Research Article

An Enhanced Slime Mould Algorithm and Its Application for Digital IIR Filter Design

Xiaodan Liang ¹, Dong Wu ¹, Yang Liu ², Maowei He ¹ and Liling Sun ¹

¹School of Computer Science and Technology, Tiangong University, Tianjin 300387, China

²Liaoning Inspection, Examination and Certification Centre, Shenyang 110000, China

Correspondence should be addressed to Liling Sun; sunliling2019@outlook.com

Received 6 October 2021; Accepted 10 November 2021; Published 10 December 2021

Academic Editor: Juan L. G. Guirao

Copyright © 2021 Xiaodan Liang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the past few decades, metaheuristic algorithms (MA) have been developed tremendously and have been successfully applied in many fields. In recent years, a large number of new MA have been proposed. Slime mould algorithm (SMA) is a novel swarm-based intelligence optimization algorithm. SMA solves the optimization problem by imitating the foraging and movement behavior of slime mould. It can effectively obtain a promising global optimal solution. However, it still suffers some shortcomings such as the unstable convergence speed, the imprecise search accuracy, and incapability of identifying a local optimal solution when faced with complicated optimization problems. With the purpose of overcoming the shortcomings of SMA, this paper proposed a multistrategy enhanced version of SMA called ESMA. The three enhanced strategies are chaotic initialization strategy (CIS), orthogonal learning strategy (OLS), and boundary reset strategy (BRS). The CIS is used to generate an initial population with diversity in the early stage of ESMA, which can increase the convergence speed of the algorithm and the quality of the final solution. Then, the OLS is used to discover the useful information of the best solutions and offer a potential search direction, which enhances the local search ability and raises the convergence rate. Finally, the BRS is used to correct individual positions, which ensures the population diversity and enhances the overall search capabilities of ESMA. The performance of ESMA was validated on the 30 IEEE CEC2014 functions and three IIR model identification problems, compared with other nine well-regarded and state-of-the-art algorithms. Simulation results and analysis prove that the ESMA has a superior performance. The three strategies involved in ESMA have significantly improved the performance of the basic SMA.

1. Introduction

Optimization problems are involved in many fields. With the expansion of production scale, optimization problems are becoming more and more tough [1]. Most of the optimization problems in the real world are NP-hard, and these problems cannot be solved by precise algorithms [2]. Conventional accurate gradient-based algorithms are helpless when facing with noncontinuous, nondirected, multipeak, and dynamic problems [3, 4]. The MA is a kind of bionic approximation algorithm and an effectual method to solve such convoluted optimization problems. For the past several decades, MAs have made tremendous research progress. Various algorithms inspired by nature have been proposed. Well-known MA includes Genetic Algorithm (GA) [5], Particle Swarm Optimization (PSO) [6],

Differential Evolution (DE) [7], Bacterial Foraging Optimization (BFO) [8], Artificial Bee Colony (ABC) Algorithm, Biogeography-based Optimization (BBO) [9], and Gravitational Search Algorithm (GSA) [10]. The new methods proposed in recent years are Grey Wolf Optimizer (GWO) [11], Moth Flame Optimization (MFO) [12], Cuckoo Search Algorithm (CSA) [13], Sine Cosine Algorithm (SCA) [14], Harris Hawks Optimization (HHO) [15], Whale Optimization Algorithm (WOA) [16], Political Optimizer (PO) [17], Slime Mould Algorithm (SMA) [18], Aquila Optimizer (AO) [19], and Hunger Games Search (HGS) [20].

SMA was proposed by Li et al. in 2020 and inspired by the foraging and movement behavior of slime mould in nature. Slime mould is a single-celled amoeba without brain and nerves. Based on its veins, its foraging behavior becomes efficient. Under the oscillating model of veins, it can obtain

the highest concentration of food, and this food position represents the optimal solution to global optimization problems. The performance of SMA has been confirmed in [18], and it has better performance than many well-regarded and state-of-the-art algorithms. SMA has a simple and flexible structure. Based on this flexible structure, it is easy to further improve its performance.

Due to the excellent performance of SMA, the appearance of SMA has attracted the attention of many scholars. There have been many researches and applications about SMA. Gupta et al. in 2021 [21] used SMA to solve the problem of proton exchange membrane fuel cell model parameter estimation, and the experimental results show that, compared with other algorithms, SMA has prediction results that are more consistent with actual experimental results. Yıldız [22] used SMA in the optimization design of auto components for the first time. In [23], Zobaa et al. used SMA to optimize the design of a double-tuned filter to make the performance of the power system better and reduce the impact of harmonics in the power system. Moreover, the SMA was also used in different fields: to optimize extreme learning machine parameters [24], to evaluate structural damage and gauging structural health [25], to optimize and coordinate the directional overcurrent relays in meshed power networks [26], to optimize the fuzzy controller [27], to optimize the photovoltaic hosting capacity [28], and to forecast urban water demand [29].

Although the excellent performance of SMA has been confirmed in many practical applications, SMA still fails in keeping the balance between exploration and exploitation like many other MAs. In particular, when facing with complex problems, it would accidentally be trapped at the local minimum value, and the convergence speed may decline. To overcome such deficiencies, an enhanced SMA with chaotic initialization strategy (CIS), orthogonal learning strategy (OLS), and boundary reset strategy (BRS), named ESMA, is proposed.

Chaos is a nonlinear phenomenon which is ubiquitous in nature that has the characteristics of pseudorandomness and ergodicity and is very sensitive to original conditions. Even a tiny discrepancy in original conditions will lead to significant differences in the long-term behavior [30]. In the past few decades, chaos control has been used in a large number of fields, including parameter optimization, feature selection, and chaos control [3, 31]. With the development of swarm intelligence technology over recent years, chaos has become a very popular strategy for improving MA. The improvement of metaheuristic algorithms mainly includes three aspects: chaotic parameter distribution, chaotic initialization population, and chaotic local search (CLS) strategy. For instance, Kaur et al. [32] and Ibrahim [33] et al. used the chaotic map to adapt the exploration and exploitation of WOA and SSA, respectively. Chou and Truong [34] and Ma et al. [35], respectively, used chaotic maps to generate a highly diverse initial population of their algorithm. Ibrahim et al. used the chaotic map to process the original solution to raise the convergence of the GWO [36]. Zhao et al. [37] used the CLS to enhance the performance of SSA. Chen et al. [38] proposed an enhanced BFO with CLS.

And in [39], Sayed et al. proposed a CCSA with the chaotic search method and used it for feature selection problems.

OLS is also an improvement strategy that arouses the interest of many scholars. It is based on orthogonal experimental design (OED). OED can find the best level combination of various factors with fewer experiments [31, 40]. Under the premise of ensuring effectiveness, compared with the exhaustive method, it greatly improves efficiency and saves resources. Based on these characteristics of OED, a large number of researchers use OED to boost the performance of MA, such as HHO combined with orthogonal learning [41], FOA with orthogonal learning schemes [42], DE combined with OED [43], CSA with orthogonal learning strategy [44], MFO combined with orthogonal learning strategy, and Broyden–Fletcher–Goldfarb–Shanno [45].

For the ESMA proposed in this article, three efficient mechanisms, including CIS, OLS, and BRS, are integrated into the basic SMA. Firstly, the integration of CIS helps the ESMA initialize a group of initial solutions with diversity, which may speed up the convergence speed of SMA, improve the quality of the final solutions, and avoid falling into a local optimal solution. Secondly, the OLS was used on the two currently ranked best individuals to dig out more useful information, which could predict a better best individual for calculating the weights of each slime mould. This will make the population develop with a potential direction, strengthen the local search ability, and raise the convergence speed and accuracy of the ESMA. Additionally, in each iteration of the ESMA, the BRS is used to adjust the position of each search agent after all their position is updated, which ensures that all search agents are in the search region. The BRS fostering the diversity of the population avoids the population from concentrating too much at the boundary and falling into a local optimum. It is worth noting that the application of the above three strategies does not change the main structure of the basic SMA, but only strengthens the local search and group diversity. This improvement inherits the advantages of SMA and overcomes the drawbacks of SMA and greatly improves the performance of SMA.

To assess the performance of ESMA, the classic IEEE CEC 2014 benchmark functions are used as numerical tests, and three IIR model identification cases are used as real-world application tests to verify the universality of ESMA. The experimental results reveal that fairly compared ESMA with other nine well-accepted and state-of-the-art algorithms, ESMA can obtain overwhelming or very competitive performance. This confirms that the ESMA based on the three improvement strategies has a comprehensive improvement over the original SMA.

In order to improve the performance of the basic SMA algorithm and overcome its slow convergence speed and low convergence accuracy, a new upgraded version of SMA (ESMA) is proposed in this article. At the same time, the proposed ESMA algorithm would be used to solve the CEC2014 optimization problems and the IIR model identification optimization problem. The main contributions of our work are briefly summarized as follows:

- (i) Based on the original SMA algorithm, an improved SMA algorithm with orthogonal learning capability is proposed. At the same time, a chaotic initialization strategy and a boundary reset strategy are introduced to improve the performance of the improved SMA.
- (ii) The improved SMA and ten advanced metaheuristic algorithms including the original SMA are applied to solve the CEC2014 problem, and their performance is compared and analyzed based on the experimental results.
- (iii) The improved SMA algorithm is used to solve the IIR model identification problem, and the simulation results are compared with ten other advanced algorithms. And the practicability and efficiency of the proposed algorithm are proven.

The structure of this paper is presented as follows. The basic description of SMA is briefly introduced in Section 2. The proposed ESMA and three improvement strategies are illustrated in great detail in Section 3. In Section 4, a series of experiments were done to assess the performance of ESMA on IEEE CEC2014 test suite and three IIR model identification cases, and the simulation results and analysis are displayed and discussed. Finally, Section 5 summarizes this study and puts forward some enlightenment for future directions.

2. Original SMA

The key idea of SMA is to utilize the foraging behavior of slime mould. The foraging behavior of slime mould can be summarized into three morphotypes: approach food, wrap food, and grabble food. Relying on the propagation wave arising from the biological oscillator to change the cytoplasmic flow in the vein, slime mould can approach high-quality food, then surround the food and use enzymes to digest it, or find other higher-quality food through adaptive transformation. A brief description of SMA is provided as follows.

The position update of slime mould is shown as follows:

$$\overrightarrow{X}(t+1) = \begin{cases} \text{rand} \cdot (U - L) + L, & \text{rand} < z, \\ \overrightarrow{X}_b(t) + \overrightarrow{vb} \cdot (W \cdot \overrightarrow{X}_A(t) - \overrightarrow{X}_B(t)), & \text{rand} > z \text{ and } r < k, \\ \overrightarrow{vc} \cdot \overrightarrow{X}(t), & \text{rand} > z \text{ and } r \geq k, \end{cases} \quad (1)$$

where $\overrightarrow{X}(t+1)$ and $\overrightarrow{X}(t)$ indicate the position of slime mould in $(t+1)$ th iteration and t th iteration, respectively, $\overrightarrow{X}_A(t)$ and $\overrightarrow{X}_B(t)$ represent two search agents randomly selected from the population at t th iteration, respectively, and $\overrightarrow{X}_b(t)$ denotes the slime mould with the best fitness value in the current iteration, that is, the search agent with the highest food concentration. U and L represent the upper bound and lower bound of the search space, respectively, rand and r mean the random value in the interval of $[0, 1]$. Here z is a very important constant parameter because its value affects the balance between the exploration and

exploitation of SMA. Here k is another essential parameter, which is calculated as follows:

$$k = \tanh[F(i) - aF], \quad (2)$$

where $F(i)$ stands for the fitness of the i th search agent and aF denotes the individual with best fitness value obtained in history.

The values of vectors $\overrightarrow{vc}$ and $\overrightarrow{vb}$ oscillate in the range of $[-1, 1]$ and $[-a, a]$, respectively. The specific expression of $\overrightarrow{vc}$ is shown in equation (3), and the expression of parameter a of $\overrightarrow{vb}$ is shown in equation (4).

$$\overrightarrow{vc}(j) = 1 - \left(\frac{t}{\max_t} \right), \quad (3)$$

$$a = \arctan h \left(- \left(\frac{t}{\max_t} \right) + 1 \right), \quad (4)$$

where $\max_t$ stands for the maximum number of iterations. $\overrightarrow{vc}$ and $\overrightarrow{vb}$ will gradually approach 0 with iteration, which play two major roles in the algorithm. Because the two vectors $\overrightarrow{vc}$ and $\overrightarrow{vb}$ are used to imitate the behavior of slime mould in the process of foraging, they will not only exploit the areas with high concentration food, but also explore other areas to find other high concentration food sources. Through the adaptive control of $\overrightarrow{vc}$ and $\overrightarrow{vb}$, SMA can more effectively deal with the balance between global exploration and local exploitation and avoid the SMA falling into a local optimal solution.

$\overrightarrow{W}$ in equation (1) represents corresponding weight, and its calculation formula is as follows:

$$\overrightarrow{W}(\text{Index}(i)) = \begin{cases} 1 + r \cdot \log \left(\frac{bF - F(i)}{bF - wF} + 1 \right), & \text{condition,} \\ 1 - r \cdot \log \left(\frac{bF - F(i)}{bF - wF} + 1 \right), & \text{others,} \end{cases} \quad (5)$$

where bF and wF denote two search agents with best fitness value and worst fitness value obtained in the present iterative procedure, respectively, condition means the individuals' fitness ranks of the first half of the swarm, and here $\text{Index} = \text{sort}(F)$. Equation (5) visually imitates the positive and negative feedback between the vein width of slime mould and the food concentration.

The pseudocode of SMA is illustrated in Algorithm 1.

3. Proposed ESMA

This study proposes a new algorithm with orthogonal learning strategy (OLS), chaotic initialization strategy (CIS), and boundary reset strategy (BRS), which is called ESMA. In the following sections, the main framework of the ESMA will be described first, and then three improved strategies will be introduced.

3.1. Framework of ESMA. In this paper, the orthogonal learning strategy (OLS), chaotic initialization strategy (CIS), and boundary reset strategy (BRS) are embedded into basic

```

(1) Initialize the slime mould position  $X^i$ ,  $i = 1, 2, \dots, n$ .
(2) While ( $t < \max\_t$ ) do for each iteration
(3)   Calculate the fitness of each slime mould
(4)   Update  $bF$ ,  $wF$ , and  $X_b$ 
(5)   Update  $W$  using equation (5)
(6)   for each individual ( $X^i$ )
(7)     Update  $p$ ,  $vb$ ,  $vc$ 
(8)     Update the position of each individual using equation (1)
(9)   end for
(10)  Amend the position of each individual based on the upper and lower bounds
(11) end while
(12) Return  $bF$  and  $X_b$ 

```

ALGORITHM 1: Pseudocode of SMA.

SMA, without major changes in the structure of SMA. Firstly, CIS is used in the initial stage of the ESMA to create a high-quality population for guaranteeing the initial population diversity, the high convergence rate, and the excellence of the final solution. Then, in the stage of calculating the weight of each slime mould, OLS is used on the two currently ranked best individuals that may dig out more useful information to predict a better best individual for calculating the weights of each slime mould. Moreover, after the position of each slime mould is updated in each loop, BRS will be used to detect and correct individuals who are beyond the boundary, ensuring that the position of each slime mould is restricted to the search region, instead of correcting them to the boundary of search region like the original SMA algorithm. The use of BRS helps to enhance the population diversity of slime mould. The pseudocode and main structure of ESMA are shown in Algorithm 2 and Figure 1, and the CIS, OLS, and a detailed description about BRS will be shown in the following sections.

3.2. Chaotic Initialization Strategy (CIS). Chaos has the characteristics of ergodicity, pseudorandomness, and sensitivity to initial conditions. Therefore, in the past few decades, chaos theory has been used in many fields such as parameter optimization, feature selection, and chaos control [3]. In recent years, chaotic mapping has also become a widely popular method to improve metaheuristic algorithms, such as chaotic parameter control [32, 33], chaotic initialization [34–36], and chaotic local search [37–39]. In this study, the logistic chaos map is utilized as a chaotic sequence. As one of the simplest and widely used chaotic sequences, the logistic map has the following expression:

$$x_{i+1} = \eta \cdot x_i (1 - x_i), \quad (6)$$

where η is a control parameter, when $\eta = 4$, the logic sequence appears chaotic, x_i denotes the chaotic sequence value of the i th slime mould, $x_0 \in (0, 1)$, and x_0 is used to create an initial population of slime mould.

A good initial population can raise the population diversity, speed up the convergence, reduce the possibility of falling into a local infinitesimal value, and enhance the

FIGURE 1: The flowchart of ESMA.

solution quality. For most metaheuristic algorithms, the initial population is randomly and uniformly distributed. This method has the drawbacks of not fast convergence


```

(1) Initialize the slime mould position  $X^i$ ,  $i = 1, 2, \dots, n$ .
(2) While ( $t < \max\_t$ ) do for each iteration
(3)   Calculate the fitness of each slime mould
(4)   Update  $bF$ ,  $wF$ , and  $X_b$ 
(5)   Update the best agent by OLS
(6)   Update the  $W$  using equation (5)
(7)   for each individual ( $X^i$ )
(8)     Update  $p$ ,  $vb$ ,  $vc$ 
(9)     Update the position of each individual using equation (1)
(10)  end for
(11)  Verify the position of each individual using equation (12)
(12) end while
(13) Return  $bF$  and  $X_b$ 

```

ALGORITHM 2: Pseudocode of ESMA.

speed and low population diversity, which may make the algorithms slip into a local minimum [2, 46]. And the existing research shows that the method with population chaos initialization is superior to the original method [47]. Therefore, in this article, a logical chaotic value is used to create an initial population, and the chaotic disturbance is added to the initial population to improve the initial population diversity. The formula is expressed as follows:

$$X_{i,j}^* = \mu_{i,j} \cdot X_{i,j}, \quad (7)$$

where $\mu_{i,j}$ is the j th value of the logistic sequence of the i th slime mould and $X_{i,j}^*$ is the i th slime mould's position with chaotic disturbance.

3.3. Orthogonal Learning Strategy (OLS). As an efficient experimental tool, orthogonal experiments have been widely used in many fields. In terms of improving metaheuristic algorithms, orthogonal learning has also been favored by many researchers. The improvements to PSO [40], BSO [48], SCA [49], FFO [42], MFO [45], HHO [41], and other algorithms have achieved significant results. OLS can vastly improve the local exploitation ability of the optimization method. In order to not alter the flexible structure and direction of the original SMA and further improve the local exploitation ability of the MA method simultaneously, OLS is employed into the SMA. The OLS is used to better balance the global and local search of the original SMA, so as to get a more promising global optimal solution.

In each iteration of SMA algorithm, before calculating the corresponding weight $\vec{W}$ (equation (5)), two individuals with the best fitness value (Index (1) and Index (2)) are selected, and the search information of these two excellent individuals is used to predict a better individual with the best fitness value (Index(1*)). If we exhaustively calculate all combinations of Index (1) and Index (2) for the optimal individual Index(1*), 2^D (D represents the problem dimension) trials are needed. Due to the exponential

complexity, this is unrealistic in real experiments. In this paper, we use OLS to predict a relatively good new individual Index(1*) from Index (1) and Index (2) through as few experiments as possible. The factors are equal to the problem dimension, and the number of levels is 2.

OLS is based on orthogonal experiment design (OED), which mainly includes two parts, orthogonal array (OA) and factor analysis (FA). The following will use a simple case to explain OED from two aspects of OA and FA.

3.3.1. Orthogonal Array (OA). OA is the basic and crucial part of OED. It is a prediction table that determines how many experiments will be performed in order to find a better candidate combination. As a graphic method of constructing mutually orthogonal Latin squares, OA can ensure a balanced comparison of levels of any factor. An OA with K factors and Q levels can be expressed as $L_M(Q^K) = [l_{ij}]_{M \times K}$, where l_{ij} is the number of levels of the j th factor in the i th combination, L indicates the OA, and M denotes the number of combinations of test example. The relationship between these numbers satisfies three constraints: (1) Q is a prime number; (2) $M = Q^J$; and (3) J must be a positive integer and meets the requirements:

$$K \leq \left(\frac{Q^J - 1}{Q - 1} \right). \quad (8)$$

In order to further illustrate OA, take an example of the conversion ratio in chemistry. In an experiment, there are three factors affecting conversion ratio, and each factor has three different levels. Factors and levels information are shown in Table 1. In this problem, if we exhaustively test all combinations of three factors and three levels, $3^3 = 27$ trials are needed. However, it can be seen from equation (9) that only 9 experimental combinations need to be considered when using OA, which significantly reduces testing times and improves the experimental efficiency.

$$L_9(3^4) = [l_{ij}]_{9 \times 4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 \\ 1 & 3 & 3 & 3 \\ 2 & 1 & 2 & 3 \\ 2 & 2 & 3 & 1 \\ 2 & 3 & 1 & 2 \\ 3 & 1 & 3 & 2 \\ 3 & 2 & 1 & 3 \\ 3 & 3 & 2 & 1 \end{bmatrix}. \quad (9)$$

3.3.2. Factor Analysis (FA). As another vital part of OED, FA will evaluate the influence of various factors according to the results of the experimental based on OA and then summarizes the optimal level combination. In the working process of FA, the following formula is mainly used:

$$E_{kq} = \frac{\sum_{m=1}^M y_m \cdot S_{mkq}}{\sum_{m=1}^M S_{mkq}}, \quad (10)$$

where y_m indicates the experimental result of the m th ($1 \leq m \leq M$) combination and E_{kq} means the effect of the q th ($1 \leq q \leq Q$) level of the k th ($1 \leq k \leq K$) factor. It is worth noting that, $S_{mkq} = 1$, if m th experiment embraces the q th level of the k th factor; otherwise, $S_{mkq} = 0$.

Assuming that the experimental results of 9 combinations obtained according to equation (9) are shown in Table 2, the FA results according to these experimental results are shown in the last three rows in Table 2. It can be clearly observed from Table 2 that in factor A, the three E_{Aq} are $E_{A1} = 41$, $E_{A2} = 48$, and $E_{A3} = 61$, respectively. Among them, E_{A3} is the largest, that is, the conversion ratio obtained at the reaction temperature of 90°C is the highest, which is the best reaction temperature among the three temperatures. Similarly, according to the results of FA in Table 2, the best level of the other two factors can be analyzed. Hence, the best levels combination of various factors that can be predicted is A3, B2, C2.

3.4. Boundary Reset Strategy (BRS). For the swarm-based algorithm, in the process of solving the problem, it is often encountered that the search agent exceeds the search region. The usual practice is to reset the position value of the individual that exceeds the search region to the boundary value, as shown in the following equation:

$$\vec{X}_j^* = \begin{cases} \text{rand} \cdot (U - L) + L, & \text{if } \vec{X}_j^* < L, \\ \text{rand} \cdot (U - L) + L, & \text{if } \vec{X}_j^* > U. \end{cases} \quad (11)$$

The original SMA also uses equation (11) to solve the problem that slime mould individuals may move outside the search region during the algorithm iteration.

However, this method has obvious shortcomings. When a great quantity of individuals exceed the search region at the

same time or when there is a local minimum near the boundary of the search region, the diversity of the population will be reduced, resulting in the algorithm may get trapped in a local optimal solution on the boundary.

This article uses a boundary reset technique called BRS to deal with the above problems. BRS has also been applied in articles [44, 50–52] and has been proven effective in ensuring population diversity and preventing algorithms from trapping into local optima. The expression of BRS is shown as follows:

$$\vec{X}_j^* = \begin{cases} \text{rand} \cdot (U - L) + L, & \text{if } \vec{X}_j^* < L, \\ \text{rand} \cdot (U - L) + L, & \text{if } \vec{X}_j^* > U, \end{cases} \quad (12)$$

where $\vec{X}_j^*$ denotes the j th dimension value of the position of slime mould.

4. Experiments

In this section, the performance of ESMA was evaluated on 30 IEEE CEC2014 benchmark functions and 3 IIR model identification optimization cases. Several well-regarded and state-of-the-art algorithms are used to compare with the proposed ESMA. The parameter setting of all used algorithms is shown in Table 3.

Moreover, nonparametric statistical analysis is extensively used to fairly analyze the performance of MA [53]. In this article, the famous Wilcoxon signed-rank test is used with a 0.05 significance level to compare the performances of the ESMA and other algorithms. In these tables, the p value of the Wilcoxon test means the significance level. When the p value is less than 0.05, it means there is a significant performance difference between ESMA and other algorithms. Otherwise, there is no significant difference between ESMA and compared methods. The symbol B/T/W indicates the performance of ESMA shows significantly better than, tied with, or worse than other methods under the significance level of 5%.

In our experimental work, for a fair comparison, the execution condition of all experiments is kept constant as follows:

- (i) CPU: Intel Core i7-8750H (2.20 GHz)
- (ii) RAM: 16.0 GB
- (iii) System: Windows 10
- (iv) Software: Matlab R2018b

4.1. IEEE CEC2014 Benchmark Functions Optimization Problem. In this section, the performance of ESMA was evaluated on IEEE CEC2014 test suite, and the performance of nine other algorithms (SMA, AO, BBO, GWO, HGS, HHO, MFO, SCA, and WOA) is compared with that of ESMA. To attain the unbiased result, each algorithm including ESMA is independently tested 50 times for each problem. The population size N and search region dimension D are all set to 30, and the maximum number of function evaluations ($MaxFes$) is set to $10000 * D$.

TABLE 1: Factors and levels information in the case.

Levels	Factor A: temperature (°C)	Factor B: time (minutes)	Factor C: alkali (%)
L_1	80	90	5
L_2	85	120	6
L_3	95	150	7

TABLE 2: Result of factor analysis.

Levels	Factor A: temperature	Factor B: time	Factor C: alkali	Results
C_1	(1) 80°C	(1) 90 minutes	(1) 5%	$F_1 = 31$
C_2	(1) 80°C	(2) 120 minutes	(2) 6%	$F_2 = 54$
C_3	(1) 80°C	(3) 150 minutes	(3) 7%	$F_3 = 38$
C_4	(2) 85°C	(1) 90 minutes	(2) 6%	$F_4 = 53$
C_5	(2) 85°C	(2) 120 minutes	(3) 7%	$F_5 = 49$
C_6	(2) 85°C	(3) 150 minutes	(1) 5%	$F_6 = 42$
C_7	(3) 90°C	(1) 90 minutes	(3) 7%	$F_7 = 57$
C_8	(3) 90°C	(2) 120 minutes	(1) 5%	$F_8 = 62$
C_9	(3) 90°C	(3) 150 minutes	(2) 6%	$F_9 = 64$
Levels	Results of FA			
L_1	$E_{A1} = (y_1 + y_2 + y_3)/3 = 41$	$E_{B1} = (y_1 + y_4 + y_7)/3 = 47$	$E_{C1} = (y_1 + y_6 + y_8)/3 = 45$	
L_2	$E_{A2} = (y_4 + y_5 + y_6)/3 = 48$	$E_{B2} = (y_2 + y_5 + y_8)/3 = 55$	$E_{C2} = (y_2 + y_4 + y_9)/3 = 57$	
L_3	$E_{A3} = (y_7 + y_8 + y_9)/3 = 61$	$E_{B3} = (y_3 + y_6 + y_9)/3 = 48$	$E_{C3} = (y_3 + y_5 + y_7)/3 = 48$	
FA result	A3	B2	C2	

TABLE 3: The parameter setting of algorithms.

Algorithm	Year and literature	Parameter setting
SMA	2020 [18]	$z = 0.03$
AO	2021 [19]	$\alpha = 0.1; \delta = 0.1; U = 0.00565; \omega = 0.005$
BBO	2008 [9]	$Mu = 0.005; \mu = 0.8$
GWO	2014 [11]	$a = [2, 0]$
HGS	2021 [20]	$l = 0.08$
HHO	2019 [15]	$E \in [0, 2]$ (energy of a rabbit)
MFO	2015 [12]	$b = 1; t = [-1, 1]; a \in [-1, -2]$
SCA	2016 [14]	$A = 2$
WOA	2016 [16]	$a_1 = [2, 0]; a_2 = [-2, -1]; b = 1$
ESMA		$z = 0.03$

4.1.1. Benchmark Function. In our experiments, the IEEE CEC2014 problem set is used to evaluate the performance of ESMA. As a classical test set, IEEE CEC2014 composite benchmark functions set contains 30 functions and can be divided into four types (F1–F3 are unimodal functions, F4–F16 are multimodal functions, F17–F22 are hybrid functions, and F23–F30 belong to composition functions). These types of functions simulate most engineering problems to a certain extent, so IEEE CEC2014 is used as a large number of algorithm performance evaluation test sets. More details of IEEE CEC2014 can be obtained from [54].

4.1.2. Experimental Results and Analysis. The experimental results obtained by ESMA and nine compared methods on IEEE CEC2014 problems are shown in Table 4, and the optimal results obtained for a test function are highlighted. From Table 4, obviously, ESMA outperformed all other compared algorithms in almost all functions. The proposed ESMA not only obtained the best mean value (Mean), but also has the minimum standard deviation value (Std) for most test functions.

As shown in Table 4, ESMA beats the basic SMA and HHO on all test functions except on F23, F24, F25, F27, and F28 with the same performance. With respect to AO, ESMA achieves superior results on 26 functions. At the same time, it gets inferior results on F29 and the same results on F23, F25, and F28. Compared with BBO, ESMA could get better results on all of the benchmark functions except on F4 and the worse results on F18. Compared with HGS, ESMA achieves better results on 26 functions, worse results on F30, and the same results on F23, F25, F27, and F28. Finally, ESMA beats the GWO, MFO, SCA, and WOA for all of the test functions. Based on the results of the Wilcoxon signed-rank test in the last row of Table 4, ESMA performs superior to or similar to the comparison algorithms in performance on all functions, and there is no case that ESMA is worse than any peers on any functions, which reveals that ESMA has superior capabilities. As Table 4 shown, ESMA shows a significant improvement on SMA.

In Table 5, the p values of the Wilcoxon test of ESMA and other competitors against IEEE CEC2014 benchmark functions are shown. Clearly, for most cases, p values are less than 0.05. This presents that the performance of ESMA is statistically superior to other competitors on most test suite. Also, these values illustrate that the improvements of ESMA are significant.

Additionally, box-whisker diagrams can clearly show the structure of the experimental results obtained through 30 tests. As box-whisker diagrams shown in Figure 2, obviously, the structure of results of ESMA was better than the other nine compared algorithms on almost all functions. It is worth noting that although SMA has also got good results, on F7, F18, and F19, ESMA has fewer outliers than SMA, which shows the effectiveness of the improvement.

Furthermore, the linear trend chart in Figure 3 can be used to compare the convergence performance of the comparison methods and ESMA more intuitively. From

Figure 2: Continued.

Figure 2: Continued.

FIGURE 2: The distribution of the fitness of ESMA and compared algorithms on 30 benchmark functions.

Figure 3, the convergence curves of ESMA are superior to other compared methods on almost all test functions. The convergence curves of ESMA indicated that, in most cases, ESMA achieves faster convergence rates and better convergence precision than other competitors, which further confirms the effectiveness of improvement.

4.2. IIR Model Identification Optimization Problem. In this section, the IIR model identification problem is solved to verify the universality of the proposed ESMA. Firstly, the problem of IIR system modeling is defined. Then, ESMA will be used to identify three unknown IIR systems, and the performance of ESMA will be compared with other nine well-regarded and state-of-the-art algorithms (SMA, AO, BBO, GWO, HGS, HHO, MFO, SCA, and WOA). The Wilcoxon test and convergence curves will be used to compare the performance of all methods. In this experimental part, in order to ensure fairness, each optimization method is independently run 50 times in each case, and the population size (N) and maximum evaluation times (MaxFes) are set to 25 and $3000 \cdot D$, respectively, where D indicates the dimension of case.

4.2.1. Definition of the Problem. As a system identification problem, the digital IIR filter model identification is to find the exact parameters of mathematical model of an unknown IIR model [55]. This problem is like a black box problem. When using the optimization method to optimize the model parameters adaptively, the output error between the unknown IIR model and the designed IIR model is minimized

to obtain an optimal unknown IIR model. This process is shown in Figure 4.

In the IIR system, the general form of the relationship between input and output is expressed as

$$y(n) = \sum_{i=1}^N a_i \cdot y(n-i) + \sum_{j=0}^M b_j \cdot x(n-i), \quad (13)$$

where $x(n)$ and $y(n)$ stand for the n th input and output signal of the IIR filter, N and M mean the numerator and denominator order of the IIR system, respectively, and a_i and b_j are coefficients of IIR filter which define the poles and zeroes of IIR system, respectively.

Equation (13) can also be expressed in the form of Z transformation:

$$\frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}}. \quad (14)$$

A set of specific parameters of an IIR filter system is unknown. Hence, how to identify an IIR model can be considered an optimization problem. According to Figure 4, the error of the identification process between the output of the actual IIR filter and the output of the estimated model is $e(n) = d(n) - y(n)$. Thus, in this article, we use the mean square error of input and output as the cost function ($f(\theta)$), and the expression of $f(\theta)$ is as follows:

$$f(\theta) = \frac{1}{K} \sum_{n=1}^K (e(n))^2, \quad (15)$$

where K stands for the length of samples used to calculate the cost function and θ indicates the series of unknown exact

Figure 3: Continued.

Figure 3: Continued.

Figure 3: Continued.

FIGURE 3: Convergence curves of ESMA and competitors on 30 benchmark functions.

FIGURE 4: Block diagram for digital IIR model identification process.

parameters of the IIR mathematical model. The goal of IIR model identification is to find an optimal θ^* to minimize $f(\theta)$.

In our experimental work, three widely used IIR identification cases are selected, which were labeled as IIR1, IIR2, and IIR3. The transfer functions of these three IIR models are demonstrated in Table 6. For IIR1, the plant is a superior order of the sixth-order system, and the IIR filter to be designed is the fourth order. The second case is a third-order plant modeled using third order. In the 3rd case, the unknown plant is fifth order, and the IIR filter is the fourth order.

4.2.2. Simulation Results of IIR System Identification. The results of solving three IIR system identification cases and comparing ESMA with nine other algorithms are shown in Table 7. The mean (Mean), best (Best), worst (Worst), and median (Median) $f(\theta)$ value and the standard deviation (Std) are all presented in Table 7. And at the end of Table 7, the symbol *B/T/W* indicates the performance of ESMA shows significantly better than, tied with, or worse than other competitors under the significance level 5%. Moreover, the p value results of the Wilcoxon test on 3 IIR system identification cases are tabulated in Table 8. Figures 5 and 6 show the solution structure and mean optimization convergence curves.

As shown in Table 7, from the three IIR model identification cases tests, ESMA outperforms the compared algorithms in two cases. In the IIR1 and IIR3 cases, ESMA is significantly better than other algorithms, obviously. Although ESMA is not comparable to BBO, MFO, and GWO in the IIR2 case, the results of ESMA are still very competitive and rank at the forefront and are significantly outperformed the other 7 competitors. These are highlighted in Table 7.

From Table 8, in Case 1 and Case 3, the p value is almost less than 0.05, and in Case 2, the p value is mostly less than 5%, which shows that ESMA is significantly superior to other compared methods in most cases. As can be seen from Figure 5, compared with other methods, the proposed ESMA algorithm has a faster convergence speed. For IIR1 and IIR3, ESMA still has strong search ability and obtains the best solution while other algorithms get trapped into local optimal. For IIR2, although ESMA does not obtain the first, it still obtained an acceptable convergence speed. Moreover, from Figure 6, ESMA has the best solution structure in IIR1 and IIR3 and has very competitive results in IIR2. Overall, in most cases, the solution structure and convergence speed of ESMA are very excellent in the three cases.

ESMA not only has superior average convergence speed and precision in dealing with IIR1 and IIR3, but also is far outperformed other algorithms in the structure of the

TABLE 4: Experimental results of ESMA and compared algorithms on IEEE CEC2014 benchmark functions.

Fn	SMA	AO	BBO	GWO	HGS	HHO	MFO	SCA	WOA	ESMA
F1	Mean	1.9120E+06	1.4402E+07	1.6202E+06	6.3242E+07	1.7173E+06	6.7229E+06	1.1603E+08	2.2637E+08	2.9210E+07
	Std	7.8209E+05	6.4693E+06	6.2846E+05	5.9117E+07	1.2117E+06	3.0394E+06	1.2049E+08	6.5863E+07	1.1147E+07
F2	Mean	1.3759E+04	7.0109E+06	2.3418E+04	2.3902E+09	9.3835E+03	3.4000E+04	1.3596E+10	1.7385E+10	4.7709E+06
	Std	1.1435E+04	2.0691E+06	9.3370E+03	2.1733E+09	1.2449E+04	1.6166E+04	8.0911E+09	3.4722E+09	8.4362E+06
F3	Mean	4.4637E+02	2.8559E+04	3.8454E+03	3.0093E+04	8.2327E+03	3.1413E+03	8.9842E+04	3.7931E+04	3.9656E+04
	Std	2.3992E+02	5.9321E+03	3.7898E+03	8.9252E+03	6.6533E+03	1.2000E+03	3.8297E+04	6.8467E+03	2.3843E+04
F4	Mean	5.0004E+02	5.6391E+02	4.7794E+02	6.7051E+02	4.9644E+02	5.2961E+02	1.5350E+03	1.3434E+03	5.8897E+02
	Std	4.7647E+01	4.6737E+01	4.0650E+01	1.2290E+02	3.0703E+01	3.7843E+01	1.2357E+03	1.8677E+02	5.8103E+01
F5	Mean	5.2077E+02	5.2080E+02	5.2002E+02	5.2093E+02	5.2005E+02	5.2006E+02	5.2025E+02	5.2093E+02	5.2008E+02
	Std	2.5047E-01	1.2216E-01	1.3419E-02	1.1666E-01	5.5660E-02	7.2479E-02	1.7185E-01	5.7798E-02	1.6405E-01
F6	Mean	6.1543E+02	6.2332E+02	6.1927E+02	6.1429E+02	6.1822E+02	6.3218E+02	6.2435E+02	6.3363E+02	6.3467E+02
	Std	3.2764E+00	3.2618E+00	4.1304E+00	2.9860E+00	3.2013E+00	3.5324E+00	2.9275E+00	2.3657E+00	2.6062E+00
F7	Mean	7.0031E+02	7.0107E+02	7.0012E+02	7.1799E+02	7.0003E+02	7.0014E+02	8.1893E+02	8.4299E+02	7.0103E+02
	Std	8.3392E-02	2.2664E-02	6.2973E-02	1.5940E+01	4.1512E-02	4.2351E-02	6.8116E+01	2.5551E+01	6.9151E-02
F8	Mean	8.1234E+02	9.1398E+02	8.5940E+02	8.8119E+02	8.0631E+02	8.9009E+02	9.5139E+02	1.0441E+03	9.9182E+02
	Std	3.6821E+00	2.0654E+01	1.5055E+01	1.9031E+01	5.6655E+00	1.6673E+01	4.0086E+01	2.0170E+01	4.0257E+01
F9	Mean	1.0004E+03	1.0576E+03	9.8882E+02	1.0024E+03	1.0248E+03	1.0868E+03	1.1122E+03	1.1706E+03	1.1373E+03
	Std	2.6778E+01	2.5763E+01	2.3183E+01	2.6688E+01	2.6908E+01	2.0202E+01	4.2873E+01	1.6759E+01	5.2212E+01
F10	Mean	1.5678E+03	3.6038E+03	2.8277E+03	3.1618E+03	1.2556E+03	2.6027E+03	4.5345E+03	6.9399E+03	5.1455E+03
	Std	2.2072E+02	6.9187E+02	5.7143E+02	4.8098E+02	1.5895E+02	6.4134E+02	8.4181E+02	4.8195E+02	7.5645E+02
F11	Mean	4.0076E+03	4.8453E+03	4.5941E+03	4.0578E+03	3.7061E+03	5.1199E+03	5.3053E+03	8.0663E+03	6.1013E+03
	Std	6.5977E+02	5.8490E+02	6.1312E+02	7.6447E+02	4.1533E+02	6.6331E+02	7.0361E+02	3.2645E+02	8.2250E+02
F12	Mean	1.2003E+03	1.2012E+03	1.2001E+03	1.2017E+03	1.2002E+03	1.2011E+03	1.2005E+03	1.2024E+03	1.2017E+03
	Std	9.2834E-02	5.0155E-01	5.4439E-02	1.2164E+00	5.5867E-02	4.4254E-01	2.4042E-01	3.0570E-01	5.3028E-01
F13	Mean	1.3005E+03	1.3006E+03	1.3003E+03	1.3004E+03	1.3007E+03	1.3005E+03	1.3022E+03	1.3029E+03	1.3003E+03
	Std	1.2273E-01	1.1474E-01	9.4763E-02	4.1156E-01	1.3493E-01	1.1889E-01	1.3639E+00	3.5938E-01	1.2144E-01
F14	Mean	1.4007E+03	1.4003E+03	1.4003E+03	1.4065E+03	1.4008E+03	1.4003E+03	1.4400E+03	1.4463E+03	1.4003E+03
	Std	2.8994E-01	1.0308E-01	4.2605E-02	8.3030E+00	2.8686E-01	9.1638E-02	2.8004E+01	8.4208E+00	1.2380E-01
F15	Mean	1.5071E+03	1.5271E+03	1.5088E+03	1.8170E+03	1.5116E+03	1.5337E+03	1.2882E+05	4.5999E+03	1.5716E+03
	Std	1.8327E+00	5.3638E+00	2.4302E+00	1.1302E+03	4.9340E+00	9.7625E+00	2.4050E+05	2.8074E+03	2.8067E+01
F16	Mean	1.6109E+03	1.6122E+03	1.6120E+03	1.6110E+03	1.6111E+03	1.6122E+03	1.6127E+03	1.6127E+03	1.6126E+03
	Std	5.8057E-01	4.6247E-01	5.8015E-01	6.4675E-01	5.7456E-01	3.8471E-01	5.9412E-01	3.0846E-01	5.0649E-01
F17	Mean	3.0604E+05	1.9493E+06	2.3160E+05	1.9689E+06	3.3175E+05	1.0292E+06	3.1469E+06	5.5802E+06	4.7009E+06
	Std	1.3631E+05	1.3080E+06	1.2592E+05	3.1512E+06	2.0675E+05	6.4449E+05	5.7517E+06	2.5735E+06	2.8020E+06
F18	Mean	2.1225E+04	5.6946E+04	3.3728E+03	8.7385E+06	1.1318E+04	9.9392E+03	5.2882E+07	1.6528E+08	6.8779E+03
	Std	8.0724E+03	3.5035E+04	1.5431E+03	1.9669E+07	9.6694E+03	2.0250E+04	2.3693E+08	8.0668E+07	5.4275E+03
F19	Mean	1.9235E+03	1.9196E+03	1.9138E+03	1.9389E+03	1.9184E+03	1.9205E+03	1.9660E+03	1.9926E+03	1.9471E+03
	Std	3.1263E+01	2.0834E+01	1.2710E+01	2.3931E+01	2.3010E+01	1.7025E+01	6.0406E+01	2.9357E+01	3.7521E+01
F20	Mean	2.1995E+03	3.4122E+04	5.1074E+03	1.7141E+04	6.0224E+03	7.8023E+03	1.0327E+05	1.6733E+04	3.3849E+04
	Std	7.5884E+01	1.8021E+04	3.0023E+03	7.4385E+03	3.8176E+03	3.8786E+03	1.6951E+05	4.8774E+03	2.7951E+04

TABLE 4: Continued.

Fn	SMA	AO	BBO	GWO	HGS	HHO	MFO	SCA	WOA	ESMA
F21	Mean	1.2711E+05	6.7939E+05	1.3899E+05	8.5813E+05	1.9888E+05	3.8772E+05	1.2068E+06	1.4909E+06	5.7799E+04
	Std	6.3519E+04	4.5314E+05	1.1146E+05	1.6214E+06	1.7553E+05	2.9216E+05	5.8797E+05	1.8902E+06	3.8891E+04
F22	Mean	2.7555E+03	2.8072E+03	2.8419E+03	2.5727E+03	2.8236E+03	3.0357E+03	2.9339E+03	2.9682E+03	2.4606E+03
	Std	1.5769E+02	2.1979E+02	2.1189E+02	1.5728E+02	2.0250E+02	2.4150E+02	1.6281E+02	2.1577E+02	1.5212E+02
F23	Mean	2.5000E+03	2.5000E+03	2.6152E+03	2.6326E+03	2.5000E+03	2.5000E+03	2.6704E+03	2.6339E+03	2.5000E+03
	Std	0	0	2.2300E-03	1.1205E+01	0	0	3.8826E+01	9.1527E+00	0
F24	Mean	2.6000E+03	2.6000E+03	2.6372E+03	2.6000E+03	2.6000E+03	2.6000E+03	2.6001E+03	2.6065E+03	2.6000E+03
	Std	0	1.8281E-05	8.1598E+00	5.6660E-04	8.0779E-05	1.0336E-04	2.2108E-01	7.0969E+00	0
F25	Mean	2.7000E+03	2.7000E+03	2.7229E+03	2.7111E+03	2.7000E+03	2.7000E+03	2.7263E+03	2.7157E+03	2.7000E+03
	Std	0	0	9.0167E+00	4.9014E+00	0	0	7.9006E+00	1.6501E+01	0
F26	Mean	2.7005E+03	2.7761E+03	2.7463E+03	2.7582E+03	2.7385E+03	2.7741E+03	2.7024E+03	2.7044E+03	2.7003E+03
	Std	9.7374E-02	4.2895E+01	5.0249E+01	4.9628E+01	4.8673E+01	4.4104E+01	5.7307E-01	1.9703E+01	8.9681E-02
F27	Mean	2.9000E+03	2.9000E+03	3.3963E+03	3.3329E+03	2.9000E+03	2.9000E+03	3.4011E+03	3.8468E+03	2.9000E+03
	Std	0	6.6028E-03	1.9747E+02	1.2362E+02	0	0	2.8529E+02	3.0114E+02	0
F28	Mean	3.0000E+03	3.0000E+03	5.1955E+03	3.9600E+03	3.0000E+03	3.0000E+03	4.7989E+03	4.9757E+03	3.0000E+03
	Std	0	0	6.7722E+02	2.5731E+02	0	0	3.0996E+02	7.2699E+02	0
F29	Mean	3.8618E+03	3.7255E+03	4.2257E+03	8.5045E+05	3.8755E+03	5.2920E+03	1.1765E+07	6.2428E+06	3.8945E+03
	Std	1.0270E+03	3.1637E+03	4.2292E+02	2.7645E+06	2.3612E+03	6.3368E+03	7.7046E+06	5.0101E+06	6.8743E+02
F30	Mean	6.5833E+03	2.5716E+04	5.8461E+03	4.9750E+04	3.2507E+03	1.4615E+04	2.5278E+05	8.1651E+04	5.4375E+03
	Std	4.3144E+03	2.9540E+04	6.6748E+02	3.0714E+04	3.5841E+02	1.1207E+04	1.0159E+05	8.3282E+04	1.7954E+03
B/T/W	20/10/0	27/3/0	25/5/0	30/0/0	21/9/0	24/6/0	30/0/0	30/0/0	30/0/0	-/-/-

TABLE 5: p value of Wilcoxon test obtained on 30 functions.

Function	SMA	AO	BBO	GWO	HGS	HHO	MFO	SCA	WOA
F1	3.85E-06	7.56E-10	4.34E-03	7.56E-10	3.72E-03	7.56E-10	7.56E-10	7.56E-10	7.56E-10
F2	8.13E-01	7.56E-10	2.34E-08	7.56E-10	8.73E-01	1.09E-09	9.07E-10	7.56E-10	7.56E-10
F3	7.10E-01	7.56E-10	1.23E-09	7.56E-10	7.56E-10	7.56E-10	7.56E-10	7.56E-10	7.56E-10
F4	1.12E-01	4.51E-09	1.04E-01	7.56E-10	7.54E-01	2.91E-06	7.56E-10	7.56E-10	7.56E-10
F5	7.56E-10	8.03E-10	2.01E-01	7.56E-10	5.85E-01	4.43E-01	1.31E-05	7.56E-10	5.06E-08
F6	7.56E-10	7.56E-10	7.56E-10	7.56E-10	7.56E-10	7.56E-10	7.56E-10	7.56E-10	7.56E-10
F7	7.56E-10	7.56E-10	7.56E-10	7.56E-10	5.66E-01	7.56E-10	7.56E-10	7.56E-10	7.56E-10
F8	7.56E-10	7.56E-10	7.56E-10	7.56E-10	6.44E-04	7.56E-10	7.56E-10	7.56E-10	7.56E-10
F9	5.95E-08	7.56E-10	7.58E-09	8.53E-10	7.56E-10	7.56E-10	7.56E-10	7.56E-10	7.56E-10
F10	8.50E-09	7.56E-10	7.56E-10	7.56E-10	7.46E-01	7.56E-10	7.56E-10	7.56E-10	7.56E-10
F11	1.02E-07	1.15E-09	1.02E-09	3.45E-08	1.90E-06	7.56E-10	7.56E-10	7.56E-10	7.56E-10
F12	8.53E-10	7.56E-10	1.98E-08	2.23E-09	2.36E-09	7.56E-10	7.56E-10	7.56E-10	7.56E-10
F13	2.99E-09	7.56E-10	1.28E-01	4.75E-03	7.56E-10	1.23E-09	7.56E-10	7.56E-10	3.78E-09
F14	1.68E-08	2.34E-08	2.29E-01	1.63E-07	7.56E-10	2.89E-04	7.56E-10	7.56E-10	4.84E-02
F15	1.86E-05	7.56E-10	1.66E-09	7.56E-10	8.03E-10	7.56E-10	7.56E-10	7.56E-10	7.56E-10
F16	1.56E-02	8.53E-10	8.53E-10	3.89E-04	3.68E-06	7.56E-10	7.56E-10	7.56E-10	7.56E-10
F17	5.31E-07	7.56E-10	3.96E-03	8.50E-09	1.70E-05	7.56E-10	1.50E-08	7.56E-10	7.56E-10
F18	8.53E-10	7.56E-10	1.57E-01	7.16E-07	4.53E-05	2.74E-02	4.79E-08	7.56E-10	8.05E-03
F19	3.65E-08	7.56E-10	4.51E-09	7.56E-10	5.95E-08	7.56E-10	9.07E-10	7.56E-10	7.56E-10
F20	5.57E-05	7.56E-10	7.56E-10	7.56E-10	8.03E-10	7.56E-10	7.56E-10	7.56E-10	7.56E-10
F21	1.07E-07	1.09E-09	4.64E-06	2.48E-08	5.06E-08	9.07E-10	1.76E-09	7.56E-10	7.56E-10
F22	7.52E-07	6.02E-09	1.47E-09	8.52E-04	2.99E-09	1.23E-09	9.63E-10	9.63E-10	1.23E-09
F23	1.00E+00	1.00E+00	7.56E-10	7.56E-10	1.00E+00	1.00E+00	7.56E-10	7.56E-10	7.56E-10
F24	1.00E+00	3.13E-02	7.56E-10	7.56E-10	6.10E-05	2.56E-06	7.56E-10	7.56E-10	7.56E-10
F25	1.00E+00	1.00E+00	7.56E-10	5.18E-09	1.00E+00	1.00E+00	7.56E-10	1.11E-09	5.61E-06
F26	5.95E-08	7.56E-10	1.06E-06	2.23E-09	7.56E-10	2.82E-09	7.56E-10	7.56E-10	3.35E-06
F27	1.00E+00	1.56E-02	7.56E-10	7.56E-10	1.00E+00	1.00E+00	7.56E-10	7.56E-10	7.56E-10
F28	1.00E+00	1.00E+00	7.56E-10	7.56E-10	1.00E+00	1.00E+00	7.56E-10	7.56E-10	1.63E-09
F29	2.28E-01	1.71E-04	2.54E-02	7.56E-10	1.87E-02	2.08E-01	7.56E-10	7.56E-10	7.56E-10
F30	3.52E-01	8.81E-04	4.84E-02	7.56E-10	5.68E-08	3.26E-07	7.56E-10	7.56E-10	7.56E-10

TABLE 6: Transfer functions of the unknown plant and filter.

Case	Ref	Plant, $H_p(z^{-1})$	Model, $H_M(z^{-1})$
IIR1	[56–60]	$H_p(z^{-1}) = ((1 - 0.4z^{-1} - 0.65z^{-4} + 0.26z^{-6}) / (1 - 0.72z^{-2} - 0.8498z^{-4} + 0.6486z^{-6}))$	$H_M(z^{-1}) = ((b_0 + b_1z^{-1} + b_2z^{-2} + b_3z^{-3} + b_4z^{-4}) / (1 + a_1z^{-1} + a_2z^{-2} + a_3z^{-3} + a_4z^{-4}))$
IIR2	[55, 61, 62]	$H_p(z^{-1}) = ((-0.2 - 0.4z^{-1} - 0.5z^{-2}) / (1 - 0.6z^{-1} - 0.25z^{-2} + 0.2z^{-3}))$	$H_M(z^{-1}) = ((b_0 + b_1z^{-1} + b_2z^{-2}) / (1 + a_1z^{-1} + a_2z^{-2} + a_3z^{-3}))$
IIR3	[55, 61–64]	$H_p(z^{-1}) = ((0.1084 - 0.5419z^{-1} - 1.0837z^{-2} + 1.0837z^{-3} + 0.5419z^{-4} + 0.1084z^{-5}) / (1 + 0.9853z^{-1} - 0.9738z^{-2} + 0.3864z^{-3} + 0.1112z^{-4} + 0.0113z^{-5}))$	$H_M(z^{-1}) = ((b_0 + b_1z^{-1} + b_2z^{-2} + b_3z^{-3} + b_4z^{-4}) / (1 + a_1z^{-1} + a_2z^{-2} + a_3z^{-3} + a_4z^{-4}))$

TABLE 7: Experimental results of ESMA and compared algorithms on 3 IIR identification cases.

Case	SMA	AO	BBO	GWO	HGS	HHO	MFO	SCA	WOA	ESMA
IIR1	Mean	1.1012E-02	9.2255E-03	2.9052E-03	1.2004E-02	1.8952E-02	6.8040E-03	4.0956E-03	1.9357E-02	2.0109E-03
	Best	9.1087E-05	1.9104E-03	6.5049E-05	1.7662E-03	2.3096E-03	6.2293E-05	5.1177E-04	1.3392E-03	5.3420E-05
	Worst	3.1739E-02	2.8735E-02	1.2109E-02	3.0785E-02	2.7388E-02	2.9060E-02	2.1198E-02	9.5033E-02	1.7219E-02
	Median	7.3059E-03	7.4362E-03	1.8063E-03	1.1044E-02	1.9421E-02	5.4307E-03	2.6515E-03	1.6827E-02	1.1571E-03
	Std	9.5252E-03	6.3062E-03	3.3073E-03	7.0418E-03	5.1053E-03	6.7020E-03	4.3626E-03	1.5622E-02	3.2592E-03
IIR2	Mean	3.4077E-04	3.8513E-04	1.5689E-04	1.6618E-04	4.1685E-04	1.6026E-04	4.7920E-04	1.7389E-03	3.0256E-04
	Best	3.8622E-05	2.6994E-05	1.3354E-11	2.6887E-10	5.2293E-09	5.3563E-23	1.0569E-04	1.3807E-04	6.8172E-05
	Worst	7.4025E-04	9.6945E-04	3.4506E-04	9.5517E-04	5.5949E-03	4.3902E-04	1.3167E-03	1.9226E-02	5.6127E-04
	Median	3.3885E-04	3.3963E-04	1.8245E-04	1.9439E-04	1.8240E-04	1.3119E-04	4.4804E-04	6.1910E-04	3.0788E-04
	Std	1.7045E-04	2.2570E-04	1.1225E-04	1.7286E-04	7.7839E-04	1.4025E-04	2.3949E-04	3.0865E-03	1.0333E-04
IIR3	Mean	4.2663E-03	1.2663E-02	4.4409E-02	1.6208E-02	4.7827E-02	1.2380E-02	2.1663E-02	8.0895E-02	1.0130E-03
	Best	2.6697E-04	2.3750E-03	4.4789E-04	4.4459E-04	6.5064E-04	8.7676E-05	5.4884E-03	2.9573E-04	4.7937E-05
	Worst	3.3041E-02	1.1272E-01	2.4610E-01	2.7414E-01	3.5585E-01	1.1648E-01	7.6178E-02	4.7560E-01	5.2843E-03
	Median	1.5646E-03	7.9389E-03	6.1055E-03	1.5811E-03	9.7231E-03	3.7898E-03	1.5413E-02	2.6649E-02	7.2303E-04
	Std	7.1968E-03	1.6379E-02	6.8218E-02	4.1551E-02	9.2787E-02	2.2027E-02	1.6599E-02	1.1247E-01	8.9648E-04
B/T/W	2/1/0	2/1/0	2/0/1	1/1/1	2/1/0	2/1/0	2/0/1	3/0/0	3/0/0	-/-/-

TABLE 8: p value of Wilcoxon test obtained on 3 IIR models.

	SMA	AO	BBO	GWO	HGS	HHO	MFO	SCA	WOA
IIR1	1.8126E-07	9.6371E-08	2.2980E-06	5.5343E-02	3.0918E-08	7.5569E-10	1.5614E-05	6.0478E-05	1.2265E-09
IIR2	1.1229E-01	8.4869E-02	9.1686E-07	2.4074E-05	1.8264E-02	4.5171E-02	8.3898E-06	1.4304E-05	6.9928E-08
IIR3	2.1162E-05	1.0872E-09	5.0638E-09	1.0495E-05	3.3668E-09	1.0872E-09	2.3448E-08	7.5569E-10	1.0872E-09

FIGURE 5: The distribution of the fitness of ESMA and compared algorithms on 3 IIR models.

FIGURE 6: Convergence curves of ESMA and compared algorithms on 3 IIR models.

solution. In the IIR2 case, although ESMA did not get the best solution, it is still very competitive and ranked high compared to other nine optimization methods.

According to the obtained results and analysis above, compared with the original SMA and several other compared algorithms, ESMA has a better ability to identify IIR unknown systems. This reveals that ESMA has a strong global optimization capability and efficiency.

5. Conclusion

In this work, we propose an improved ESMA algorithm. To improve the performance of SMA, the orthogonal learning strategy, chaotic initialization strategy, and boundary reset strategy are embedded into ESMA. The successful integration of these three key strategies further enhances the original SMA's population diversity and global and local search capabilities. These strategies also avoid getting stuck at local optimality and enhance convergence speed and precision. Nine

well-regarded and state-of-the-art optimization methods are used to compare the performance with the proposed ESMA on solving IEEE CEC2014 benchmark function problems. The results obtained show that the performance of ESMA is significantly superior to other compared algorithms. In addition, three IIR model identification problems are selected to illustrate the performance of ESMA in real-world engineering application optimization problems. The simulation results show that ESMA still has a significant advantage in IIR model identification problems. Through the above benchmark problems and IIR model identification problems, it is revealed that ESMA not only has a significant improvement over SMA, but also has more superior performance compared to other algorithms. It also shows that the fusion of the three strategies has a significant improvement on SMA. Therefore, the proposed ESMA with efficient and outstanding performance has reached the initial expectation of improving SMA.

Although ESMA has achieved excellent performance, it still has some shortcomings. For example, in some cases

(F4&5), ESMA is nonideal in terms of convergence accuracy. There is still some room for improvement convergence accuracy. In future work, on the one hand, some new strategies should be considered to further enhance the performance of ESMA. On the other hand, the proposed algorithm could be further utilized to solve more optimization problems in more fields.

Data Availability

Some or all data, models, or codes generated or used during the study are available from the corresponding author upon request.

Conflicts of Interest

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

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant nos. 41772123, 61772365, 61802280, and 61806143.

References

- [1] L. Ma, M. Huang, S. Yang, R. Wang, and X. Wang, "An adaptive localized decision variable analysis approach to large-scale multiobjective and many-objective optimization," *IEEE Transactions on Cybernetics*, pp. 1–13, 2021.
- [2] M. A. Elaziz and S. Mirjalili, "A hyper-heuristic for improving the initial population of whale optimization algorithm," *Knowledge-Based Systems*, vol. 172, pp. 42–63, 2019.
- [3] J. Luo, H. Chen, A. A. Heidari, Y. Xu, Q. Zhang, and C. Li, "Multi-strategy boosted mutative whale-inspired optimization approaches," *Applied Mathematical Modelling*, vol. 73, pp. 109–123, 2019.
- [4] L. Ma, N. Li, Y. Guo et al., "Learning to optimize: reference vector reinforcement learning adaption to constrained many-objective optimization of industrial copper burdening system," *IEEE Transactions on Cybernetics*, pp. 1–14, 2021.
- [5] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–72, 1992.
- [6] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the ICNN'95-International Conference on Neural Networks*, vol. 4, pp. 1942–1948, IEEE, Perth, Australia, November 1995.
- [7] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, Springer Science & Business Media, Berlin, Germany, 2006.
- [8] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems*, vol. 22, no. 3, pp. 52–67, 2002, in English.
- [9] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.
- [10] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.
- [11] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [12] S. Mirjalili, "Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.
- [13] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm," *Computers & Structures*, vol. 169, pp. 1–12, 2016.
- [14] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 120–133, 2016.
- [15] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.
- [16] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
- [17] Q. Askari, I. Younas, and M. Saeed, "Political optimizer: a novel socio-inspired meta-heuristic for global optimization," *Knowledge-Based Systems*, vol. 195, 2020 in English, Article ID 105709.
- [18] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: a new method for stochastic optimization," *Future Generation Computer Systems*, vol. 111, pp. 300–323, 2020.
- [19] L. Abualigah, D. Yousri, M. Abd Elaziz, A. A. Ewees, M. A. A. Al-qaness, and A. H. Gandomi, "Aquila Optimizer: a novel meta-heuristic optimization algorithm," *Computers & Industrial Engineering*, vol. 157, Article ID 107250, 2021.
- [20] Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, "Hunger games search: visions, conception, implementation, deep analysis, perspectives, and towards performance shifts," *Expert Systems with Applications*, vol. 177, Article ID 114864, 2021.
- [21] J. Gupta, P. Nijhawan, and S. Ganguli, "Optimal parameter estimation of PEM fuel cell using slime mould algorithm," *International Journal of Energy Research*, vol. 45, pp. 14732–14744, 2021.
- [22] B. S. Yildiz, "Slime mould algorithm and kriging surrogate model-based approach for enhanced crashworthiness of electric vehicles," *International Journal of Vehicle Design*, vol. 83, no. 1, pp. 54–68, 2020.
- [23] A. M. Zobaa, S. H. E. Abdel Aleem, and H. K. M. Youssef, "Comparative analysis of double-tuned harmonic passive filter design methodologies using slime mould optimization algorithm," in *Proceedings of the 2021 IEEE Texas Power and Energy Conference (TPEC)*, College Station, TX, USA, February 2021.
- [24] G. Vashishtha, S. Chauhan, M. Singh, and R. Kumar, "Bearing defect identification by swarm decomposition considering permutation entropy measure and opposition-based slime mould algorithm," *Measurement*, vol. 178, Article ID 109389, 2021.
- [25] S. Tiachacht, S. Khatir, C. L. Thanh, R. V. Rao, S. Mirjalili, and M. Abdel Wahab, "Inverse problem for dynamic structural health monitoring based on slime mould algorithm," *Engineering with Computers*, 2021.
- [26] A. Draz, M. M. Elkholy, and A. A. El-Fergany, "Slime mould algorithm constrained by the relay operating time for optimal coordination of directional overcurrent relays using multiple standardized tripping curves," *Neural Computing and Applications*, vol. 33, no. 18, pp. 11875–11887, 2021.
- [27] P. Radu-Emil, D. Radu-Codrut, R. Raul-Cristian, M. P. Emil, and S.-S. Alexandra-Iulia, "Slime mould algorithm-based tuning of cost-effective fuzzy controllers for servo systems,"

- International Journal of Computational Intelligence Systems*, vol. 14, no. 1, pp. 1042–1052, 2021.
- [28] T. Gush, C.-H. Kim, S. Admasie, J.-S. Kim, and J.-S. Song, "Optimal smart inverter control for PV and BESS to improve PV hosting capacity of distribution networks using slime mould algorithm," *IEEE Access*, vol. 9, pp. 52164–52176, 2021.
 - [29] S. L. Zubaidi, I. H. Abdulkareem, K. S. Hashim et al., "Hybridised artificial neural network model with slime mould algorithm: a novel methodology for prediction of urban stochastic water demand," *Water*, vol. 12, no. 10, p. 2692, 2020.
 - [30] B. Liu, L. Wang, Y.-H. Jin, F. Tang, and D.-X. Huang, "Improved particle swarm optimization combined with chaos," *Chaos, Solitons & Fractals*, vol. 25, no. 5, pp. 1261–1271, 2005.
 - [31] D. Wu, X. Liang, and M. He, "Orthogonal learning-based improved slime mould algorithm for global optimization," in *Proceedings of the 2021 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, pp. 257–263, Shenyang, China, July 2021.
 - [32] G. Kaur and S. Arora, "Chaotic whale optimization algorithm," *Journal of Computational Design and Engineering*, vol. 5, no. 3, pp. 275–284, 2018.
 - [33] A. Ibrahim, S. Mohammed, H. A. Ali, and S. E. Hussein, "Breast cancer segmentation from thermal images based on chaotic salp swarm algorithm," *IEEE Access*, vol. 8, pp. 122121–122134, 2020.
 - [34] J.-S. Chou and D.-N. Truong, "A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean," *Applied Mathematics and Computation*, vol. 389, Article ID 125535, 2021.
 - [35] B. Ma, H. Ni, X. Zhu, and R. Zhao, "A comprehensive improved salp swarm algorithm on redundant container deployment problem," *IEEE Access*, vol. 7, pp. 136452–136470, 2019.
 - [36] R. A. Ibrahim, M. A. Elaziz, and S. Lu, "Chaotic opposition-based grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization," *Expert Systems with Applications*, vol. 108, pp. 1–27, 2018.
 - [37] X. Zhao, F. Yang, Y. Han, and Y. Cui, "An opposition-based chaotic salp swarm algorithm for global optimization," *IEEE Access*, vol. 8, pp. 36485–36501, 2020.
 - [38] H. Chen, Q. Zhang, J. Luo, Y. Xu, and X. Zhang, "An enhanced Bacterial Foraging Optimization and its application for training kernel extreme learning machine," *Applied Soft Computing*, vol. 86, Article ID 105884, 2020.
 - [39] G. I. Sayed, A. E. Hassanien, and A. T. Azar, "Feature selection via a novel chaotic crow search algorithm," *Neural Computing and Applications*, vol. 31, no. 1, pp. 171–188, 2019.
 - [40] Z.-H. Zhan, J. Zhang, Y. Li, and Y.-H. Shi, "Orthogonal learning particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 6, pp. 832–847, 2011.
 - [41] S. Jiao, G. Chong, C. Huang et al., "Orthogonally adapted Harris hawks optimization for parameter estimation of photovoltaic models," *Energy*, vol. 203, Article ID 117804, 2020.
 - [42] Y. Yang, H. Chen, S. Li, A. A. Heidari, and M. Wang, "Orthogonal learning harmonizing mutation-based fruit fly-inspired optimizers," *Applied Mathematical Modelling*, vol. 86, pp. 368–383, 2020.
 - [43] Y.-X. Lei, J. Gou, C. Wang, W. Luo, and Y.-Q. Cai, "Improved differential evolution with a modified orthogonal learning strategy," *IEEE Access*, vol. 5, pp. 9699–9716, 2017.
 - [44] X. Li, J. Wang, and M. Yin, "Enhancing the performance of cuckoo search algorithm using orthogonal learning method," *Neural Computing and Applications*, vol. 24, no. 6, pp. 1233–1247, 2014.
 - [45] H. Zhang, R. Li, Z. Cai et al., "Advanced orthogonal moth flame optimization with Broyden-Fletcher-Goldfarb-Shanno algorithm: framework and real-world problems," *Expert Systems with Applications*, vol. 159, Article ID 113617, 2020.
 - [46] J.-S. Chou and N.-T. Ngo, "Modified firefly algorithm for multidimensional optimization in structural design problems," *Structural and Multidisciplinary Optimization*, vol. 55, no. 6, pp. 2013–2028, 2017.
 - [47] V. Jothiprakash and R. Arunkumar, "Optimization of hydropower reservoir using evolutionary algorithms coupled with chaos," *Water Resources Management*, vol. 27, no. 7, pp. 1963–1979, 2013.
 - [48] L. Ma, S. Cheng, and Y. Shi, "Enhancing learning efficiency of brain storm optimization via orthogonal learning design," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, 2020.
 - [49] H. Chen, A. A. Heidari, X. Zhao, L. Zhang, and H. Chen, "Advanced orthogonal learning-driven multi-swarm sine cosine optimization: framework and case studies," *Expert Systems with Applications*, vol. 144, Article ID 113113, 2020.
 - [50] P. Civicioglu, "Backtracking search optimization algorithm for numerical optimization problems," *Applied Mathematics and Computation*, vol. 219, no. 15, pp. 8121–8144, 2013.
 - [51] E. Bogar and S. Beyhan, "Adolescent Identity Search Algorithm (AISA): a novel metaheuristic approach for solving optimization problems," *Applied Soft Computing*, vol. 95, Article ID 106503, 2020.
 - [52] P. Civicioglu and E. Besdok, "Bezier search differential evolution algorithm for numerical function optimization," *Expert Systems with Applications*, vol. 165, Article ID 113875, 2021.
 - [53] N. A. Aziz, Z. Ibrahim, M. Mubin, S. W. Nawawi, and M. S. Mohamad, "Improving particle swarm optimization via adaptive switching asynchronous-synchronous update," *Applied Soft Computing*, vol. 72, pp. 298–311, 2018.
 - [54] J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization," vol. 635 Technical report, p. 490, Zhengzhou University, Zhengzhou, China, 2013.
 - [55] A. Mohammadi and S. H. Zahir, "Inclined planes system optimization algorithm for IIR system identification," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 3, pp. 541–558, 2018.
 - [56] Y. Ling, Y. Zhou, and Q. Luo, "Lévy flight trajectory-based whale optimization algorithm for global optimization," *IEEE Access*, vol. 5, pp. 6168–6186, 2017.
 - [57] M. Sharifi and H. Mojallali, "Design of IIR digital filter using modified chaotic orthogonal imperialist competitive algorithm," in *Proceedings of the 2013 13th Iranian Conference on Fuzzy Systems (IFSC)*, pp. 1–6, IEEE, Qazvin, Iran, August 2013.
 - [58] M. A. Sharifi and H. Mojallali, "A modified imperialist competitive algorithm for digital IIR filter design," *Optik*, vol. 126, no. 21, pp. 2979–2984, 2015.
 - [59] N. Karaboga, "A new design method based on artificial bee colony algorithm for digital IIR filters," *Journal of the Franklin Institute*, vol. 346, no. 4, pp. 328–348, 2009.
 - [60] E. Cuevas, J. Gálvez, S. Hinojosa, O. Avalos, D. Zaldivar, and M. Pérez-Cisneros, "A comparison of evolutionary computation techniques for IIR model identification," *Journal of Applied Mathematics*, vol. 2014, Article ID 827206, 9 pages, 2014.

- [61] G. Panda, P. M. Pradhan, and B. Majhi, "IIR system identification using cat swarm optimization," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12671–12683, 2011.
- [62] A. Mohammadi, S. H. Zahiri, and S. M. Razavi, "Infinite impulse response systems modeling by artificial intelligent optimization methods," *Evolving Systems*, vol. 10, no. 2, pp. 221–237, 2019.
- [63] A. Mohammadi and S. H. Zahiri, "IIR model identification using a modified inclined planes system optimization algorithm," *Artificial Intelligence Review*, vol. 48, no. 2, pp. 237–259, 2017, in English.
- [64] T. Mostajabi, J. Poshtan, and Z. Mostajabi, "IIR model identification via evolutionary algorithms," *Artificial Intelligence Review*, vol. 44, no. 1, pp. 87–101, 2015.