustness.

5) FRIEDMAN TEST OF CEHHO AND OTHER ALGORITHMS

In order to verify that the experimental results are not accidental, this paper uses Friedman statistics to test. The statistical results are shown in Table 5. This table shows the Friedman ranking of the seven algorithms on the 8 test functions. It can be seen from the table that the CEHHO algorithm ranks first. Therefore, the algorithm proposed in this paper is superior in solving unimodal and multimodal functions.

V. APPLICATION OF CEHHO ALGORITHM IN DNAMIC OPTIMIZATION OF CHEMACAL INDUSTRY

In this paper, five classical chemical dynamic optimization problems are selected for experimental research to verify the feasibility and effectiveness of the CEHHO algorithm for chemical dynamic optimization. Each chemical problem is independently run 30 times, and the optimal value is selected as the experimental result. Compare the results of different methods. The experiment is as follows:

(1) Divide the problem into equal parts.

(2) Numerical solution with Runge–Kutta.

(3) Use the CEHHO algorithm to optimize the problem.

A. CASE STUDIES AND ANALYSIS

1) CASE 1: CLASSICAL CHEMICAL DYNAMICS PROBLEM

This problem is a classic benchmark dynamic optimization problem of the chemical industry dynamic optimization problem. In the system, the problem is optimally controlled by two state variables. The mathematical model is shown below.

$$\min J(u) = x_2(t_f)$$

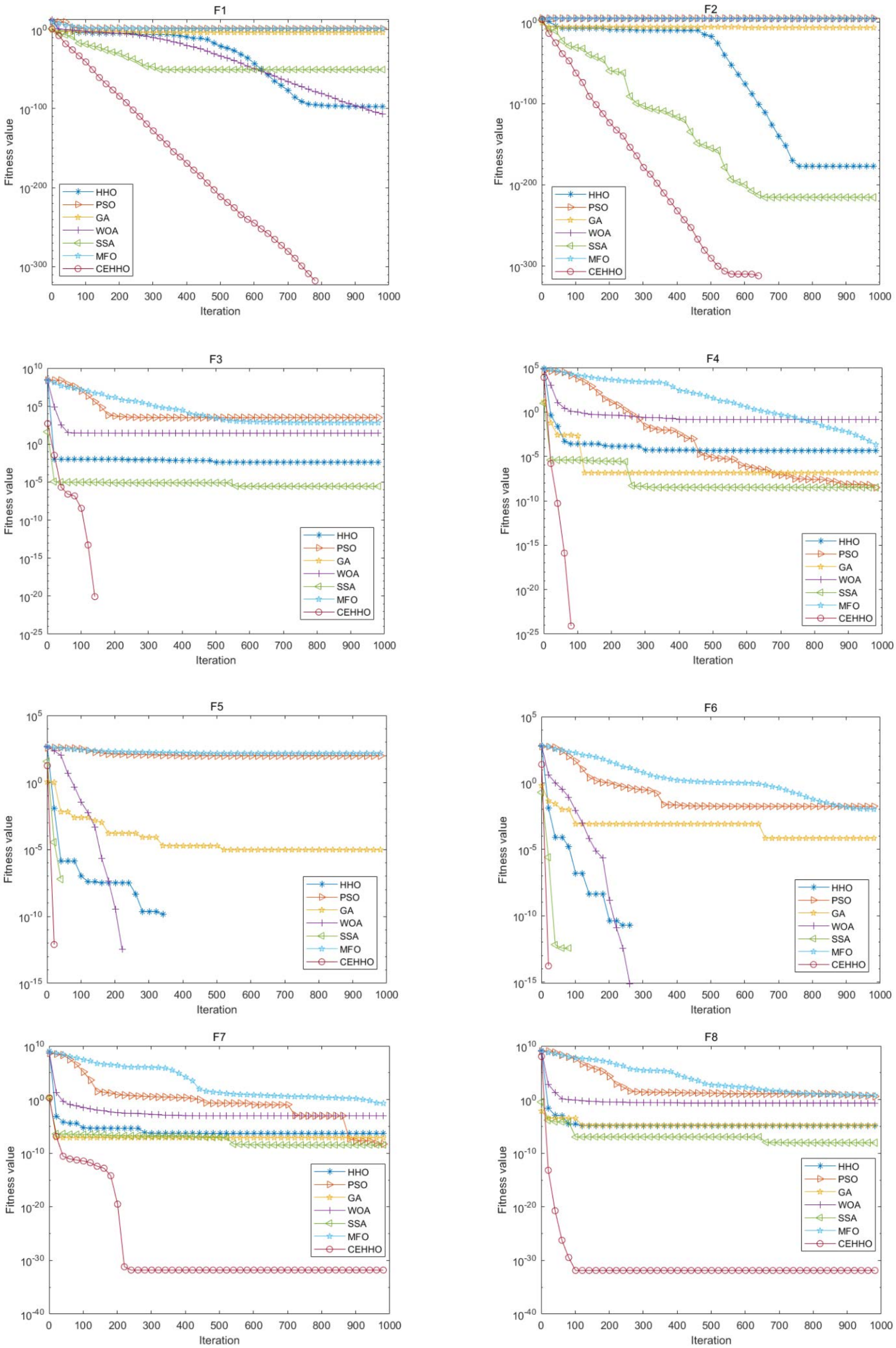


FIGURE 6. Convergence graph.

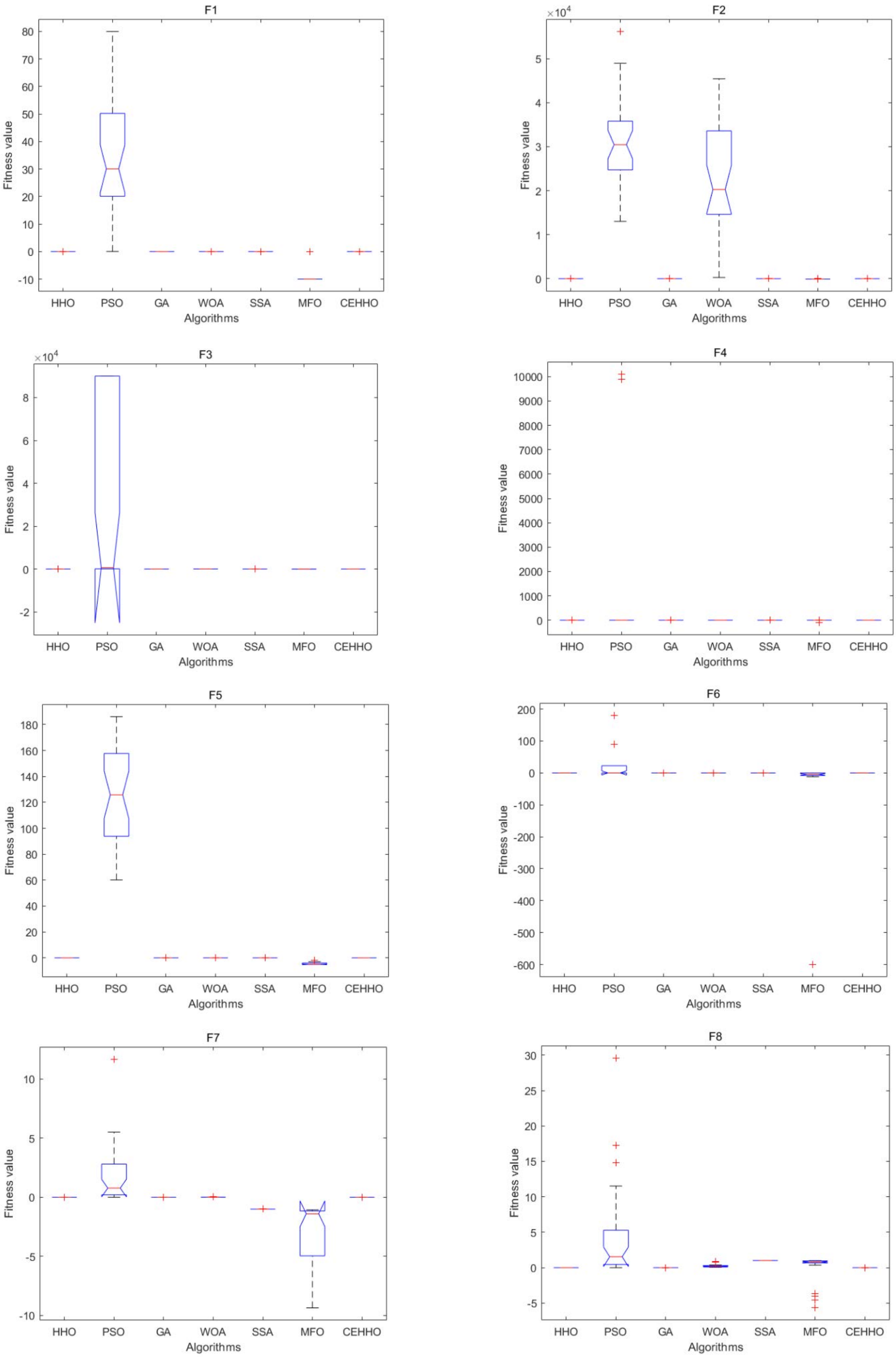


FIGURE 7. **Boxplot.**

TABLE 4. IQR value.

Function	HHO	PSO	GA	WOA	SSA	MFO	CEHHO
F1	1.72E-99	2.51E+01	5.96E-04	8.23E-106	3.19E-78	**0.00E+00**	8.30E-199
F2	1.16E-153	1.25E+04	9.89E-07	1.85E+04	3.23E-75	**0.00E+00**	**0.00E+00**
F3	2.57E-03	8.99E+04	0.00E+00	8.08E-01	9.79E-05	2.93E+01	**0.00E+00**
F4	2.72E-05	9.16E-10	6.48E-06	1.81E-01	8.44E-05	3.18E-03	**0.00E+00**
F5	**0.00E+00**	6.28E+01	2.36E-06	**0.00E+00**	3.28E-09	1.14E+00	**0.00E+00**
F6	**0.00E+00**	6.77E+01	6.58E-05	**0.00E+00**	1.24E-08	6.74E+00	**0.00E+00**
F7	2.98E-06	2.42E+00	6.62E-06	7.11E-03	4.61E-04	3.79E+00	**0.00E+00**
F8	4.63E-05	4.66E+00	1.32E-05	1.81E-01	5.93E-04	2.74E-01	**0.00E+00**

TABLE 5. Friedman test of CEHHO and other algorithms.

Algorithms	Function								Friedman	Friedman
	F1	F2	F3	F4	F5	F6	F7	F8	test	rank
HHO	7.35E-94	1.57E-149	4.21E-03	5.27E-05	0.00E+00	0.00E+00	1.12E-06	2.11E-05	3.1250	3
PSO	3.59E+01	3.02E+04	3.03E+04	2.34E+03	1.24E+02	3.98E+01	1.70E+00	4.35E+00	6.8750	7
GA	3.60E-04	1.26E-06	0.00E+00	5.69E-06	1.98E-06	7.73E-05	1.42E-05	1.51E-05	3.5625	4
WOA	2.08E-103	2.06E+04	2.72E+01	5.98E-02	0.00E+00	3.97E-02	7.53E-03	1.92E-01	4.6875	5
SSA	-2.83E-39	-1.67E-40	1.00E+00	-5.00E-01	0.00E+00	-1.12E-08	-1.00E+00	1.00E+00	2.1875	2
MFO	-6.50E+00	-9.53E+01	7.96E-01	-5.48E+00	-4.65E+00	-3.31E+01	-4.28E+00	-1.48E-01	5.6250	6
CEHHO	1.41E-198	8.07E-293	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.96E-22	8.18E-23	1.9375	1

$$s.t. \begin{cases} \dfrac{dx_1}{dt} = u \\ \dfrac{dx_2}{dt} = x_1^2 + u^2 \\ -1 \leq u \leq 0, \quad t_f = 1 \\ x(0) = [1\ 0]^T \end{cases} \tag{23}$$

In the formula, x1 and x2 are the two-state variables of the system. u is the control variable, t_f is the end time of the system, The solution to this problem is to find an optimal control variable u that minimizes the value of the state variable x2 at the end time, Table 6 lists the results of different methods to solve this problem. Figure 6 shows the change diagram of the control variable U with the number of segments N = 10 and N = 50. Figure 7 is the convergence diagram of the objective function with N = 50 and 100 iterations, and Figure 8 is the change diagram of the state variables in the iterative process.

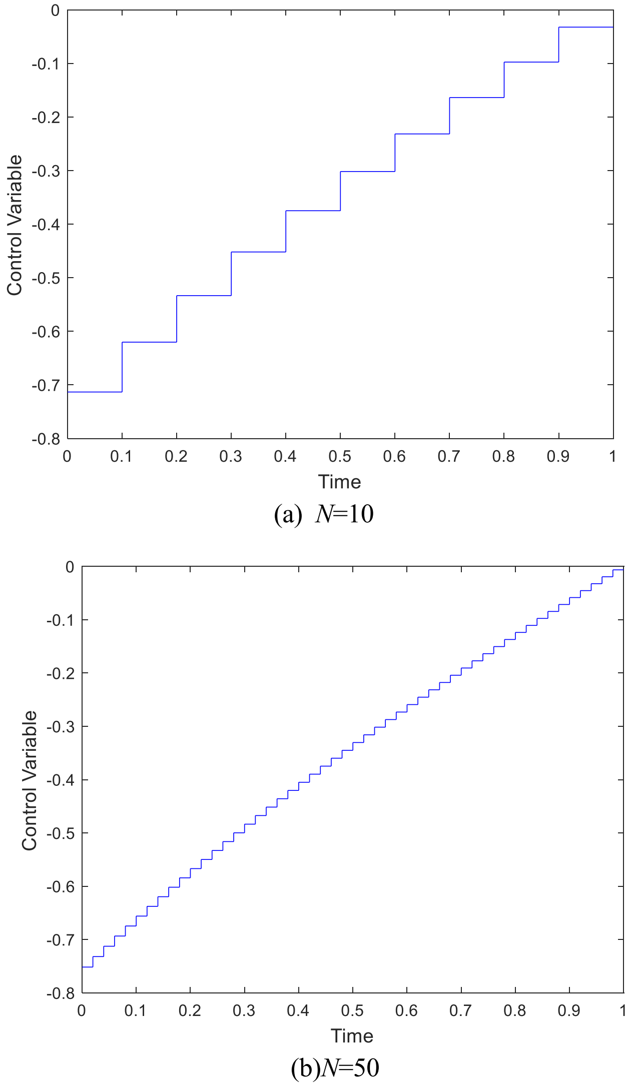
In Table 6, Time/s represents the run time for the method solving the chemical optimization problem.

It can be seen from Table 6 that the solution of this problem solved by the analytical solution method in Reference [44] is 0.761594156. Reference [6] proposes a hybrid differential evolution algorithm combined with Alopex. When the number of segments is $N = 50$, the obtained solution is 0.76162. The ADIWO-CVP method proposed by the literature [40] has a solution of 0.76159417 when the number of segments is $N = 50$, and the difference from the analytical solution is 0.000000014. The IWO-CVP method proposed by the literature [45] is in the When the number of segments is $N = 50$, the solution at this time is 0.76159793, and the difference from the analytical solution is 0.000003774. The CEHHO method proposed in this paper has a solution of 0.761596089 when the number of segments is $N = 50$, and the difference from the analytical solution is 0.000001933. Among the six methods proposed above,

TABLE 6. Experimental results of different methods in case 1.

Methods	N	Value	Time/s
IACO-CVP[43]	-[1]	0.76160	-
ACO-CP[29]	-	0.76238	-
OCT[44]	-	0.761594156	-
Alopex-DE[6]	10	0.762 09	-
Alopex-DE[6]	30	0.761 65	-
Alopex-DE[6]	50	0.761 62	-
IGA-CVP[45]	-	0.761595	-
IWO-CVP[45]	50	0.76159793	-
ADIWO-CVP[45]	50	0.76159417	-
CEHHO	10	0.762086624	27
CEHHO	30	0.761613849	70
CEHHO	50	0.761596089	117

[1] This symbol means not reported.



(a) N=10

(b)N=50

FIGURE 8. Optimal control trajectory diagram.

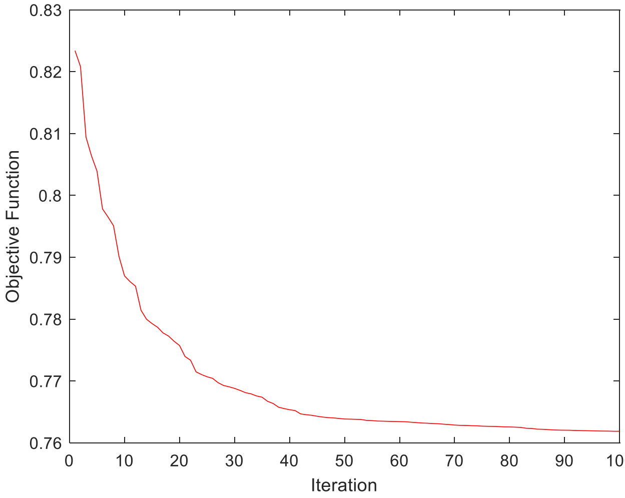


FIGURE 9. Iterative convergence diagram.

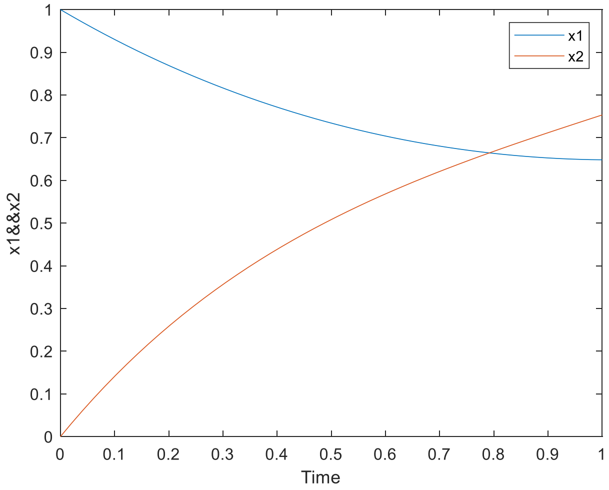


FIGURE 10. Trajectory convergence diagram of state variables.

the difference between the solution accuracy of the CEHHO algorithm and the ADIWO-CVP method is the smallest, and it is also the closest to the theoretical value. This proves the feasibility of the CEHHO algorithm.

2) CASE 2: CONTINUOUS REACTION IN BATCH REACTOR

A batch reactor refers to a device that performs chemical reactions intermittently. In the process of chemical production, it is often used to produce products of different specifications in batches. The reaction process of the batch reactor takes A as the production raw material, finally generates the target product B, and makes the target product B. The concentration value of product B is the largest at the terminal time, which is accompanied by the formation of by-product C. The reaction process is $A \rightarrow B \rightarrow C$. The reaction device is shown in Figure 9, and the reaction model is as follows:

$$\max \mathrm{J} = C_B(t_f)$$

$$\begin{cases} \dfrac{dC_A}{dt} = -k_1 C_A^2 \\ \dfrac{dC_B}{dt} = k_2 C_A^2 - k_2 C_B \\ 298K \leq T \leq 398K,\ C_A(0) = 1mol/L, \\ \qquad C_B(0) = 0mol/L,\ t_f = 1h \\ k_1 = 4 \times 10^3 \times e^{\frac{-2500}{T}},\ k_2 = 6.2 \times 10^5 \times e^{\frac{-5000}{T}} \end{cases} \tag{24}$$

Among them, C_A is the concentration of reactants in the reaction process, C_B is the concentration of the target product in the reaction process, T is the temperature in the reaction process, and J is the performance index of the model. Table 7 lists the solution results of different methods, Fig. 10 is the temperature control trajectories with the number of segments $N = 10, 20, 50$ in the reaction process, Fig. 11 is the iterative graph of the objective function, Fig. 12 is the reactant, and A graph of the concentration change of the product during the process.

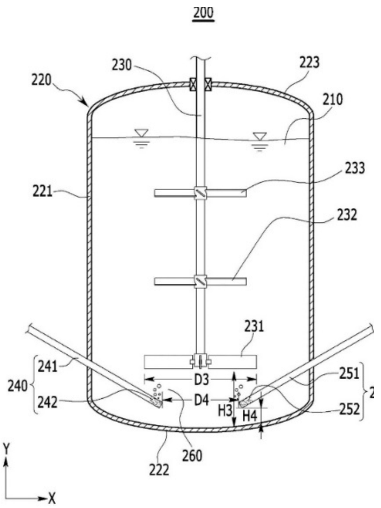
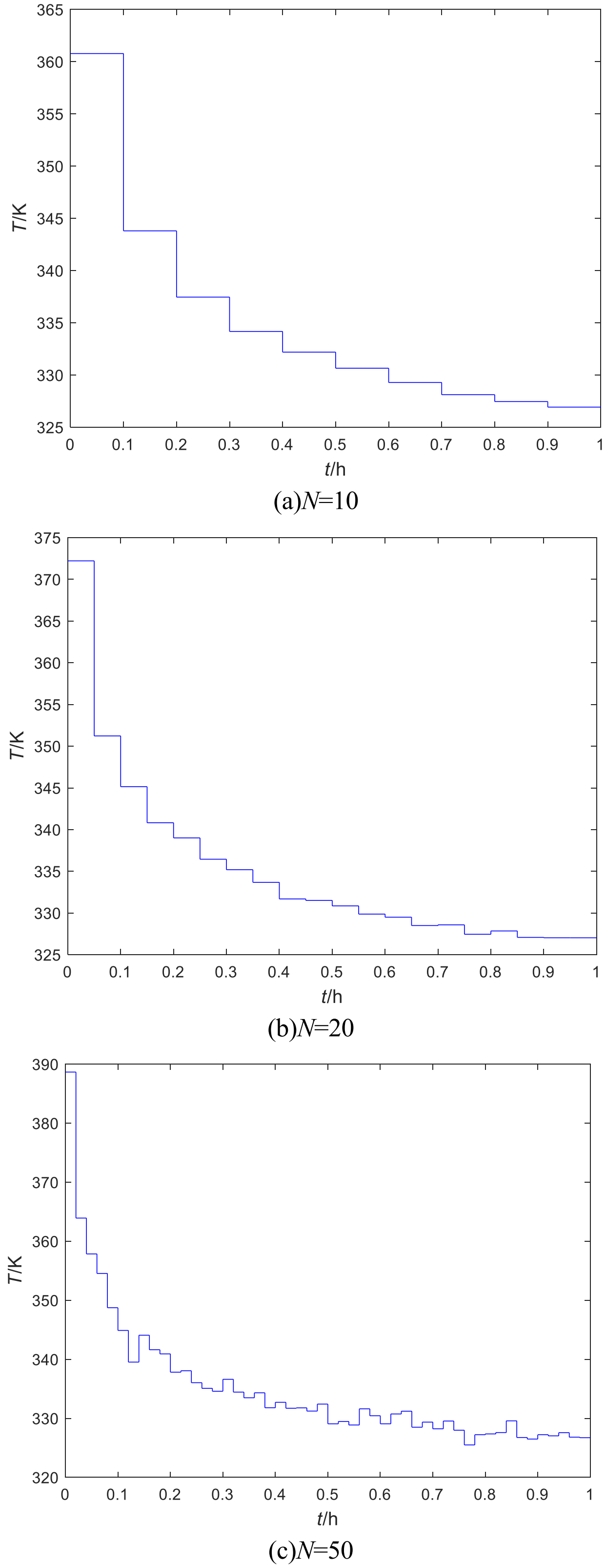


FIGURE 11. Batch reactor.

TABLE 7. Experimental results of different methods in case 2.

Methods	N	Value	Time/s
Alopex-DE[6]	10	0.61007	-
Alopex-DE[6]	20	0.61045	-
Alopex-DE[6]	50	0.61071	-
PSO-CVP[46]	25	0.6105359	-
SA[47]	50	0.6107	-
CVP-DE[17]	60	0.61072	-
ACO-CP[29]	-	0.61045	-
IACA[23]	-	0.6104	-
ISOA[48]	10	0.6101	-
ISOA[48]	25	0.61053	272
ISOA[48]	50	0.6107724	-
EBSO[49]	4	0.61045	-
EBSO[49]	25	0.61055712	-
EBSO[49]	50	0.61071215	-
CEHHO	10	0.61016918	35
CEHHO	20	0.61058090	50
CEHHO	50	0.61077985	120

It can be seen from Table 7 that when the number of segments is $N = 10$, the result of the literature [6] using the combination of Alopex and the differential evolution algorithm is 0.61007, and the literature [48] using the improved seagull optimization algorithm combined with the inequality method, a solution is 0.6101, the solution result of the CEHHO algorithm is 0.61016918, and the solution accuracy of the problem is relatively high. When the number of segments is $N = 25$, the result obtained by using the particle swarm combined with the control variable method in Reference [46] is 0.6105359, that of Reference [48] is 0.61053, and the solution of Reference [49] is 0.61055712 using the improved beetle, while CEHHO When the number of subsections $N = 20$, the solution result is 0.61058090, with fewer subsections and higher solution accuracy. When $N = 50$, the solution accuracy of the CEHHO algorithm is also the highest. The experimental results show that the CEHHO algorithm is effective in solving the continuous reaction in batch reactors.



(a)N=10

(b)N=20

(c)N=50

FIGURE 12. Variation diagram of control variables.

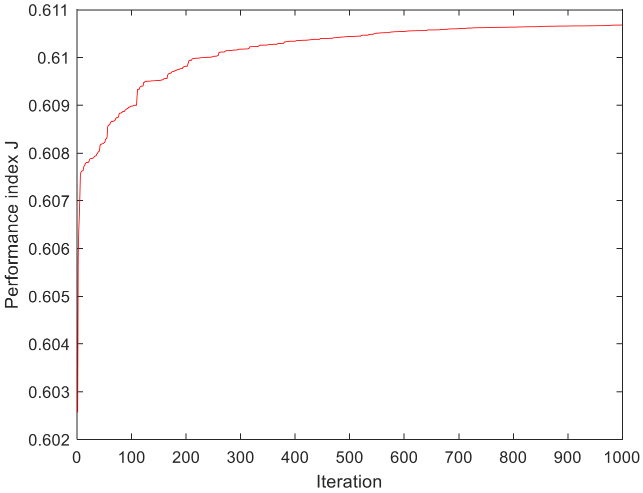


FIGURE 13. Iterative convergence diagram.

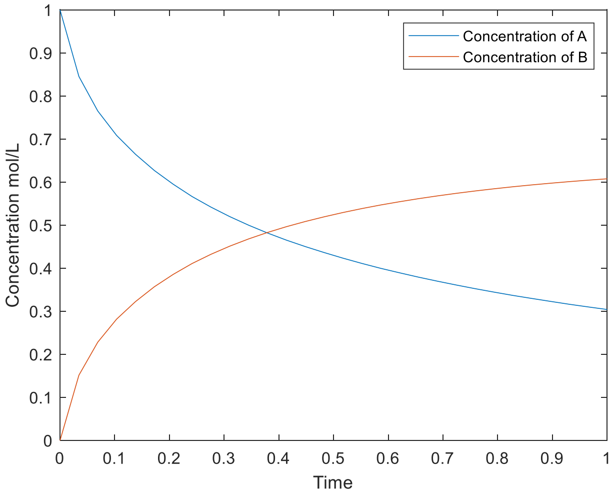


FIGURE 14. Concentration trajectories of reactants and products.

3) CASE 3: TUBULAR REACTOR

In modern industry, a tubular reactor is one of the most commonly used reactor forms due to its high volumetric efficiency, large heat transfer area, simple structure, and convenient processing. By mixing two reactants A and B, a chemical reaction occurs, The target product C is finally generated, and the concentration of the target product C is maximized. The reaction process is $A \leftrightarrow B \rightarrow C$. The model of the tubular reactor is as follows:

$$\max J(z_f) = 1 - C_A(z_f) - C_B(z_f)$$
$$\begin{cases} \dfrac{dC_A}{dz} = -u(z)[10 \times C_B(z) - C_A(z)] \\ \dfrac{dC_B}{dz} = u(z)[10 \times C_B(z) - C_A(z)] \\ \qquad - [1 - u(z)] \times C_B(z) \\ 0 \leq u(z) \leq 1, \ C_A(0) = 1mol/L, \\ \qquad C_B(0) = 0mol/L, \ z_f = 12m \end{cases} \tag{25}$$

In the formula, z is the length of the tubular reactor, C_A and C_B are the two reactants, and $u(z)$ is the catalyst content. Table 8 lists the solution results of different methods. Figure 13 shows the control trajectory of the control variable $u(z)$ when the number of segments is $N = 10, 20, 40,$ and 50 during the reaction process. Figure 14 shows the iteration of the objective function. Fig. 15 is a graph showing the concentration changes of reactants A and B during the reaction.

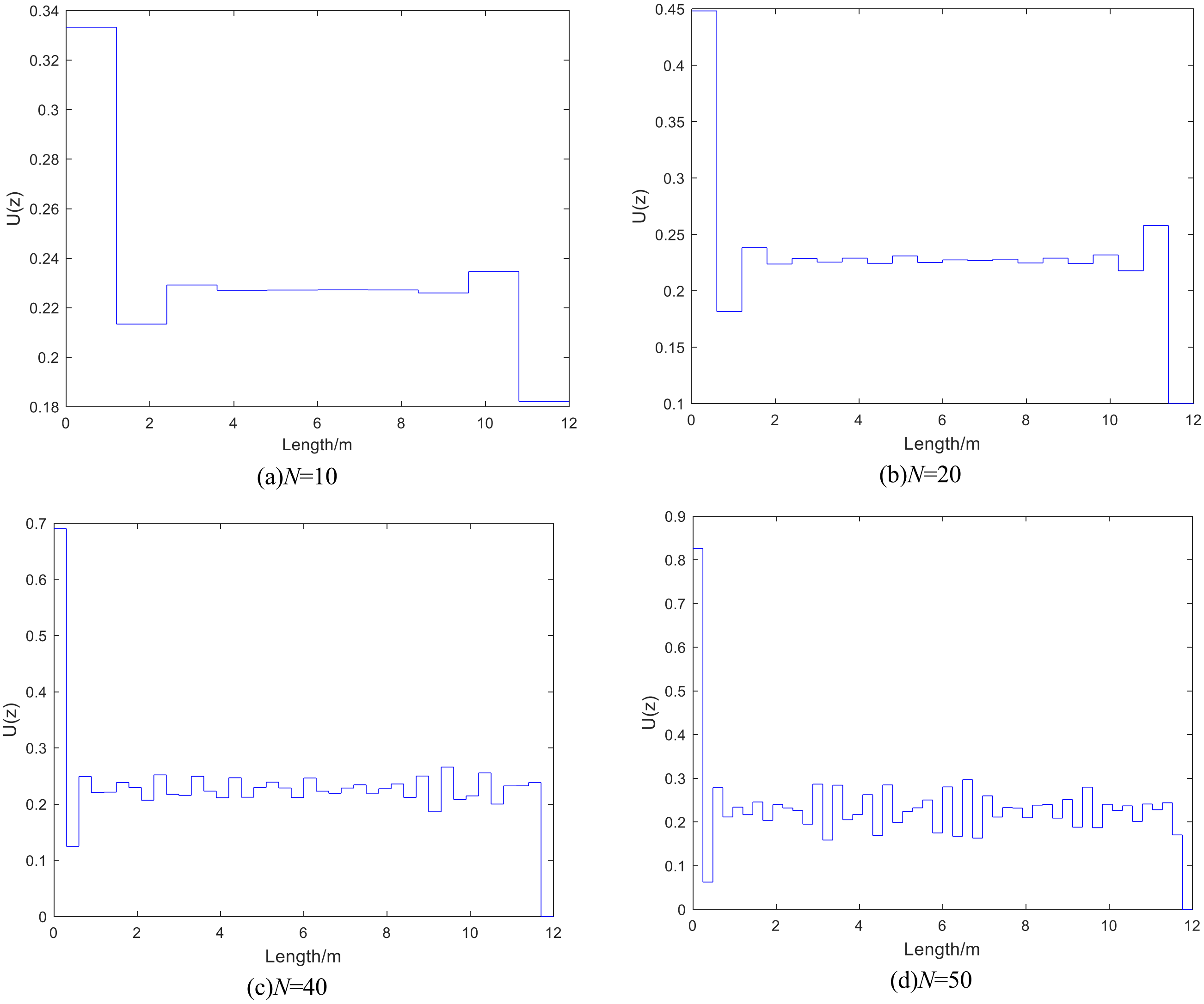
It can be seen from Table 8 that when the number of segments is $N = 10$, the result obtained by the control vector parameterized dynamic optimization based on the state transition algorithm adopted in the literature [9] is 0.47636, and the result in the literature [6] is 0.47363. The CEHHO algorithm proposed in this paper The result obtained is 0.47363015. When the number of segments is $N = 20$, the result obtained by the literature [3] is 0.475273, the result obtained by the literature [13] is 0.47527, and the result obtained by the CEHHO algorithm is 0.47527302. When $N = 40$, the literature [3] The result obtained is 0.476946, the result obtained by the literature [49] is 0.47697288, and the result obtained by the CEHHO algorithm is 0.47698071. When $N = 50$, the result obtained by the literature [6] is 0.47727, and the result obtained by the CEHHO algorithm is 0.47734337, the feasibility of the CEHHO algorithm is demonstrated by comparing the results of the above different numbers of segments.

TABLE 8. Experimental results of different methods in case 3.

Methods	Segments	Value	Time/s
STA[9]	5	0.47260	-
STA[9]	10	0.47363	-
STA[9]	15	0.47453	-
Reference[3]	20	0.475273	-
Reference[3]	40	0.476946	-
Alopex-DE[6]	10	0.47363	-
Alopex-DE[6]	30	0.47636	-
Alopex-DE[6]	50	0.47727	-
EBSO[49]	5	0.47426117	-
EBSO[49]	15	0.46011742	-
EBSO[49]	40	0.47697288	-
TDE[13]	20	0.47527	-
TDE[13]	40	0.47683	-
CEHHO	10	0.47363015	108
CEHHO	20	0.47527302	216
CEHHO	40	0.47698071	420
CEHHO	50	0.47734337	540

4) CASE 4: OPTIMIZATION OF CONTINUOUS STIRRED TANK REACTOR

A continuous stirred tank reactor is a tool in chemical production. In the CSTR system, the parameter with influence is the



(a)N=10

(b)N=20

(c)N=40

(d)N=50

FIGURE 15. **Variation diagram of control variable.**

reaction temperature or concentration. If the temperature and concentration exceed a deviation, the quality of the product will be affected.so, it is particularly important to control the variation range of temperature and concentration. The reaction device is shown in Figure 16, and the reaction model is as follows:

$$\min_{u} J = \int_{0}^{t_f} (x_1^2(t) + x_2^2(t) + 0.1 * u^2(t))dt$$

$$s.t. \begin{cases} x_1 = -(2+u)(x_1+0.25) \\ \qquad\qquad + (x_2+0.5)\exp(\dfrac{25x_1}{x_1+2}) \\ x_2 = 0.5 - x_2 - (x_2+0.5)\exp(\dfrac{25x_1}{x_1+2}) \end{cases}$$

$$x(0) = [0.09\ 0.09], \quad 0 \le u(t) \le 5, \ t_f = 0.78 \qquad (26)$$

In the formula, x_1 and x_2 represent the deviation of temperature and concentration, respectively, u is the control variable, which represents the control of the flow rate of the cooling liquid inserted into the reactor through the coil, and the function of the control variable u is to keep x_1 and x_2 in their steady-state value. Table 9 shows the solution results of different methods, Fig. 17 is the control variable trajectory diagram, Fig. 18 is the objective function iteration diagram, and Fig. 19 is the change diagram of temperature and concentration.

It can be seen from Table 9 that when the number of segments is $N = 10$, the solution result of literature [2] is 0.13406, the solution result of literature [25] using the differential evolution algorithm is 0.13559, the result obtained by literature [51] is 0.13342, the solution of literature [9] is 0.13342 The result obtained is 0.13729, and the result obtained by the CEHHO algorithm proposed in this paper is 0.13529454. At this time, the solution result of Reference [52]

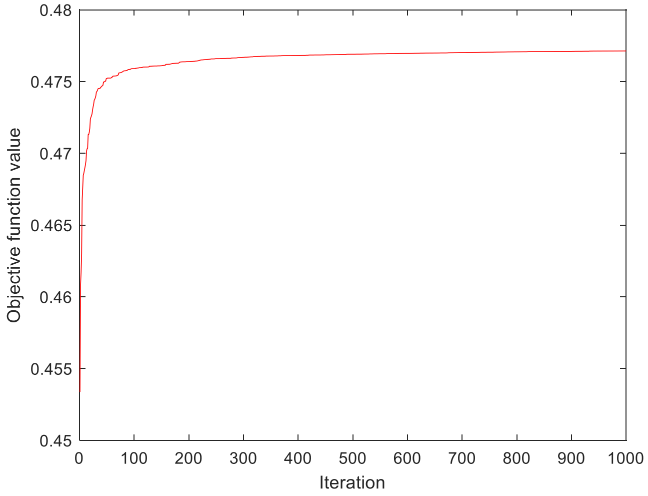


FIGURE 16. Iteration diagram of objective function.

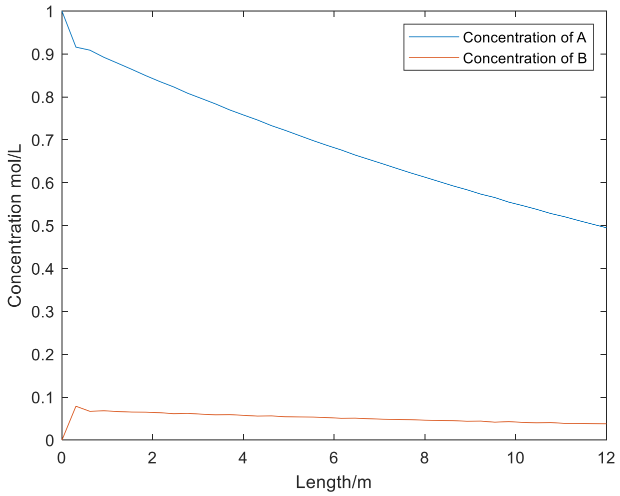


FIGURE 17. Concentration changes of reactants A and B.

is better. When the number of segments is $N = 20$, the solution result of the CEHHO algorithm is better than that of Reference [50] and Reference [52], [4] has higher precision and saves time overhead based on fewer segments. When the number of segments is N > 40, the CEHHO solution performance gets worse, and the CEHHO algorithm obtains the optimal solution as $N = 40$.

5) CASE 5: PARALLEL REACTIONS IN TUBULAR REACTORS

Parallel reaction refers to the process in which the reactants participating in the reaction can carry out several different reactions in parallel without affecting each other, and finally generate the main product and the secondary product. The faster reaction or larger proportion is called the main reaction. The rest are sided reactions. The reaction process is A- > B, A- > C, and the mathematical model established is as follows:

$$\max J(t_f) = C_2(t_f)$$

TABLE 9. Experimental results of different methods in case 4.

Methods	Segments	Value	Time/s
ndCVP_HGPSO[2]	5	0.13642	-
ndCVP_HGPSO[2]	10	0.13406	-
ndCVP_HGPSO[2]	15	0.13366	-
DE[25]	13	0.13559	-
IDP[50]	78	0.13321	-
LCA[51]	5	0.13324	-
LCA[51]	10	0.13342	-
PMP[4]	78	0.13340	-
STA[9]	10	0.13729	-
STA[9]	15	0.13500	-
STA[9]	20	0.13419	-
CEHHO	10	0.13529454	65
CEHHO	20	0.13272280	128
CEHHO	40	0.13247583	253
CEHHO	50	0.13342693	327
CEHHO	70	0.13478319	448

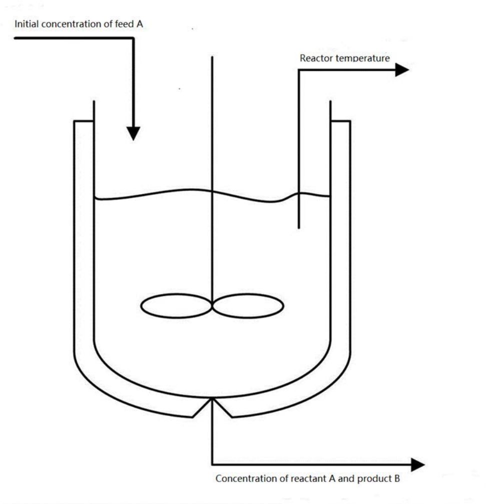
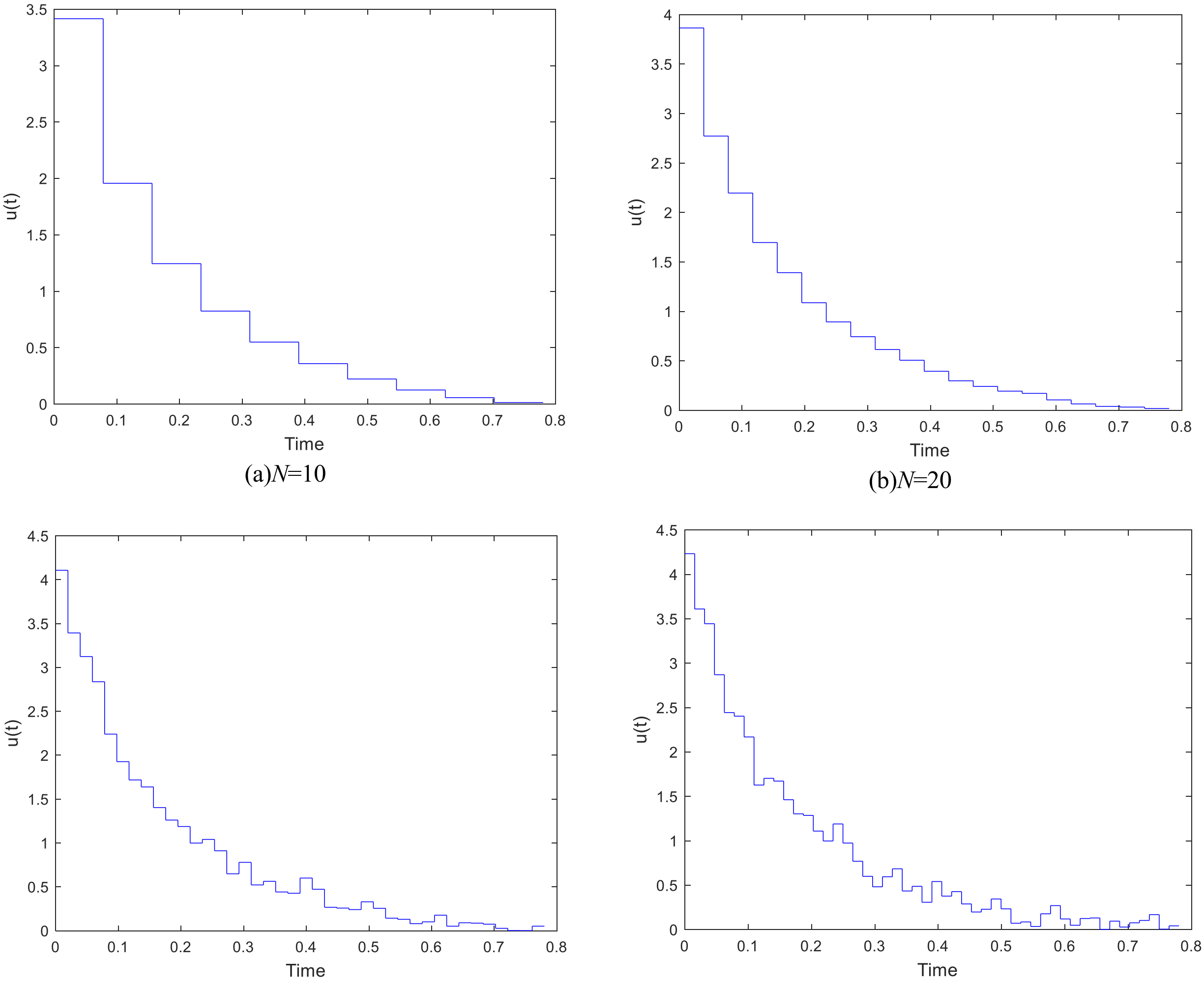

Initial concentration of feed A
Reactor temperature
Concentration of reactant A and product B

FIGURE 18. Continuous stirred tank reactor.

$$\begin{cases} \dfrac{dC_1}{dt} = -[u(t) + 0.5u^2(t)]C_1(t) \\ \dfrac{dC_2}{dt} = u(t)C_1(t) \\ 0 \leq u(t) \leq 5,\ C_1(0) = 1mol/L, \\ \qquad\qquad C_2(0) = 0mol/L,\ t_f = 1h \end{cases} \tag{27}$$

In the formula, $C_1(t)$ is the concentration of reactant A participating in the reaction, $C_2(t)$ is the concentration of product B, and $u(t)$ is the saturation of the control variable. Table 10 shows the solution results of different methods. Figure 20 shows the trajectory of the control variables with $N = 10$, 20, 30, and 50. Figure 21 shows the iteration of the objective function. Figure 22 shows the concentration changes of reactants and products.

It can be seen from Table 10 that when the number of segments is $N = 10$, the solution result of literature [6] is



(a)N=10

(b)N=20

(c)N=40

(d)N=50

FIGURE 19. Control variable trajectory diagram.

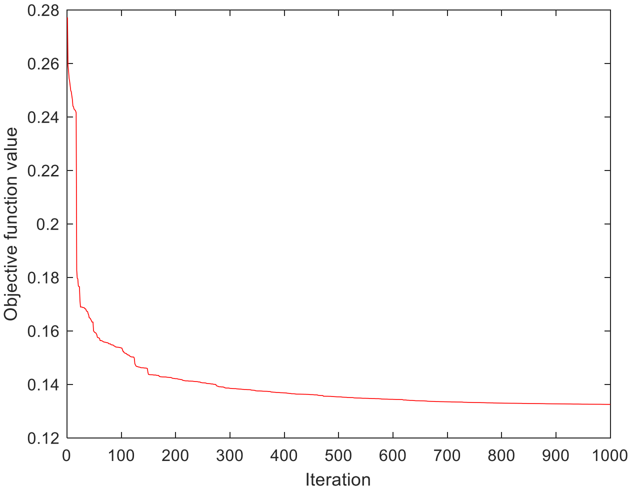


FIGURE 20. Iteration diagram of objective function(N = 40).

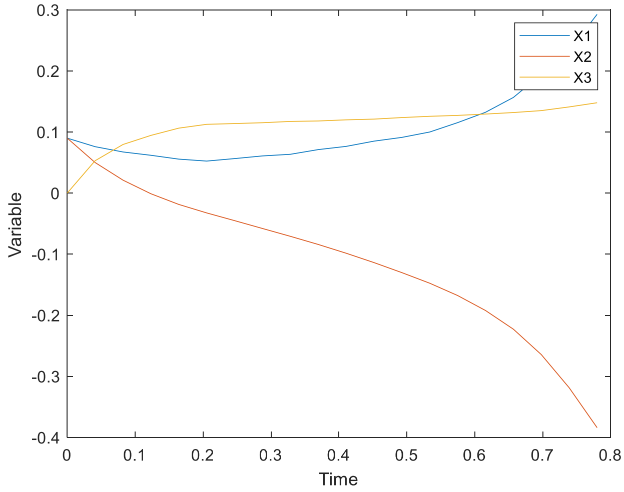
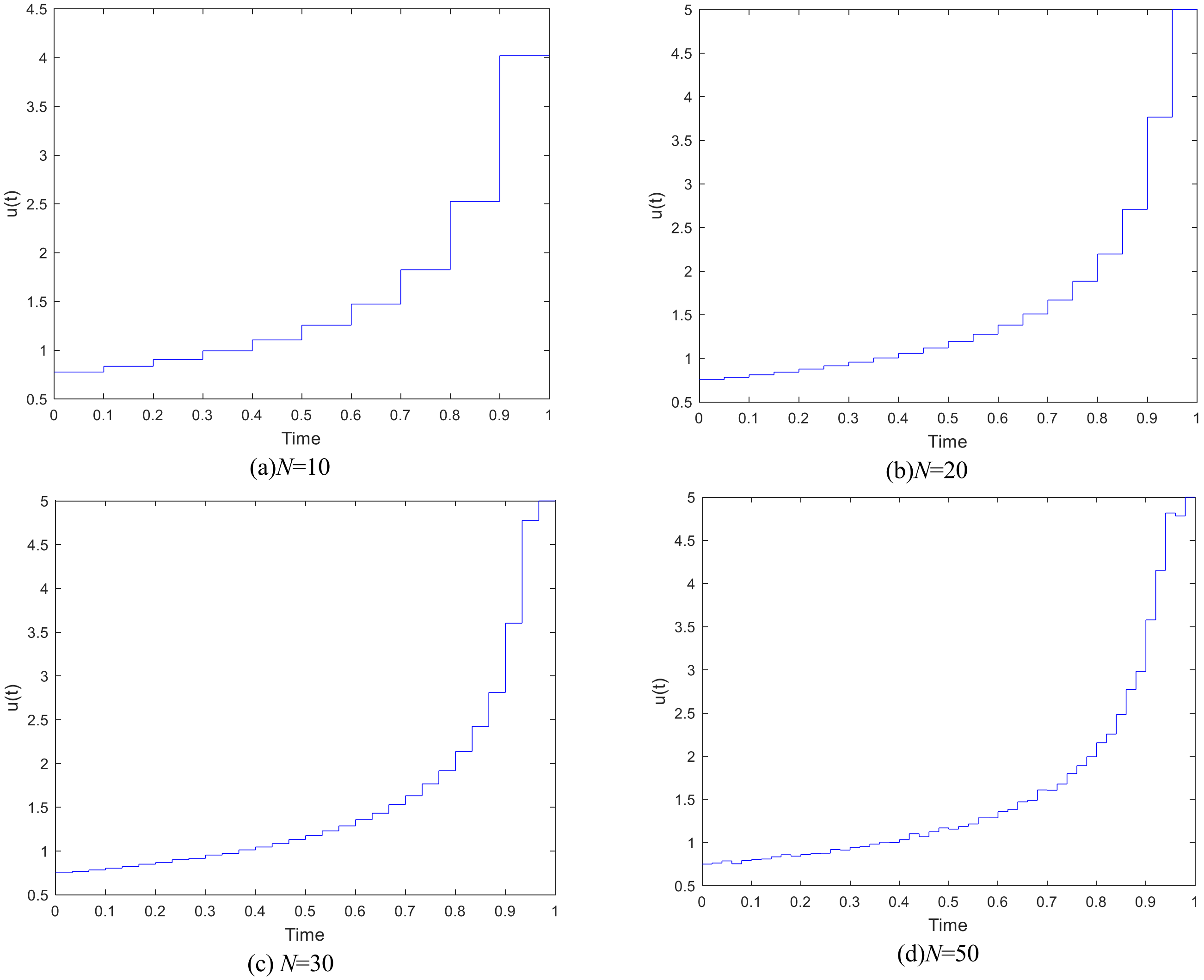


FIGURE 21. Variable change diagram.

0.57224, the solution result of literature [3] is 0.572241, and the solution result of the CEHHO algorithm proposed in this paper is 0.57224198, and the solution result is similar to the above literature. When the number of segments is $N = 20$, the result of the literature [3] is 0.57330, and the solution result of the CEHHO algorithm is 0.57339573. When the number



(a)*N*=10

(b)*N*=20

(c) *N*=30

(d)*N*=50

FIGURE 22. Control variable trajectory diagram.

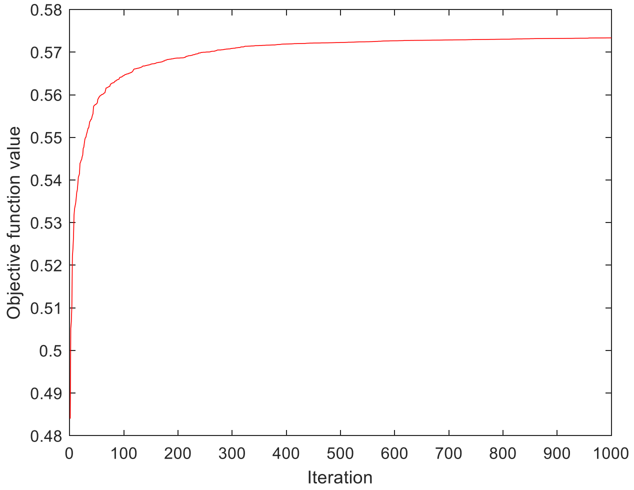


FIGURE 23. Iteration diagram of objective function.

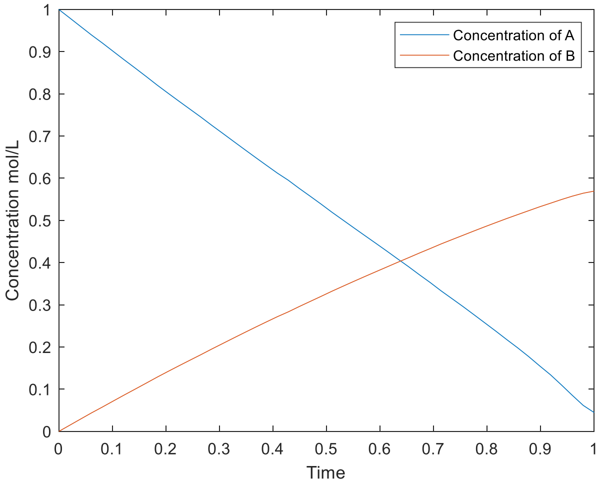


FIGURE 24. Graph of height change of reactants and products.

of segments is $N = 30$, the result of the literature [6] is 0.57342, and the solution result of the CEHHO algorithm is 0.57349470. The solution accuracy of the CEHHO algorithm is higher than that of the literature [45] with the number of segments $N = 40$, which adopts the average segmentation combined with the seagull optimization algorithm and

TABLE 10. Experimental results of different methods in case 5.

Methods	Segments	Value	Time/s
Equal Division(ISOA)[51]	40	0.573073	326
Unequal Division(ISOA)[51]	40	0.573535	302
CPT[52]	-	0.57353	-
Alopex-DE[6]	10	0.57224	-
Alopex-DE[6]	30	0.57342	-
Alopex-DE[6]	50	0.5735	-
Alopex-DE[6]	80	0.57353	-
AEPF[3]	10	0.572241	-
AEPF[3]	20	0.57330	-
AEPF[3]	40	0.57348	-
AEPF[3]	80	0.57353	-
CEHHO	10	0.57224198	18
CEHHO	20	0.57339573	25
CEHHO	30	0.57349470	40
CEHHO	50	0.57357526	58

literature [3]. When $N = 50$, the solution result of the CEHHO algorithm is 0.57357526, which is the highest in Table 10. the figure of merit.

VI. CONCLUSION

This paper proposes a chaotic elite Harris hawk optimization algorithm. First, since the initialized population influences the quality of the solution, Logistics chaos is used instead of random initialization to enhance the diversity of the population, which plays a role in preventing the algorithm from falling into local optimum. Secondly, since the transition from the exploration stage to the exploration stage is determined by the escape energy factor, and the escape energy factor used in the original text is linearly decreasing, only the global search is carried out in the early stage of the population evolution, and only local exploration is carried out in the later stage, which is easy to fall into Local optimization, to better adjust the global and local dynamic balance, this paper proposes a nonlinear decreasing escape energy factor. Finally, for the hard siege and soft siege strategies in the development stage, according to the average position information of individuals, this paper proposes a new update strategy and based on the idea of greedy selection, the position with better fitness value is reserved. Through the numerical experiments of unimodal and multimodal functions, the experimental results show that the CEHHO algorithm has the advantages of avoiding the population falling into local optimum, premature convergence, and improving the solution accuracy. The CEHHO algorithm combined with the bisection method is used to solve various optimization problems in the dynamic optimization of the chemical industry. Through five chemical experiments, it is shown that the CEHHO algorithm has a good solution ability.

Although the improved HHO algorithm shows good solution performance, in terms of time complexity, due to too many divisions, the time cost is the largest. Therefore, in the future research direction, non-uniform discretization can be used to control the control variables are split.

REFERENCES

[1] R. Irizarry, "A generalized framework for solving dynamic optimization problems using the artificial chemical process paradigm: Applications to particulate processes and discrete dynamic systems," *Chem. Eng. Sci.*, vol. 60, no. 21, pp. 5663–5681, Nov. 2005.

[2] X. Chen, W. Du, H. Tianfield, R. Qi, W. He, and F. Qian, "Dynamic optimization of industrial processes with nonuniform discretization-based control vector parameterization," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 4, pp. 1289–1299, Oct. 2014.

[3] S. A. Dadebo and K. B. Mcauley, "Dynamic optimization of constrained chemical engineering problems using dynamic programming," *Comput. Chem. Eng.*, vol. 19, no. 5, pp. 513–525, May 1995.

[4] L. Bayón, J. M. Grau, M. M. Ruiz, and P. M. Suárez, "Initial guess of the solution of dynamic optimization of chemical processes," *J. Math. Chem.*, vol. 48, no. 1, pp. 28–37, Jul. 2010.

[5] R. Sundaralingam, "Two-step method for dynamic optimization of inequality state constrained systems using iterative dynamic programming," *Ind. Eng. Chem. Res.*, vol. 54, no. 31, pp. 7658–7667, Jul. 2015.

[6] J. Zhu, X. Yan, and W. Zhao, "Chemical process dynamic optimization based on the differential evolution algorithm with an adaptive scheduling mutation strategy," *Eng. Optim.*, vol. 45, no. 10, pp. 1205–1221, Oct. 2013.

[7] F. Assassa and W. Marquardt, "Dynamic optimization using adaptive direct multiple shooting," *Comput. Chem. Eng.*, vol. 60, pp. 242–259, Jan. 2014.

[8] D. Huang and X. L. Luo, "Process transition based on dynamic optimization with the case of a throughput-fluctuating ethylene column," *Ind. Eng. Chem. Res.*, vol. 57, no. 18, pp. 6292–6302, 2018.

[9] M. Huang, X. Zhou, C. Yang, and W. Gui, "Dynamic optimization using control vector parameterization with state transition algorithm," in *Proc. 36th Chin. Control Conf. (CCC)*, Dalian, China, Jul. 2017, pp. 4407–4412, doi: 10.23919/ChiCC.2017.8028052.

[10] Q. Fan, Z. Chen, and Z. Xia, "A novel quasi-reflected Harris hawks optimization algorithm for global optimization problems," *Soft Comput.*, vol. 24, no. 19, pp. 14825–14843, Mar. 2020.

[11] Y. Zhang and Y. Mo, "Dynamic optimization of chemical processes based on modified sailfish optimizer combined with an equal division method," *Processes*, vol. 9, no. 10, p. 1806, Oct. 2021.

[12] M. Vallerio, D. Telen, L. Cabianca, F. Manenti, J. V. Impe, and F. Logist, "Robust multi-objective dynamic optimization of chemical processes using the sigma point method," *Chem. Eng. Sci.*, vol. 140, pp. 201–216, Feb. 2016.

[13] R. Angira and A. Santosh, "Optimization of dynamic systems: A trigonometric differential evolution approach," *Comput. Chem. Eng.*, vol. 31, no. 9, pp. 1055–1063, Sep. 2007.

[14] F. Sun, W. Du, R. Qi, F. Qian, and W. Zhong, "A hybrid improved genetic algorithm and its application in dynamic optimization problems of chemical processes," *Chin. J. Chem. Eng.*, vol. 21, no. 2, pp. 144–154, Feb. 2013.

[15] X. Chen, W. Du, and F. Qian, "Solving chemical dynamic optimization problems with ranking-based differential evolution algorithms," *Chin. J. Chem. Eng.*, vol. 24, no. 11, pp. 1600–1608, Nov. 2016.

[16] X. Chen, C. Mei, B. Xu, K. Yu, and X. Huang, "Quadratic interpolation based teaching-learning-based optimization for chemical dynamic system optimization," *Knowl.-Based Syst.*, vol. 145, pp. 250–263, Apr. 2018.

[17] F. Sun, W. Zhong, H. Cheng, and F. Qian, "Novel control vector parameterization method with differential evolution algorithm and its application in dynamic optimization of chemical processes," *Chin. J. Chem. Eng.*, vol. 21, no. 1, pp. 64–71, Jan. 2013.

[18] G. G. Tejani, V. J. Savsani, and V. K. Patel, "Adaptive symbiotic organisms search (SOS) algorithm for structural design optimization," *J. Comput. Des. Eng.*, vol. 3, no. 3, pp. 226–249, Jul. 2016.

[19] G. G. Tejani, N. Pholdee, S. Bureerat, and D. Prayogo, "Multiobjective adaptive symbiotic organisms search for truss optimization problems," *Knowl.-Based Syst.*, vol. 161, pp. 398–414, Dec. 2018.

[20] G. G. Tejani, N. Pholdee, S. Bureerat, D. Prayogo, and A. H. Gandomi, "Structural optimization using multi-objective modified adaptive symbiotic organisms search," *Expert Syst. Appl.*, vol. 125, no. 1, pp. 425–441, Jul. 2019.

[21] V. J. Savsani, G. G. Tejani, and V. K. Patel, ''Truss topology optimization with static and dynamic constraints using modified subpopulation teaching–learning-based optimization,'' *Eng. Optim.*, vol. 48, no. 11, pp. 1990–2006, Mar. 2016.

[22] S. Kumar, P. Jangir, G. G. Tejani, M. Premkumar, and H. H. Alhelou, ''MOPGO: A new physics-based multi-objective plasma generation optimizer for solving structural optimization problems,'' *IEEE Access*, vol. 9, pp. 84982–85016, 2021.

[23] B. Zhang, D. Chen, and W. Zhao, ''Iterative ant-colony algorithm and its application to dynamic optimization of chemical process,'' *Comput. Chem. Eng.*, vol. 29, no. 10, pp. 2078–2086, Sep. 2005.

[24] M. Huang, X. Zhou, T. Huang, C. Yang, and W. Gui, ''Dynamic optimization based on state transition algorithm for copper removal process,'' *Neural Comput. Appl.*, vol. 31, no. 7, pp. 2827–2839, Jul. 2019.

[25] I. L. Lopez Cruz, L. G. Van Willigenburg, and G. Van Straten, ''Efficient differential evolution algorithms for multimodal optimal control problems,'' *Appl. Soft Comput.*, vol. 3, no. 2, pp. 97–122, Sep. 2003.

[26] S. R. Upreti, ''A new robust technique for optimal control of chemical engineering processes,'' *Comput. Chem. Eng.*, vol. 28, no. 8, pp. 1325–1336, Jul. 2004.

[27] P. S. Shelokar, V. K. Jayaraman, and B. D. Kulkarni, ''Multicanonical jump walk annealing assisted by Tabu for dynamic optimization of chemical engineering processes,'' *Eur. J. Oper. Res.*, vol. 185, no. 3, pp. 1213–1229, Mar. 2008.

[28] Q. T. Pham, ''Using fuzzy logic to tune an evolutionary algorithm for dynamic optimization of chemical processes,'' *Comput. Chem. Eng.*, vol. 37, pp. 136–142, Feb. 2012.

[29] J. Rajesh, K. Gupta, H. S. Kusumakar, V. K. Jayaraman, and B. D. Kulkarni, ''Dynamic optimization of chemical processes using ant colony framework,'' *Comput. Chem.*, vol. 25, no. 6, pp. 583–595, Nov. 2001.

[30] H. Hadiyanto, D. C. Esveld, R. M. Boom, G. van Straten, and A. J. B. van Boxtel, ''Control vector parameterization with sensitivity based refinement applied to baking optimization,'' *Food Bioproducts Process.*, vol. 86, no. 2, pp. 130–141, Jun. 2008.

[31] P. Liu, G. Li, X. Liu, and Z. Zhang, ''Novel non-uniform adaptive grid refinement control parameterization approach for biochemical processes optimization,'' *Biochem. Eng. J.*, vol. 111, pp. 63–74, Jul. 2016.

[32] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, ''Harris hawks optimization: Algorithm and applications,'' *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.

[33] M. Javidi and R. Hosseinpourfard, ''Chaos genetic algorithm instead genetic algorithm,'' *Int. Arab J. Inf. Technol.*, vol. 12, no. 2, pp. 163–168, Mar. 2015.

[34] X. Yuan, J. Zhao, Y. Yang, and Y. Wang, ''Hybrid parallel chaos optimization algorithm with harmony search algorithm,'' *Appl. Soft Comput.*, vol. 17, pp. 12–22, Apr. 2014.

[35] A. H. Gandomi, X-S. Yang, S. Talatahari, and A. H. Alavi, ''Firefly algorithm with chaos,'' *Commun. Nonlinear Sci. Numer. Simul.*, vol. 18, no. 1, pp. 89–98, Jan. 2013.

[36] H. R. Tizhoosh, ''Opposition-based learning: A new scheme for machine intelligence,'' in *Proc. Int. Conf. Comput. Intell. Modeling, Control Autom. Int. Conf. Intell. Agents, Web Technol. Internet Commerce (CIMCA-IAWTIC)*, Nov. 2005, pp. 695–701, doi: 10.1109/CIMCA.2005.1631345.

[37] A. A. Ewees, M. A. Elaziz, and E. H. Houssein, ''Improved grasshopper optimization algorithm using opposition-based learning,'' *Expert Syst. Appl.*, vol. 112, pp. 156–172, Dec. 2018.

[38] S. Gupta and K. Deep, ''A hybrid self-adaptive sine cosine algorithm with opposition based learning,'' *Expert Syst. Appl.*, vol. 119, pp. 210–230, Apr. 2019.

[39] H. Chen, W. Li, and X. Yang, ''A whale optimization algorithm with chaos mechanism based on quasi-opposition for global optimization problems,'' *Expert Syst. Appl.*, vol. 158, Nov. 2020, Art. no. 113612.

[40] S. Mirjalili and A. Lewis, ''The whale optimization algorithm,'' *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.

[41] J. Xue and B. Shen, ''A novel swarm intelligence optimization approach: Sparrow search algorithm,'' *Syst. Sci. Control Eng.*, vol. 8, no. 1, pp. 22–34, Jan. 2020.

[42] S. Mirjalili, ''Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm,'' *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.

[43] S. A. Asgari and M. R. Pishvaie, ''Dynamic optimization in chemical processes using region reduction strategy and control vector parameterization with an ant colony optimization algorithm,'' *Chem. Eng. Technol.*, vol. 31, no. 4, pp. 507–512, Apr. 2008.

[44] D. Leonard, N. Van Long, and V. L. Ngo, *Optimal Control Theory and Static Optimization in Economics*. Cambridge, U.K.: Cambridge Univ. Press, 1992.

[45] J. Tian, P. Zhang, Y. Wang, X. Liu, C. Yang, W. Gui, and Y. Sun, ''Control vector parameterization-based adaptive invasive weed optimization for dynamic processes,'' *Chem. Eng. Technol.*, vol. 41, no. 5, pp. 964–974, Mar. 2018.

[46] B. Shi, Y. Yin, and F. Liu, ''Optimal control strategies combined with PSO and control vector parameterization for batchwise chemical process,'' *CIESC J.*, pp. 979–986, 2019.

[47] R. Faber, T. Jockenhövel, and G. Tsatsaronis, ''Dynamic optimization with simulated annealing,'' *Comput. Chem. Eng.*, vol. 29, no. 2, pp. 273–290, Jan. 2005.

[48] L. Xu, Y. Mo, Y. Lu, and J. Li, ''Improved seagull optimization algorithm combined with an unequal division method to solve dynamic optimization problems,'' *Processes*, vol. 9, no. 6, p. 1037, Jun. 2021.

[49] Y. Lyu, Y. Mo, Y. Lu, and R. Liu, ''Enhanced beetle antennae algorithm for chemical dynamic optimization problems' non-fixed points discrete solution,'' *Processes*, vol. 10, no. 1, p. 148, Jan. 2022.

[50] R. Luus, ''Optimal control by dynamic programming using systematic reduction in grid size,'' *Int. J. Control*, vol. 51, no. 5, pp. 995–1013, Jan. 1990.

[51] D.-Y. Sun, P.-M. Lin, and S.-P. Lin, ''Integrating controlled random search into the line-up competition algorithm to solve unsteady operation problems,'' *Ind. Eng. Chem. Res.*, vol. 47, no. 22, pp. 8869–8887, Oct. 2008.

[52] V. S. Vassiliadis, R. W. H. Sargent, and C. C. Pantelides, ''Solution of a class of multistage dynamic optimization Problems. 2. problems with path constraints,'' *Ind. Eng. Chem. Res.*, vol. 33, no. 9, pp. 2123–2133, Sep. 1994.

LILA HONG received the B.S. degree in computer science and technology from Zunyi Normal University, Zunyi, China, in 2019. Her current research interests include computational intelligence, swarm intelligent optimization, system optimization, and numerical simulation of control systems.

YUANBIN MO received the M.S. degree in mathematics from Guizhou University, Guiyang, China, in 2001, and the Ph.D. degree in control theory and control engineering from Zhejiang University, Hangzhou, China, in 2007. He is currently a Professor with Guangxi University for Nationalities. His research interests include computation intelligence, system optimization, system optimization, and numerical simulation of control systems.

DONGXUE BAO received the B.S. degree in Internet of Things from Tangshan Normal University, Tangshan, China, in 2020. Her current research interests include natural language processing and text summarization.

RONG GONG received the B.S. degree in computer science and technology from Aba Teachers University, Aba, China, in 2019. His current research interests include rough set and swarm intelligent optimization.