The Journal of Supercomputing (2022) 78:13040–13085
https://doi.org/10.1007/s11227-022-04367-w

# Dynamic opposite learning enhanced artificial ecosystem optimizer for IIR system identification

Yanbiao Niu[1] · Xuefeng Yan[1] · Yongzhen Wang[1] · Yanzhao Niu[2]

Accepted: 10 February 2022 / Published online: 14 March 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

## Abstract

The contrive principle of the adaptive infinite impulse response (IIR) filter is to find the filter parameters based on the error function, thus obtaining the best model of the unbeknown plant. Since the error function has a multimodal error surface, it is challenging to get the ideal identification result by traditional methods. In this work, a modified artificial ecosystem optimizer based on the novel dynamic opposite learning (DOL) strategy and a well-designed nonlinear adaptive weight coefficient, called the DAEO, is proposed to minimize the error function. The DOL adopted a random model to dynamically generate the asymmetric opposite solutions of the current population for generation jumping and population formation. To obtain more chances to find the optimal parametric solution, the DAEO is formed from two phases: The first phase produces the initial population by adopting DOL strategy, and the second phase is that DOL is employed as an extra phase to renew the AEO population in each iteration. The asymmetric search area of DOL holistically enhances the exploitation ability of DAEO, and the dynamically changing feature increases the diversity of the swarm, improving the exploration capability of the algorithm. Meanwhile, introducing the well-designed nonlinear adaptive weight coefficient makes search agents explore search space adaptively and poises exploration and exploitation phases. The classical set of benchmark problems is employed to test the performance of DAEO. The experimental results indicate that DAEO ranked first in terms of mean and variance values compared with other algorithms, except $f_{13}$. Furthermore, the DAEO algorithm is also applied to the IIR system identification problem. Simulation results on five benchmarked IIR systems show DAEO outperforms the comparison approach in improving the accuracy of recognition results and can obtain the minimum values of 0 and 1.69E-05 for mean square error (MSE) in the same-order and reduced-order system, respectively, which proves that DAEO is effective and valuable.

**Keyword** Artificial ecosystem optimizer · Dynamic opposite learning · Nonlinear adaptive weight coefficient · IIR system identification

Extended author information available on the last page of the article

## 1 Introduction

Adaptive infinite impulse response (IIR) filter has the advantages of high accuracy, low order, and fewer parameters. Therefore, it has drawn the increasing concern of many researchers and scholars and has been successfully applied to all kinds of complex engineering fields, such as target tracking [1], communication systems [2, 3], signal processing [4, 5], navigation and positioning [6], signal denoising [7–9], and data transmission [10]. The implementation of IIR system identification primarily relies on two aspects: choosing the suitable identification structure and evaluating the filter parameters acquired by the adaptive algorithm. Fortunately, the structure of the majority of systems has been determined through system rationale and working experience and can represent the actual system well. Therefore, the IIR filter identification problem is generally reduced to the parameter optimization problem, which is based on minimization of error from the output of the adaptive filter and the output of the plant to model the unknown system to obtain the optimal model of the actual system [11]. In other words, when this error is minimum, the best IIR filter can be achieved.

However, the error surface (objective function) of the adaptive IIR filter is usually multimodal, non-convex, and non-quadratic [12]. Therefore, traditional gradient-based optimization approaches, such as Newton's method, least squares method, and its variants, can easily fall into local extremes of the error surface and fail to obtain the global optimal solution [13].

To overcome the above disadvantages, the researchers utilized an efficient and robust metaheuristic algorithm for IIR system identification. For example, Yao et al. first adopted genetic algorithm (GA) to estimate parameters for linear and nonlinear systems, which can obtain good results [14]. Krusienski et al. applied particle swarm optimization algorithm (PSO) to construct IIR filter [15]. This algorithm can obtain the minimum value of the MSE. However, GA and PSO are easy to have premature convergence problems, and GA cannot perform local search in high-dimensional space. Later, Panda et al. used the cat swarm optimization (CSO) algorithm to estimate the parameters of IIR modeling [16]. This method can give the desired results. However, this algorithm cannot balance the balance between exploration and exploitation and still suffers from the tendency to fall into local optima. Mandal et al. used an improved differential evolution algorithm (DEWM) to structure IIR filters [17]. This algorithm increases the diversity of the population by wavelet mutation and, to some extent, improves the performance of the algorithm. However, the algorithm requires more control parameters to be adjusted, which reduces the convergence speed. Saha et al. adopted an improved bat algorithm (OBA) to work out IIR identification [18]. Upadhyay et al. applied a modified harmony search algorithm (OHS) to IIR identification [19]. These two methods are based on the opposite learning method to expand the search space and improve the exploration capability of the algorithm, but for complex optimization problems, the above algorithm still suffers from premature convergence. Jiang et al. adopted a novel hybrid algorithm (HPSO–GSA) to deal with filter design problems [20]. Lagos-Eulogio et al. presented a hybrid algorithm (CPSO-DE) to obtain optimal filter parameters [21].

These two methods can obtain desirable recognition results, but the performance of the algorithms also depends on the parameter values and may be affected by premature convergence and stagnation. M Kumar et al. applied a modified interior search algorithm with Lévy flight to IIR system identification [22]. Luo et al. adopted an improved whale optimization algorithm (RWOA) to deal with the identification problem [13]. This algorithm can get a fast convergence rate at the beginning of the iteration, but the population tends to lose diversity and fall into local extremes at the end of the iteration. Despite the advantages of the above studies, most of them have the problem of local convergence in the face of complex optimization problems. Moreover, according to the theory of no-free-lunch (NFL), there is no optimization algorithm that can solve all optimization problems [23]. Therefore, it inspires researchers and scholars to propose new methods to cope with the IIR identification problem.

Artificial ecosystem-based optimizer (AEO) is an effective swarm optimization algorithm motivated by energy flow in the ecosystem [24]. It mimics the natural patterns of living organisms through production, consumption, and decomposition phases. The AEO algorithm has the advantages of an uncomplicated structure and fewer control parameters. Moreover, the poise between exploration and exploitation of the AEO mainly relied on the degression value in the search process, saving time, and burden. To further improve the search capabilities of AEO, dynamic opposite learning (DOL) is first embedded in AEO, which guides individuals to learn in an asymmetrical dynamic search area. This method will significantly boost the possibility of obtaining the global solution. Secondly, a self-adaptive nonlinear weight coefficient was adopted to better poise the relationship between exploration and exploration. The classical test function set entirely verifies the effectiveness of the DAEO algorithm. The DAEO algorithm is also utilized for the IIR system identification and compared with the other algorithms. Experimental results show that DAEO can obtain better parameters for adaptive IIR filters, verifying the effectiveness and excellence of the proposed algorithm. Motivated by the above discussion, our aim is to propose a dynamic opposite learning enhanced artificial ecosystem algorithm (DAEO) that determines the parameters of the filter to obtain an optimal set of parameters so as to the output of the adaptive IIR filter can accurately trace that of the unknown system when both systems are supplied with the same input signal. The main contributions of this paper are as follows: (1) A novel DOL strategy is embedded in AEO for the first time, increasing the diversity of the population and helping individuals to find the best solution quickly. (2) A new adaptive nonlinear weight coefficient is designed to balance the global and local search capability of DAEO. (3) The DAEO is proposed as a new optimization tool for IIR system identification. Experiment on benchmark functions and IIR system cases validates the feasibility and effectiveness of the approach.

The research content of the paper is formed as follows: Adaptive IIR filter system identification is described in Sect. 2. The basic definition of AEO is shown in Sect. 3. Section 4 introduces the DAEO algorithm. To attest to the advantageous performance of the DAEO, two kinds of experimental results are presented in Sect. 5. Discussion and analyses of simulation results are conducted in Sect. 6. Finally, Sect. 7 gives conclusions and future directions.

## 2 Adaptive IIR system identification

Adaptive IIR filter is one of the effective methods to deal with model identification problems. The primary work of system recognition is to adjust the adaptive IIR filter coefficients through the optimization algorithm to make the output of the filter closer to that of the unbeknown actual system when the same input signal is transmitted to both the unbeknown actual system and the adaptive filter [13]. The flow of filter identification is presented in Fig. 1.

In Fig. 1, $X(t)$ expresses the input signal values of the filter and unknown actual system at time instant $t$, $y_0(t)$ represents the noiseless output value of unbeknown actual system at time instant $t$, $V(t)$ expresses the noise value of the recognition system at time instant $t$, and $y(t)$ represents the actual export of the unbeknown actual model with noise added. $\widehat{y}(t)$ expresses the export of the IIR filter at time instant $t$, and $e(t)$ denotes the error between the export of the actual system and the export of the adaptive IIR filter.

The relationship between input and output of the IIR system can be denoted by the following difference equation [25]:

$$\hat{y}(t) = \sum_{i=0}^{M} \hat{b}_i X(t-i) - \sum_{i=1}^{N} \hat{a}_i \hat{y}(t-i) \tag{1}$$

where $N(\geq M)$ represents the order of the adaptive IIR filter. Based on Eq. (1), the transport function of the IIR filter is described as follows [26]:

$$G_M(z) = \frac{\sum_{i=0}^{M} \hat{b}_i z^{-i}}{1 + \sum_{i=1}^{N} \hat{a}_i z^{-i}} \tag{2}$$

where $\widehat{b}_i$ and $\widehat{a}_i$ express the input and output optimization parameters of the filter, severally.

It is assumed that the transport function construction of the unbeknown real system is the same as that of the IIR filter, and the transport function of the unbeknown system is given as [11]:

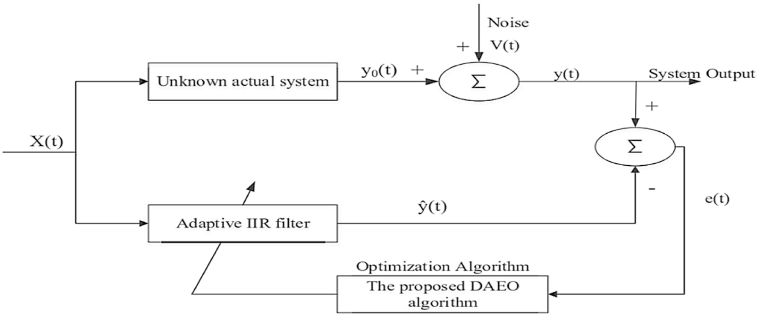


**Fig. 1** System identification of adaptive IIR filter

$$G_p(z) = \frac{\sum_{i=0}^{M} b_i z^{-i}}{1 + \sum_{i=1}^{N} a_i z^{-i}} \tag{3}$$

where $a_i$ and $b_i$ represent the real parameters of actual system.

IIR system identification aims to utilize an adaptive IIR filter with the transfer function $G_M(z)$ to identify an unknown system with the transfer function $G_p(z)$. In this case, the problem transforms into an optimization problem when the adaptive IIR filter parameters $\left[\hat{a}_i, \hat{b}_i\right]^T$ approximate the unknown system parameters $\left[a_i, b_i\right]^T$. As shown in Fig. 1, the error $e(t)$ is minimized using an optimization algorithm to obtain the optimal parameter vector $\hat{w} = \left[a_1, a_2, \cdots, a_N, b_0, b_1, \cdots, b_M\right]^T$. The error is usually expressed by mean square error (MSE), which is computed as follows.

$$MSE = \frac{1}{L}\sum_{t=1}^{L}(y(t) - \hat{y}(t))^2 \tag{4}$$

where $L$ represents the number of input samples.

The IIR system identification problem has been extensively studied in the literature [13–22]. In this paper, we try to apply an improved AEO algorithm (DAEO) to the optimal design of the IIR filter $G_M(z)$ and thus to fit the system model $G_p(z)$ to be identified. In fact, the problem can ultimately be reduced to nonlinear optimization problems with multiple local extrema. Equation (4) is the fitness function (error surface function) for IIR model identification. The DAEO algorithm uses the fitness function to determine the merit of the currently obtained parameters. If the order of the IIR filter $G_M(z)$ and the order of the unknown system model $G_p(z)$ to be identified are equal, then only one global optimum exists for the error surface function MSE. Conversely, then there exist multiple local minima for the error surface MSE. In practical applications, the system model to be identified is considered as a black box structure. Therefore, the order of the IIR filter and the order of the unknown system model are usually not equal, which means that the fitness function usually has many local minima. In summary, due to the complexity of the error surface of the IIR model for identifying, which is actually an optimization problem with multiple local extrema, it has essential applications in practical optimization problems. Therefore, the use of a suitable optimization algorithm is crucial to the identification results.

## 3 Artificial ecosystem optimizer (AEO)

The artificial ecosystem optimizer is an efficient metaheuristic algorithm presented by Zhao et al. in 2019 [24]. There are three main operators in the AEO, which are production, consumption, and decomposition. The optimization process goes through the following stages:

### 3.1 Production

In AEO, the best agent ($X_n$) is regarded as decomposer. The producer is considered the worst agent ($X_1$) in the current population, which can direct other searching agents, such as herbivores and omnivores, to search in promising areas. A new agent (producer) is created through the production phase between the best agent $X_n$ and a randomly selected agent $X_{rand}$ within the search area to replace the previous position. The updated model for the production phase is shown as follows:

$$a = \left(1 - t/T_{max}\right)r_1 \tag{5}$$

$$X_{rand} = r(ub - lb) + lb \tag{6}$$

$$X_1(t+1) = (1-a)X_n(t) + aX_{rand}(t) \tag{7}$$

where $r_1$ denotes a random number inside the extent from 0 to1, $r \in [0, 1]$ is a random vector, parameter $a$ is weight coefficient, $T_{max}$ is the total iteration times, and $t$ expresses present iteration value. lb and ub are lower and upper boundaries of the search area.

### 3.2 Consumption

The search agents in this phase are called consumers, which can eat either the producer or other consumers with low energy levels, or both. There are three types of consumers, each using a different strategy to get the best candidate solution.

The first type, named herbivore, can only eat producer and renews itself by Eq. (8).

$$X_i(t+1) = X_i(t) + CF\left(X_i(t) - X_1(t)\right),\quad i \in [2, \cdots, n] \tag{8}$$

where $CF$ is consumption factor, and it is considered as random walk with levy flight behavior to avoid the parameter setting, which is calculated as follows:

$$CF = \frac{1}{2}\frac{v_1}{\left|v_2\right|},\quad v_1 \sim N(0,1),\quad v_2 \sim N(0,1) \tag{9}$$

where $N(0, 1)$ denotes a normal distribution.

The second mold is carnivore that can only randomly choose other consumers with advanced energy levels for food. The model of the carnivore is expressed by Eq. (10).

$$X_i(t+1) = X_i(t) + CF\left(X_i(t) - X_j(t)\right),\quad i \in [3, \cdots, n],\quad j = randi\ ([2, i-1]) \tag{10}$$

The third type is omnivore, which signifies that not only producers but also consumers with high energy levels can eat randomly. It updates its location by Eq. (11).

$$X_i(t+1) = X_i(t) + CF\left(r_2\left(X_i(t) - X_1(t)\right) + \left(1 - r_2\right)\left(X_i(t) - X_j(t)\right)\right),\quad i = 3, \cdots, n,\quad j = randi([2, i-1]) \tag{11}$$

where $r_2 \in [0, 1]$ is a random number, and $j = randi([2, i-1])$.

### 3.3 Decomposition

On the basis of ecosystem function, decomposition is a critical phase that provides energy for producers. The decomposition stage is shown as follows:

$$X_i(t+1) = X_n(t) + DF\left(gX_n(t) - kX_i(t)\right),\ \ \forall i \in n,\ DF = 3u,\quad u \sim N(0,1),\ g = r_3 randi([1\ \ 2]) - 1,\ k = 2r_3 - 1 \tag{12}$$

where $DF$ is the decomposition factor. $g$ and $k$ represent the weight coefficient.

Since the AEO algorithm has been proposed, it has been proved to provide very competitive results compared with other well-known swarm intelligence algorithms. This algorithm has attracted the attention of many scholars and has also been increasingly used in many application areas [27–29]. Although the basic AEO shows good performance, it still has the disadvantage of slow convergence and low accuracy. The reasons for this phenomenon are as follows: The first one is that updating methods that rely only on linear weight parameters may degrade the performance of the algorithm, causing imbalance between exploration and exploitation of the algorithm, resulting in producing suboptimal solutions. The second is that no effective means is added to keep the diversity of the swarm from fending off declining into local optima. To address the above deficiencies, a novel enhanced AEO algorithm (DAEO) is proposed in this paper, which will be described in detail in the following sections.

The pseudo-code of AEO is displayed in Algorithm 1.

**Algorithm 1.** AEO pseudo-code

```
Generate the population of ecosystem $X_i (i = 1,2,....n)$
Compute the fitness of each search agent
Sorted by fitness value from highest to lowest
$X^*$=the best search agent
while $(t < T_{max})$
    //Production phase//
    For searching agent $X_1$, renew the position by Eq.(7)
    //Consumption phase//
    For searching agent $X_k$, $k = 2{:}n$
        if $\left(rand < \frac{1}{3}\right)$
            //Herbivore//
            Renew the position of the present agent by Eq. (8)
        else if $\left(\frac{1}{3} \le rand \le \frac{2}{3}\right)$
            // Carnivore //
            Renew the position of the present agent by Eq. (10)
        else
            // Omnivore //
            Renew the position of the present agent by Eq. (11)
        end if
    end for
    Compute the fitness of each agent
    Update $X^*$
    //Decomposition phase//
    Renew the position of the every agent by Eq.(12)
    Compute the fitness of each agent
    Update $X^*$
    $t = t + 1$
end while
Return $X^*$
```

# 4 Proposed DAEO algorithm

In this paper, the variant of the AEO algorithms is proposed to enhance the performance of the algorithm, where two strategies are implemented. In the following subsection, we first introduce the nonlinear weight coefficient and dynamic opposite learning strategy and then present our DAEO in detail.

## 4.1 Proposed nonlinear weight coefficient

For metaheuristic algorithms, exploration and exploitation of the algorithm are executed together. Excessive exploitation will prevent the search agent from moving toward the global solution. Excessive exploration will decrease the quality of the solution. Therefore, the performance of the evolutionary algorithm can be enhanced by balancing the two stages. According to the working principle of AEO, parameter *a* controls the exploration and exploitation capability of the algorithm. The higher value of *a* is propitious to the exploration of the algorithm, while the lower value of *a* is propitious to local exploitation of the algorithm. The parameter *a* that plays an essential role in the transition from exploration to exploitation obtains a smaller value as the iteration times grow. According to Eq. (7), the update of the new position depends mainly on two components: the product of the position of the randomly selected search agent $X_{rand}$ and $a$ and the product of the current best position and $(1-a)$. The variation of original parameter $a$ with the number of iterations is shown in Fig. 2. The parameter $a$ takes values ranging from 1 to 0 during the iterative process. According to Algorithm 1, the proportion of parameters $a$ greater than 0.5 is small, which leads to a further reduction of the exploration capability. The algorithm mainly focuses on the exploitation stage. All of these will lead to premature convergence and local extremum. Moreover, many optimization issues need good nonlinear changes to avoid the local optimal solution. The value of parameter $a$ cannot imitate the actual situation. Therefore, a nonlinear adaptive weight coefficient is proposed. Since the search process of the AEO is nonlinear, our goal is to spend more time in the exploration than in the exploitation. Therefore, the nonlinear adaptive weight coefficient in DAEO is described as follows:

$$a = r_1 cos\left(t/T_{max}\right) \tag{13}$$

Compared with the original weight coefficient, the nonlinear weight is more focused on exploration during iterations. The nonlinear parameter is compared with the original parameter, as shown in Fig. 2. Equation (13) and Fig. 2 indicate that the higher value of the proposed nonlinear parameter during the iteration indicates that it tends to be an exploration for a longer time.

## 4.2 Dynamic opposite learning (DOL) strategy

The opposition-based learning (OBL) is adopted to enhance the quality of the solution space of the algorithm, which gives more chances to approach the global

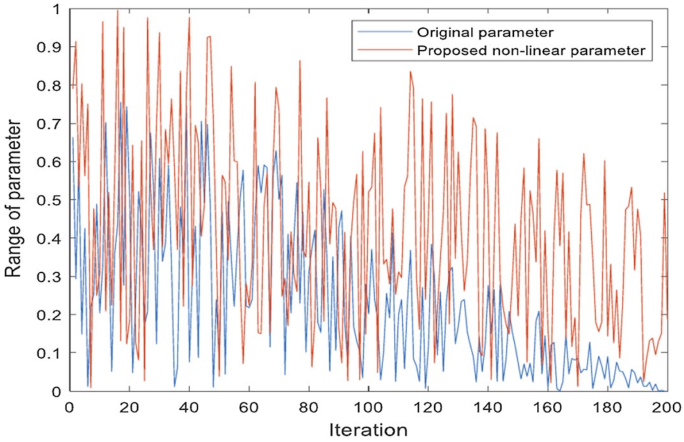


**Fig. 2** Comparison of parameters

solution by introducing the opposite point[30]. Much literature shows that effective expansion of search space can increase the optimization performance of different algorithms [31, 32]. However, if the local extremum exists in the searching area from the present position to the reverse position, the OBL strategy will tend to converge to the local optimal position. A dynamic opposite learning (DOL) strategy is applied to solve these problems, and the idea is exemplified in Fig. 3 [33]. First, a random reverse number $X^{AO}$ which is expressed as $X^{AO} = rand * X^O$ is set to avoid the candidate solution declining into local extremum. $X^O$ is opposite number calculated by OBL. By replacing $X^O$ with $X^{AO}$, the asymmetric search area which is dynamically adjusted with the $X^{AO}$ will be formed to change symmetric search space. Then, the DOL strategy can get a random number $\widehat{X}$ from $X$ to $X^{AO}$ as the asymmetric opposite solution of $X\left(\widehat{X} = X + rand\left(X^{AO} - X\right)\right)$. Although $X^{AO}$ makes the search space diverse, as the iteration goes on, the scope of the search space may become smaller, which will decrease the exploration capacity of the algorithm. To palliate this effect, the weighting factor $W$ is applied to get the optimal performance of DOL. DOL is redeclared as $\widehat{X} = X + W * rand\left(X^{AO} - X\right)$, where $W$ is an invariant number. The asymmetry of search space can prevent the algorithm from declining into the local extremum and enhance the exploitation capability. The dynamics of the search area enhance the diversity of the swarm and makes it has a good exploration capacity.

The model of the DOL is displayed as follows:

### 4.2.1 Dynamic opposite number

$\boldsymbol{X} \in [\boldsymbol{lb}, \boldsymbol{ub}]$ is defined as an actual value. The dynamic reverse value $\widehat{X}$ can be computed by Eq. (14), where $lb$ and $ub$ are the boundaries of the search space. $rand$ is a random number with a scope of (0,1). ($W > 0$) is the weighting coefficient and is usually set to 1.

$$\hat{X} = X + w * rand * (rand * (ub + lb - X) - X) \tag{14}$$

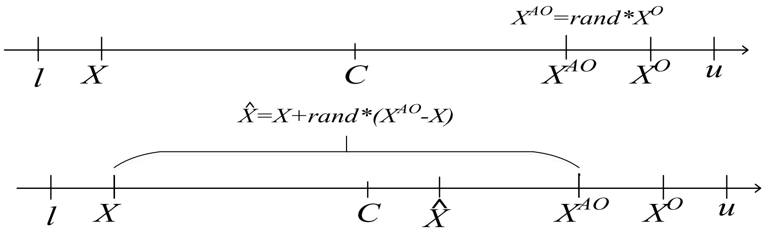


**Fig. 3** DOL asymmetric dynamic search area

### 4.2.2 Dynamic opposite point

$X = \left(X_1, X_2, \ldots, X_D\right)$ represents search agent in the population, which is a point in D-dimensional space. Its opposite position in the search area will be expressed by $\widehat{X} = \left(\widehat{X}_1, \widehat{X}_2, \ldots, \widehat{X}_D\right)$. Therefore, the values of all multi-dimensional dynamic reverse points in $\widehat{X}$ will be calculated by Eq. (15).

$$\hat{X}_j = X_j + w * rand * \left(rand * \left(ub_j + lb_j - X_j\right) - X_j\right) \;\; j = 1 : D \tag{15}$$

### 4.2.3 DOL-based optimization process

1. Generate the ecosystem population $X$ as $X_i$ where $(i = 1, 2, \cdots, n)$.
2. Dynamically update interval boundaries $[lb, ub]$ by $lb = min(X)$,$ub = max(X)$, and determine the dynamic opposite positions of ecosystem $X$ as $\widehat{X}_i$ where $(i = 1, 2, \cdots, n)$, according to Eq. (15). If $\widehat{X}_i \notin [l, u]$, $\widehat{X}_i$ should be reset as a random number in $[lb, ub]$.
3. Select the $n$ search agents with good fitness values from $\left\{X \cup \widehat{X}\right\}$ as new population.

## 4.3 DAEO framework

The segment describes the primary details of the DAEO algorithm. To overcome the shortcoming of the AEO, the nonlinear weight coefficient and DOL strategy are put forward to enhance the performance of the algorithm. The steps of the DAEO are listed as follows:

Step 1. DAEO initialization: DAEO randomly produces the initial population according to the population size and the boundary value of the search space.

Step 2. Apply DOL: According to Eq. (15), the DOL was adopted to obtain the reverse solution of each solution. Then, DOL selects $n$ search agents with good fitness values as the new population from the generated population and its reverse solutions which are obtained in Step 2.

Step 3. Update position: The weight factor $a$ is calculated by replacing Eq. (5) with Eq. (13). Then, the position of search agents will be renewed based on Eq. (6)–(12). If the search agent is out of the search area, it will be amended.

Step 4. Fitness evaluation: The fitness values of all search agents in the population are calculated, which gets the best value as the best solution.

Step 5. Execution termination: The optimum solution will be output if the terminus situation is satisfied. Otherwise, Steps 2–4 are executed.

The pseudo-code of DAEO is displayed in Algorithm 2. The flow diagram of the proposed algorithm is shown in Fig. 4.

**Algorithm 2.** DAEO pseudo-code

```
Generate the population of ecosystem $X_i (i = 1,2,....n)$
Compute the fitness of each search agent
Sorted by fitness value from highest to lowest
$X^*$=the best search agent
while $(t < T_{max})$
Dynamically updated Interval boundaries
Apply the DOL to produce the opposite solutions by Eq. (15)
if any opposite solution is out of the search area, it is amended
Calculate the fitness of each search agent
Select $N$ search agents with good fitness values as the population
//Production phase//
For searching agent $\boldsymbol{X_1}$,renew the position by Eq.(7)
//Consumption phase//
For searching agent $X_k,\ k = 2:n$
if $\left(rand < \frac{1}{3}\right)$
//Herbivore//
    Renew the position of the present agent by Eq. (8)
else if $\left(\frac{1}{3} \le rand \le \frac{2}{3}\right)$
// Carnivore //
    Renew the position of the present agent by Eq. (10)
else
// Omnivore //
Renew the position of the present agent by Eq. (11)
end if
end for
Compute the fitness of each agent
Update $X^*$
//Decomposition phase//
Renew the position of the every agent by Eq.(12)
Compute the fitness of each agent
Update $X^*$
    $t = t + 1$
 end while
 Return $X^*$
```

### 4.4 Time complexity analysis

In this segment, the worst time complexity of the original AEO algorithm and the proposed DAEO algorithm is calculated by using $big - O$. The complexity of AEO and DAEO is described as follows:

#### 4.4.1 Original AEO

The original AEO generates population of ecosystem in $O(n \times d)$, where $n$ is the population size and $d$ is dimension.

It takes $O(n)$ to calculate the fitness value.

The position renewal in the original AEO needs $O(n \times d)$.

Therefore, for $T_{max}$ iterations, the total time complexity of the AEO needs $O(n \times d \times T_{max})$.

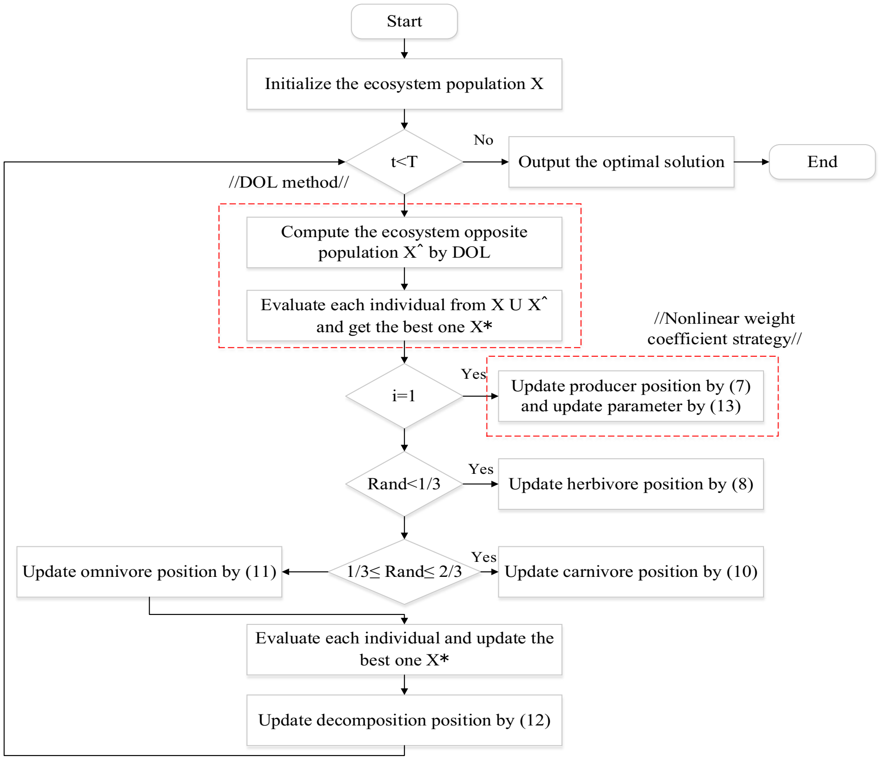

Start
Initialize the ecosystem population X
t<T
No
Output the optimal solution
End
//DOL method//
Compute the ecosystem opposite population X^ by DOL
Evaluate each individual from X U X^ and get the best one X*
//Nonlinear weight coefficient strategy//
i=1
Yes
Update producer position by (7) and update parameter by (13)
Rand<1/3
Yes
Update herbivore position by (8)
1/3≤ Rand≤ 2/3
Yes
Update carnivore position by (10)
Update omnivore position by (11)
Evaluate each individual and update the best one X*
Update decomposition position by (12)

**Fig. 4** The proposed DAEO algorithm

### 4.4.2 The proposed DAEO

The initial population of DAEO needs $O(n \times d)$, where $n$ is the population size and $d$ is dimension.

It takes $O(n)$ to calculate the fitness value.

DOL-based strategy needs $O(n \times d)$.

The position renewal in the DAEO needs $O(n \times d)$.

In summary, for $T_{max}$ iterations, the total time complexity of the DAEO needs $O\left(n \times d \times T_{max}\right)$. Therefore, from what has been described above, we may safely draw the conclusion that the two algorithms are the same in terms of complexity.

## 5 Experiment and results

To ensure the efficiency of the proposed DAEO algorithm for the system identification optimization problem, the DAEO algorithm should first be evaluated and tested on several benchmark test problems. A common way is to use benchmark functions with different characteristics to evaluate optimization algorithms with stochasticity.

Various scholars have adopted these benchmark functions to verify the performance of their algorithms [34, 35]. Using these functions makes it possible to ensure that the results obtained by the algorithm are not accidental. Experiment 1: We first present the test functions used in this work, and then, in order to evaluate the DAEO algorithm, the results of the algorithm are compared with those of other algorithms. Experiment 2: DAEO is applied to five benchmark IIR systems using the same-order and reduced-order systems to obtain the filter parameters, aiming to verify the capability of the DAEO to work out actual problems.

### 5.1 Experiment 1: test functions

In this segment, a group of challenging functions is employed to validate the performance of DAEO. Table 1 includes 21 functions with various characteristics and complexity levels, which are symbolized by the letter $f$ and numbered $f_1$, $f_2$,…, $f_{21}$ [36, 37]. In Table 1, $\left(f_1 \sim f_{21}\right)$ is the objective function that the algorithm needs to optimize, "Dim" represents the number of the function variables, "Range" represents the search range of the variables, and "$f_{min}$" represents the theoretical optimal value of the function. The functions $\left(f_1 \sim f_7\right)$ are high-dimensional unimodal functions, which are employed to examine the local search capability of the algorithm. The functions $\left(f_8 \sim f_{15}\right)$ are high-dimensional multimodal functions, which are used to test the global search capability of the algorithm. The functions $\left(f_{16} \sim f_{21}\right)$ are fixed-dimensional multimodal functions that can reveal the ability to verify the balance between exploration and exploitation of algorithms in realizing more complex and challenging landscapes.

#### 5.1.1 Comparative study

To better validate the performance of the proposed algorithm, DAEO is compared with other six algorithms, including moth–flame optimization algorithm (MFO) [38], salp swarm algorithm (SSA) [39], a hybrid algorithm of particle swarm optimization and gravitational search algorithm (PSOGSA) [40], grey wolf optimizer (GWO) [41], whale optimization algorithm (WOA) [42], and artificial ecosystem-based optimizer (AEO) [24]. For an impartial comparison, each algorithm executes 30 times independently, and the total iteration times are 800. The population size of all algorithms is 30, and the main parameter settings are set in Table 2. All algorithms were implemented on a PC with an Intel Core i5-10,400 and 8 GB of RAM using MATLAB R2017 (a). In this study, the Wilcoxon nonparametric rank-sum test [43] was also employed to test significant differences between the DAEO and the other algorithms on benchmark function. The *P*-values of comparisons are listed in Table 4.

#### 5.1.2 Simulation result

In order to fully evaluate the solution quality of the DAEO algorithm, the mean and standard deviation values obtained for each benchmark function executed 30 times

**Table 1** Benchmark functions

| Benchmark functions | Dim | Range | $f_{\min}$ |
|---|---|---|---|
| $f_1(x) = \sum_{i=1}^{n} x_i^2$ | 50 | $x_i \in [-100, 100]$ | 0 |
| $f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$ | 50 | $x_i \in [-10, 10]$ | 0 |
| $f_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$ | 50 | $x_i \in [-100, 100]$ | 0 |
| $f_4(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le D\}$ | 50 | $x_i \in [-100, 100]$ | 0 |
| $f_5(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$ | 50 | $\mathrm{x_i} \in [-30, 30]\, \mathrm{x_i} \in [-30, 30]$<br>$x_i \in [-30, 30]$ | 0 |
| $f_6(x) = \sum_{i=1}^{n} (random(0,1) * \lvert x_i - 1/i \rvert)$ | 50 | $x_i \in [-5, 5]$ | 0 |
| $f_7(x) = \sum_{i=1}^{n} x_i^4 + random(0,1)$ | 50 | $x_i = [-1.28, 1.28]$ | 0 |
| $f_8(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$ | 50 | $x_i \in [-500, 500]$ | 0 |
| $f_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$ | 50 | $x_i \in [-5.12, 5.12]$ | 0 |
| $f_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i\right)\right) + 20 + e$ | 50 | $x_i \in [-32, 32]$ | 0 |
| $f_{11} = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ | 50 | $x_i \in [-600, 600]$ | 0 |

**Table 1** (continued)

| Benchmark functions | Dim | Range | $f_{\min}$ |
|---|---|---|---|
| $f_{12}(x) = \frac{\pi}{n}\left\{10\sin\left(\pi y_1\right) + \sum_{i=1}^{n-1}\left(y_i - 1\right)^2\left[1 + 10\sin^2\left(\pi y_{i+1}\right)\right] + \left(y_n - 1\right)^2\right\}$<br>$+ \sum_{i=1}^{n}\mu\left(x_i, 10, 100, 4\right)$<br>$y_i = 1 + \frac{x_i + 1}{4}\mu\left(x_i, a, k, m\right) = \begin{cases} k\left(x_i - a\right)^m > a \\ 0 - a < x_i < a \\ k\left(-x_i - a\right)^m x_i < -a \end{cases}$ | 50 | $x_i \in [-50, 50]$ | 0 |
| $f_{13} = 0.1\left\{\sin^2\left(3\pi x_1 + \sum_{i=1}^{n}\begin{matrix}\left(x_i - 1\right)^2\left[1 + \sin^2\left(3\pi x_i + 1\right)\right] \\ +\left(x_n - 1\right)^2\left[1 + \sin^2\left(2\pi x_n\right)\right]\end{matrix}\right)\right\} + \sum_{i=1}^{n}U\left(x_i, 5, 100, 4\right)$ | 50 | $x_i \in [-50, 50]$ | 0 |
| $f_{14}(x) = \sum_{i=1}^{n}x_i\sin\left(x_i\right) + 0.1x_i$ | 50 | $x_i \in [-10, 10]$ | 0 |
| $f_{15}(x) = \sum_{i=1}^{n}x_i^2 + \left(\sum_{i=1}^{n}0.5ix_i\right)^2 + \left(\sum_{i=1}^{n}0.5ix_i\right)^4$ | 50 | $x_i \in [-5, 10]$ | 0 |
| $f_{16}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}\left(x_i - a_{ij}\right)^6}\right)^{-1}$ | 2 | $x_i \in [-65, 65]$ | 1 |
| $f_{17}(x) = \sum_{j=1}^{11}\left[a_i - \frac{x_1\left(b_i^2 + b_i x_2\right)}{b_i^2 + b_i x_3 + x_4}\right]^2$ | 4 | $x_i \in [-5, 5]$ | 0.0003075 |
| $f_{18}(x) = -\sum_{i=1}^{5}\left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$ | 4 | $x_i \in [0, 10][0, 10]$ | $-10.1532$ |
| $f_{19}(x) = -\sum_{i=1}^{7}\left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$ | 4 | $x_i \in [0, 10]$ | $-10.4029$ |

**Table 1** (continued)

| Benchmark functions | Dim | Range | $f_{\min}$ |
|---|---|---|---|
| $f_{20}(x) = -\sum_{i=1}^{10} [(x - a_i)(x - a_i)^T + c_i]^{-1}$ | 3 | $x_i \in [0, 10]$ | $-10.5364$ |
| $f_{21}(x) = \sum_{k=1}^{n} c_i \exp[-\sum_{j=1}^{6} a_{ij}\left(x_j - p_{ij}\right)^2]$ | 6 | $x_i \in [0, 1]$ | $-3.32$ |

independently are presented in Table 3. These two indexes are utilized to evaluate the overall performance of the algorithm. Moreover, the convergence behavior of the average fitness values of all algorithms applied to 21 classic benchmark functions is presented in Fig. 5.

The data marked in bold in Table 3 are the best values obtained by comparing the results calculated by the seven algorithms under each test function. From Table 3, it is clear that DAEO obtained the best results in terms of the mean and standard deviation of all unimodal functions $\mathrm{f}_1$ ~ $\mathrm{f}_7$ . In particular, for $f_1 \sim f_5$, DAEO can obtain the theoretical optimal values of the test function. For unimodal functions, the global optimal solution is within a narrow parabolic-shaped valley, thus making it difficult for many algorithms to search for the global optimum. But DAEO shows superior performance. From the convergence curves of the unimodal functions in Fig. 5, it can be seen that DAEO outperforms all other algorithms in terms of convergence speed, converging rapidly toward the optimal solution. It is clear that DAEO displays the best mining ability in the comparison algorithm.

Unlike unimodal functions, the number of local optima for multimodal functions grows exponentially. Therefore, these functions are more appropriate for verifying the exploration ability of the algorithm. In multimodal functions, compared to other algorithms, DAEO obtained the best values on $f_8 \sim f_{15}$, except for function $f_{13}$. In particular, DAEO can obtain the theoretical values of the test functions on $f_8$, $f_9$, $f_{11}$, $f_{14}$, and $f_{15}$ as well. In terms of mean value, although it failed to achieve the best results on $f_{13}$, DAEO ranked only behind AEO and WOA, indicating that DAEO has a strong competitive capability compared to the other algorithms. Furthermore, the convergence curves in Fig. 5 show that DAEO converges faster than other algorithms in most multimodal benchmark functions and is more promising in other cases. The above results indicate that DAEO gives an excellent exploration capability.

The ability to maintain a suitable balance between exploration and exploitation can be well verified on the fixed-dimensional test problem. In Table 3, the mean value obtained by the DAEO algorithm ranks first compared with other algorithms. Good results are obtained for DAEO on all fixed-dimensional functions. The excellent performance of DAEO is well verified on the convergence curve. At the beginning of the iteration, DAEO has converged to the global optimal solution. Among the 21 benchmark functions, in addition to $f_{13}$, the standard variance obtained by

**Table 2** The parameter settings of algorithms

| Algorithms | Parameter values |
|---|---|
| MFO | During the iteration, $r$ decreases linearly from -2 to -1 |
| SSA | $C_1 = rand, C_2 = rand, C_3 = rand$ |
| PSOGSA | $G_0 = 1, a = 20, C_1 = 0.5, C_2 = 1.5, w\epsilon[0, 1]$ |
| GWO | During the iteration, $\vec{a}$ decreases linearly from 2 to 0 |
| WOA | During the iteration, $\vec{a}$ decreases linearly from 2 to 0 |
| AEO | $r = rand, r_1 = rand, r_2 = rand$ |
| DAEO | $r = rand, r_1 = rand, r_2 = rand$ |

**Table 3** DAEO optimization results on benchmark functions

| ID | Result | Algorithm | | | | | | DAEO |
|---|---|---|---|---|---|---|---|---|
| | | MFO | SSA | PSOGSA | GWO | WOA | AEO | |
| $f_1$ | Mean | 8.91E+03 | 1.41E−06 | 7.72E+03 | 4.48E−34 | 4.38E−115 | 3.87E−257 | **0** |
| | Std | 1.17E+04 | 2.28E−06 | 7.22E+03 | 8.59E−34 | 2.40E−114 | **0** | **0** |
| $f_2$ | Mean | 6.39E+01 | 4.99E+00 | 8.54E+01 | 1.16E−20 | 3.85E−84 | 4.93E−128 | **0** |
| | Std | 2.99E+01 | 2.55E+00 | 1.33E+02 | 5.75E−21 | 1.16E−83 | 2.69E−127 | **0** |
| $f_3$ | Mean | 5.52E+04 | 7.20E+03 | 3.62E+04 | 1.39E−03 | 1.52E+05 | 6.54E−249 | **0** |
| | Std | 2.66E+04 | 3.42E+03 | 1.66E+04 | 7.20E−03 | 3.54E+04 | **0** | **0** |
| $f_4$ | Mean | 8.49E+01 | 1.82E+01 | 7.04E+01 | 3.19E−07 | 6.41E+01 | 4.05E−128 | **0** |
| | Std | 4.34E+00 | 4.09E+00 | 1.44E+01 | 4.11E−07 | 2.78E+01 | 1.77E−127 | **0** |
| $f_5$ | Mean | 2.52E+08 | 2.04E+06 | 3.06E+08 | 7.27E−31 | 1.10E−116 | 2.00E−245 | **0** |
| | Std | 1.76E+08 | 8.81E+06 | 4.60E+08 | 9.47E−31 | 5.86E−116 | **0** | **0** |
| $f_6$ | Mean | 2.65E+01 | 1.20E+01 | 2.69E+01 | 8.58E−01 | 9.32E−01 | 1.07E+00 | **2.47E−01** |
| | Std | 4.40E+00 | 1.85E+00 | 4.34E+00 | 2.18E−01 | 1.37E−01 | 7.81E−02 | **7.54E−02** |
| $f_7$ | Mean | 2.81E+01 | 3.42E−01 | 3.58E−01 | 1.86E−03 | 1.40E−03 | 3.57E−04 | **1.17E−04** |
| | Std | 3.42E+01 | 1.22E−01 | 1.04E−01 | 9.96E−04 | 1.44E−03 | 3.09E−04 | **8.25E−05** |
| $f_8$ | Mean | 1.37E+01 | 3.34E+00 | 1.98E+01 | 2.03E−01 | 1.17E−01 | 3.60E−121 | **0** |
| | Std | 5.53E+00 | 5.77E−01 | 4.84E+00 | 4.14E−02 | 7.46E−02 | 1.96E−120 | 0 |
| $f_9$ | Mean | 3.18E+02 | 9.27E+01 | 2.52E+02 | 9.11E−01 | **0** | **0** | **0** |
| | Std | 5.07E+01 | 2.64E+01 | 3.69E+01 | 1.97E+00 | **0** | **0** | **0** |
| $f_{10}$ | Mean | 1.95E+01 | 3.61E+00 | 1.84E+01 | 5.31E−14 | 4.44E−15 | **8.88E−16** | **8.88E−16** |
| | Std | 5.49E−01 | 1.05E+00 | 1.42E+00 | 6.41E−15 | 2.80E−15 | **0** | **0** |
| $f_{11}$ | Mean | 7.69E+01 | 2.51E−02 | 9.29E+01 | 2.66E−03 | **0** | **0** | **0** |
| | Std | 1.16E+02 | 1.60E−02 | 8.11E+01 | 6.22E−03 | **0** | **0** | **0** |
| $f_{12}$ | Mean | 8.60E+06 | 9.91E+00 | 5.12E+07 | 1.10E−01 | 1.34E−02 | 3.81E−07 | **2.88E−10** |
| | Std | 4.67E+07 | 2.83E+00 | 1.04E+08 | 6.83E−02 | 5.86E−03 | 3.10E−07 | **1.30E−09** |
| $f_{13}$ | Mean | 2.84E+07 | 6.24E+01 | 8.20E+07 | 1.80E+00 | 7.51E−01 | **9.88E−02** | 9.53E−01 |
| | Std | 1.04E+08 | 2.07E+01 | 1.67E+08 | 3.23E−01 | 3.27E−01 | **1.10E−01** | 6.14E−01 |
| $f_{14}$ | Mean | 1.07E+01 | 7.73E+00 | 7.81E+00 | 3.13E−04 | 3.23E−83 | 7.81E−129 | **0** |
| | Std | 1.13E+00 | 2.75E+00 | 3.99E+00 | 3.99E−04 | 1.15E−82 | 4.26E−128 | **0** |
| $f_{15}$ | Mean | 1.04E+03 | 1.54E+02 | 4.65E+02 | 2.84E−03 | 1.67E+03 | 3.60E−228 | **0** |
| | Std | 3.10E+02 | 1.23E+02 | 2.04E+02 | 1.36E−02 | 2.36E+02 | **0** | **0** |
| $f_{16}$ | Mean | 2.71E+00 | 1.03E+00 | 4.25E+00 | 3.87E+00 | 2.96E+00 | **9.98E−01** | **9.98E−01** |
| | Std | 2.04E+00 | 1.81E−01 | 4.63E+00 | 3.79E+00 | 3.21E+00 | **0** | **0** |
| $f_{17}$ | Mean | 1.10E−03 | 7.85E−04 | 7.02E−03 | 5.69E−03 | 7.18E−04 | 3.69E−04 | **3.07E−04** |
| | Std | 4.45E−03 | 2.78E−04 | 1.22E−02 | 9.00E−03 | 3.90E−04 | 2.32E−04 | **1.12E−19** |
| $f_{18}$ | Mean | −6.31E+00 | −7.40E+00 | −5.23E+00 | −9.48E+00 | −8.00E+00 | −9.90E+00 | **−1.02E+01** |
| | Std | 3.33E+00 | 3.50E+00 | 3.18E+00 | 1.75E+00 | 2.67E+00 | 1.37E+00 | **7.25E−15** |
| $f_{19}$ | Mean | −7.55E+00 | −8.87E+00 | −6.71E+00 | −1.00E+01 | −8.05E+00 | −1.02E+01 | **−1.04E+01** |
| | Std | 3.41E+00 | 2.62E+00 | 3.59E+00 | 1.35E+00 | 3.01E+00 | 1.22E+00 | **1.55E−15** |
| $f_{20}$ | Mean | −7.32E+00 | −9.40E+00 | −6.11E+00 | −1.03E+01 | −8.34E+00 | −1.03E+01 | **−1.05E+01** |
| | Std | 3.60E+00 | 2.64E+00 | 3.76E+00 | 1.48E+00 | 2.96E+00 | 1.22E+00 | **1.81E−15** |
| $f_{21}$ | Mean | −3.23E+00 | −3.22E+00 | −3.28E+00 | −3.26E+00 | −3.26E+00 | −3.27E+00 | **−3.31E+00** |
| | Std | 6.33E−02 | 4.65E−02 | 5.99E−02 | 7.15E−02 | 9.63E−02 | 5.92E−02 | **4.11E−02** |

**Table 4** *p*-value results of Wilcoxon.

| Functions | DAEO vs | | | | | |
|---|---|---|---|---|---|---|
| | MFO | SSA | PSOGSA | GWO | WOA | AEO |
| $f_1$ | 1.21E−12 | 1.21E−12 | 1.21E−12 | 1.21E−12 | 1.21E−12 | 1.21E−12 |
| $f_2$ | 1.21E−12 | 1.21E−12 | 1.21E−12 | 1.21E−12 | 1.21E−12 | 1.21E−12 |
| $f_3$ | 1.21E−12 | 1.21E−12 | 1.21E−12 | 1.21E−12 | 1.21E−12 | 1.21E−12 |
| $f_4$ | 1.21E−12 | 1.21E−12 | 1.21E−12 | 1.21E−12 | 1.21E−12 | 1.21E−12 |
| $f_5$ | 1.21E−12 | 1.21E−12 | 1.21E−12 | 1.21E−12 | 1.21E−12 | 1.21E−12 |
| $f_6$ | 3.02E−11 | 3.02E−11 | 3.02E−11 | 3.02E−11 | 3.02E−11 | 3.02E−11 |
| $f_7$ | 3.02E−11 | 3.02E−11 | 3.02E−11 | 3.02E−11 | 5.60E−07 | 9.52E−04 |
| $f_8$ | 1.21E−12 | 1.21E−12 | 1.21E−12 | 1.21E−12 | 1.17E−12 | 1.21E−12 |
| $f_9$ | 1.21E−12 | 1.21E−12 | 1.21E−12 | 5.66E−10 | na | na |
| $f_{10}$ | 1.21E−12 | 1.21E−12 | 1.21E−12 | 1.05E−12 | 3.76E−08 | na |
| $f_{11}$ | 1.21E−12 | 1.21E−12 | 1.21E−12 | 2.16E−02 | na | na |
| $f_{12}$ | 3.02E−11 | 3.02E−11 | 3.02E−11 | 3.02E−11 | 3.02E−11 | 3.02E−11 |
| $f_{13}$ | 3.02E−11 | 3.02E−11 | 3.02E−11 | 3.01E−07 | 1.86E−01 | 6.01E−08 |
| $f_{14}$ | 1.21E−12 | 1.21E−12 | 1.21E−12 | 1.21E−12 | 1.21E−12 | 1.21E−12 |
| $f_{15}$ | 1.21E−12 | 1.21E−12 | 1.21E−12 | 1.21E−12 | 1.21E−12 | 1.21E−12 |
| $f_{16}$ | 5.37E−06 | 5.57E−13 | 2.20E−06 | 1.21E−12 | 1.21E−12 | na |
| $f_{17}$ | 2.82E−11 | 2.82E−11 | 2.82E−11 | 2.83E−11 | 2.83E−11 | 1.25E−03 |
| $f_{18}$ | 8.42E−07 | 1.21E−12 | 1.85E−10 | 1.21E−12 | 1.21E−12 | 4.19E−02 |
| $f_{19}$ | 7.98E−03 | 8.87E−12 | 8.64E−10 | 8.87E−12 | 8.87E−12 | 8.60E−03 |
| $f_{20}$ | 6.86E−05 | 1.21E−12 | 5.96E−10 | 1.21E−12 | 1.21E−12 | 4.19E−02 |
| $f_{21}$ | 5.90E−06 | 5.23E−11 | 5.28E−03 | 5.52E−09 | 1.63E−08 | 1.09E−02 |

DAEO also ranks first, indicating that the DAEO algorithm has strong robustness. To evaluate the statistically meaningful differences between the algorithms, Wilcoxon's nonparametric statistical test with a significance level of 5% was used. In particular, if the *P*-values are less than 5%, the algorithm has a statistically significant advantage. The *P*-values in Table 4 further demonstrate the advantage of DAEO because lots of the *P*-values are much smaller than 0.05.

### 5.2 Experiment 2: DAEO for IIR filter identification

In the segment, the DAEO is applied to the IIR filter identification application to verify the effectiveness of the algorithm. Definitions of the IIR system identification problem set are shown in Table 5 [13, 22]. Usually, they are split into two categories: One is the real-order filter model, and the other is the reduced-order filter model. The six state-of-the-art algorithms, including PSO[15], BA[12], M-ISA[22], RWOA[13], RGWO[44], and LWOA [45], are adopted to compare with the DAEO algorithm.

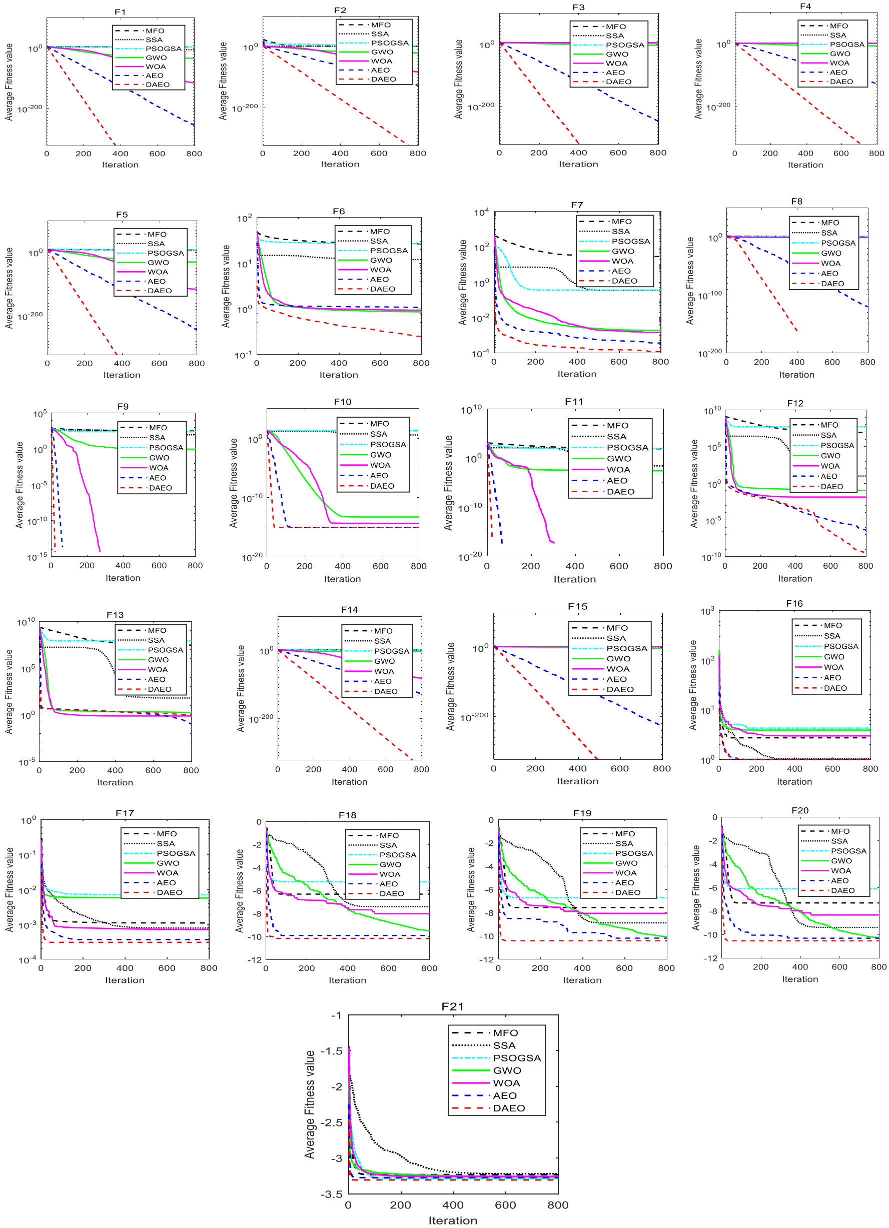


**Fig. 5** Convergence diagrams of benchmark functions $f_1 \sim f_{21}$

### 5.2.1 Parameter setup

In all cases, the input signal is a white sequence, and the number of input samples is 100. The parameters for all algorithms are listed as follows: The population size of

**Table 5** Five IIR system identification problems

| The transfer function | Same order | Reduced order |
|---|---|---|
| $G_p(z) = \frac{0.05-0.4z^{-1}}{1-1.1314z^{-1}+0.25z^{-2}}$ | $G_M(z) = \frac{b_0-b_1z^{-1}}{1-a_1z^{-1}-a_2z^{-2}}$ | $G_M'(z) = \frac{b_0'}{1-a_1'z^{-1}}$ |
| $G_p(z) = \frac{-0.3+0.4z^{-1}-0.5z^{-2}}{1-1.2z^{-1}+0.5z^{-2}-0.1z^{-3}}$ | $G_M(z) = \frac{b_0+b_1z^{-1}+b_2z^{-2}}{1-a_1z^{-1}-a_2z^{-2}-a_3z^{-3}}$ | $G_M'(z) = \frac{b_0'+b_1'z^{-1}}{1-a_1'z^{-1}-a_2'z^{-2}}$ |
| $G_p(z) = \frac{1-0.9z^{-1}+0.81z^{-2}-0.729z^{-3}}{\left(\begin{array}{c}1+0.04z^{-1}+0.2775z^{-2}- \\ 0.2101z^{-3}+0.1401z^{-4}\end{array}\right)}$ | $G_M(z) = \frac{b_0+b_1z^{-1}+b_2z^{-2}+b_3z^{-3}}{\left(\begin{array}{c}1-a_1z^{-1}-a_2z^{-2}- \\ a_3z^{-3}-a_4z^{-4}\end{array}\right)}$ | $G_M'(z) = \frac{b_0'+b_1'z^{-1}+b_2'z^{-2}}{1-a_1'z^{-1}-a_2'z^{-2}-a_3'z^{-3}}$ |
| $G_p(z) = \frac{\left(\begin{array}{c}0.1084+0.5419z^{-1}+ \\ 1.0837z^{-2}+1.0837z^{-3}+ \\ 0.5419z^{-4}+0.1084z^{-5}\end{array}\right)}{\left(\begin{array}{c}1+0.9853z^{-1}+ \\ 0.9738z^{-2}+0.3864z^{-3}+ \\ 0.1112z^{-4}+0.0113z^{-5}\end{array}\right)}$ | $G_M(z) = \frac{\left(\begin{array}{c}b_0+b_1z^{-1}+ \\ b_2z^{-2}+b_3z^{-3}+ \\ b_4z^{-4}+b_5z^{-5}\end{array}\right)}{\left(\begin{array}{c}1-a_1z^{-1}- \\ a_2z^{-2}-a_3z^{-3}- \\ a_4z^{-4}-a_5z^{-5}\end{array}\right)}$ | $G_M'(z) = \frac{\left(\begin{array}{c}b_0'+b_1'z^{-1}+b_2'z^{-2}+ \\ b_3'z^{-3}+b_4'z^{-4}\end{array}\right)}{\left(\begin{array}{c}1-a_1'z^{-1}-a_2'z^{-2}- \\ a_3'z^{-3}-a_4'z^{-4}\end{array}\right)}$ |
| $G_p(z) = \frac{1-0.4z^{-2}-0.65z^{-4}+0.26z^{-6}}{\left(\begin{array}{c}1-0.77z^{-2}- \\ 0.8498z^{-4}+0.6486z^{-6}\end{array}\right)}$ | $G_M(z) = \frac{\left(\begin{array}{c}b_0+b_2z^{-2}+ \\ b_4z^{-4}+b_6z^{-6}\end{array}\right)}{\left(\begin{array}{c}1-a_2z^{-2}- \\ a_4z^{-4}-a_6z^{-6}\end{array}\right)}$ | $G_M'(z) = \frac{\left(\begin{array}{c}b_0'+b_1'z^{-1}+ \\ b_2'z^{-2}+b_3'z^{-3}+ \\ b_4'z^{-4}+b_5'z^{-5}\end{array}\right)}{\left(\begin{array}{c}1-a_1'z^{-1}- \\ a_2'z^{-2}-a_3'z^{-3}- \\ a_4'z^{-4}-a_5'z^{-5}\end{array}\right)}$ |

**Table 6** The parameter settings of algorithms

| Algorithms | Parameter values |
|---|---|
| PSO | $C_1 = rand, C_2 = rand, C_3 = rand$ |
| BA | $A_0 = 0.9, r_0 = 0.5$ |
| M-ISA | $a = 0.2$ |
| LWOA | During the iteration, $\vec{a}$ decreases linearly from 2 to 0 |
| RWOA | During the iteration, $\vec{a}$ decreases linearly from 2 to 0. F = 0.7 |
| RGWO | During the iteration, $\vec{a}$ decreases linearly from 2 to 0. F = 0.7 |
| DAEO | $r = rand, r_1 = rand, r_2 = rand$ |

each algorithm is 30, and the total iteration times are 500. The main control parameter settings for adopted algorithms are set in Table 6. In addition, to remove the impact of the randomness of the algorithm, each algorithm is executed 30 times independently.

### 5.2.2 Experimental results of system identification

Five various model cases are applied to the simulation experiment. Tables 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 and 26 exhibit the simulation results, which include the parameter estimates and MSE values for identification.

In these tables, “Parameter value” indicates the parameter value of system identification obtained by different algorithms. “Actual values” represent the true value of the variable to be evaluated. “Parameters” represent the variables that need to be evaluated in the test example. The objective function for system identification is the MSE described in Sect. 2 and is used as a metric. The performance analysis for all algorithms is ensured by determining the best, worst, mean, and standard variance of the MSE for the unknown system. These values are calculated by running the values obtained from 30 simulations for each example using the algorithm. For ease of viewing, in all tested examples, the best results obtained by the identification algorithm in terms of each adopted performance metric are marked in bold in the respective tables.

Model 1 For the first test example, a second-order system is considered, whose transport function is expressed by Eq. (16).

**Table 7** Parameter values of all algorithms for model 1 (case 1)

| Parameters | Actual values | Parameter value | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | PSO | BA | M-ISA | LWOA | RWOA | RGWO | DAEO |
| $a_1$ | 1.1314 | 1.3199 | 0.2402 | **1.1314** | 1.1129 | 1.1774 | 1.1301 | **1.1314** |
| $a_2$ | −0.2500 | −0.4304 | −0.5566 | **−0.2500** | −0.2308 | −0.2915 | −0.2489 | **−0.2500** |
| $b_0$ | 0.0500 | 0.0372 | 0.2156 | **0.0500** | 0.0515 | 0.0322 | 0.0505 | **0.0500** |
| $b_1$ | −0.4000 | −0.2615 | −0.1239 | **−0.4000** | −0.4078 | −0.3661 | −0.4010 | **−0.4000** |

**Table 8** MSE values of all algorithms for model 1 (case 1)

| Algorithms | MSE value | | | |
|---|---|---|---|---|
| | Best | Worst | Mean | Std |
| PSO | 2.69E−03 | 1.60E−02 | 8.06E−03 | 3.14E−03 |
| BA | 9.72E−03 | 2.05E−01 | 6.68E−02 | 4.68E−02 |
| M-ISA | 6.07E−33 | 2.70E−03 | 3.34E−04 | 8.09E−04 |
| LWOA | 1.05E−05 | 5.38E−03 | 1.40E−03 | 1.37E−03 |
| RWOA | 4.88E−05 | 1.97E−02 | 3.42E−03 | 5.40E−03 |
| RGWO | 7.47E−08 | 1.84E−04 | 5.91E−05 | 6.96E−05 |
| DAEO | **0** | **0** | **0** | **0** |

**Table 9** Parameter values of all algorithms for model 1 (case 2)

| Parameters | Parameter value | | | | | | |
|---|---|---|---|---|---|---|---|
| | PSO | BA | M-ISA | LWOA | RWOA | RGWO | DAEO |
| $a_1$ | 0.9167 | 0.9113 | 0.9033 | 0.9183 | 0.9313 | 0.9249 | 0.9053 |
| $b_1$ | − 0.2052 | − 0.2838 | − 0.2733 | − 0.2483 | − 0.2755 | − 0.2198 | − 0.3028 |

**Table 10** MSE values of all algorithms for model 1 (case 2)

| Algorithms | MSE value | | | |
|---|---|---|---|---|
| | Best | Worst | Mean | Std |
| PSO | 1.11E−02 | 1.88E−02 | 1.61E−02 | 1.75E−03 |
| BA | 1.03E−02 | 3.62E−02 | 2.06E−02 | 6.50E−03 |
| M-ISA | 8.80E−03 | 1.80E−02 | 1.13E−02 | 1.68E−03 |
| LWOA | 1.22E−02 | 1.66E−02 | 1.46E−02 | 1.00E−03 |
| RWOA | 9.72E−03 | 1.94E−02 | 1.31E−02 | 2.20E−03 |
| RGWO | 1.00E−02 | 1.79E−02 | 1.20E−02 | 1.31E−03 |
| DAEO | **8.60E−03** | **1.12E−02** | **1.02E−02** | **5.56E−04** |

**Table 11** Parameter values of all algorithms for model 2 (case 1)

| Parameters | Actual values | Parameter value | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | PSO | BA | M-ISA | LWOA | RWOA | RGWO | DAEO |
| $a_1$ | 1.2000 | 0.8688 | – 0.1354 | 1.1992 | 1.0571 | 1.1026 | 1.1942 | **1.2000** |
| $a_2$ | – 0.5000 | – 0.1945 | 0.4146 | – 0.4986 | – 0.3422 | – 0.2758 | – 0.4885 | **– 0.5000** |
| $a_3$ | 0.1000 | 0.1373 | 0.2481 | 0.0993 | 0.0746 | – 0.0277 | 0.0932 | **0.1000** |
| $b_0$ | – 0.3000 | – 0.2966 | – 0.2736 | – 0.2999 | – 0.3198 | – 0.2959 | – 0.2991 | **– 0.3000** |
| $b_1$ | 0.4000 | 0.2845 | – 0.0400 | 0.3998 | 0.4590 | 0.3320 | 0.3974 | **0.4000** |
| $b_2$ | – 0.5000 | – 0.4961 | – 0.2006 | – 0.4998 | – 0.5808 | – 0.4242 | – 0.4982 | **– 0.5000** |

**Table 12** MSE values of all algorithms for model 2 (case 1)

| Algorithms | MSE value | | | |
|---|---|---|---|---|
| | Best | Worst | Mean | Std |
| PSO | 4.90E−03 | 3.97E-02 | 2.38E-02 | 7.95E-03 |
| BA | 1.04E−02 | 1.56E+00 | 2.71E-01 | 2.82E-01 |
| M-ISA | 8.05E−09 | 1.19E-02 | 2.08E-03 | 2.96E-03 |
| LWOA | 1.00E−03 | 1.55E-02 | 6.68E-03 | 3.93E-03 |
| RWOA | 3.14E−04 | 1.65E-02 | 5.65E-03 | 4.20E-03 |
| RGWO | 1.20E−06 | 5.64E-03 | 5.93E-04 | 1.07E-03 |
| DAEO | **1.30E−24** | **4.49E-09** | **2.13E-10** | **8.66E-10** |

**Table 13** Parameter values of all algorithms for Test model 2 (case 2)

| Parameters | Parameter value | | | | | | |
|---|---|---|---|---|---|---|---|
| | PSO | BA | M-ISA | LWOA | RWOA | RGWO | DAEO |
| $a_1$ | 0.2000 | – 1.1932 | 0.4453 | 0.4199 | 0.2539 | 0.2903 | 0.2886 |
| $a_2$ | 0.6001 | – 0.2807 | 0.3937 | 0.4112 | 0.5471 | 0.5169 | 0.5170 |
| $b_0$ | – 0.3784 | – 0.5083 | – 0.3957 | – 0.3699 | – 0.4011 | – 0.3421 | – 0.4108 |
| $b_1$ | – 0.0797 | – 0.3860 | – 0.0190 | 0.0245 | 0.0554 | – 0.0531 | 0.0794 |

**Table 14** MSE values of all algorithms for model 2 (case 2)

| Algorithms | MSE value | | | |
|---|---|---|---|---|
| | Best | Worst | Mean | Std |
| PSO | 1.22E− 02 | 2.24E− 02 | 1.73E− 02 | 2.52E− 03 |
| BA | 1.71E− 02 | 2.75E− 01 | 7.63E− 02 | 6.56E− 02 |
| M-ISA | 6.39E− 03 | 1.31E− 02 | 8.54E− 03 | 1.31E− 03 |
| LWOA | 9.96E− 03 | 1.62E− 02 | 1.24E− 02 | 1.56E− 03 |
| RWOA | 7.08E− 03 | 1.69E− 02 | 1.01E− 02 | 2.33E− 03 |
| RGWO | 6.76E− 03 | 9.78E− 03 | 8.08E− 03 | 6.89E− 04 |
| DAEO | **5.99E− 03** | **8.27E− 03** | **7.09E− 03** | **5.66E− 04** |

**Table 15** Parameter values of all algorithms for model 3 (case 1)

| Parameters | Actual values | Parameter value | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | PSO | BA | M-ISA | LWOA | RWOA | RGWO | DAEO |
| $a_1$ | − 0.0400 | − 0.1438 | 0.3187 | − 0.0386 | − 0.0750 | − 0.0506 | − 0.0266 | **− 0.0400** |
| $a_2$ | − 0.2775 | − 0.5783 | 0.4475 | − 0.2771 | − 0.4555 | − 0.2410 | − 0.2526 | **− 0.2775** |
| $a_3$ | 0.2101 | − 0.0322 | 0.2692 | 0.2099 | 0.0818 | 0.3111 | 0.2325 | **0.2101** |
| $a_4$ | − 0.1400 | − 0.2661 | − 0.2061 | − 0.1407 | − 0.1776 | − 0.0481 | − 0.1311 | **− 0.1400** |
| $b_0$ | 1.0000 | 0.4586 | 0.3551 | 1.0000 | 1.0321 | 0.9916 | 0.9989 | **1.0000** |
| $b_1$ | − 0.9000 | − 0.8491 | − 1.6274 | − 0.9015 | − 0.8740 | − 0.8745 | − 0.9129 | **− 0.9000** |
| $b_2$ | 0.8100 | 0.5771 | 1.7131 | 0.8109 | 1.0132 | 0.7789 | 0.7986 | **0.8100** |
| $b_3$ | − 0.7290 | − 0.7051 | − 0.3566 | − 0.7291 | − 0.7577 | − 0.8027 | − 0.7349 | **− 0.7290** |

**Table 16** MSE values of algorithms for model 3 (case 1)

| Algorithms | MSE value | | | |
|---|---|---|---|---|
| | Best | Worst | Mean | Std |
| PSO | 2.56E− 02 | 1.12E− 01 | 5.73E− 02 | 1.94E− 02 |
| BA | 9.44E− 02 | 1.41E+03 | 4.97E+01 | 2.56E+02 |
| M-ISA | 3.32E− 08 | 6.28E− 03 | 1.28E− 03 | 1.63E− 03 |
| LWOA | 1.48E− 03 | 4.48E− 02 | 1.28E− 02 | 1.06E− 02 |
| RWOA | 4.27E− 04 | 4.86E− 02 | 7.37E− 03 | 9.88E− 03 |
| RGWO | 1.07E− 05 | 6.46E− 03 | 6.10E− 04 | 1.21E− 03 |
| DAEO | **2.17E− 11** | **9.91E− 04** | **7.91E− 05** | **2.12E− 04** |

$$G_p(z) = \frac{0.05 - 0.4Z^{-1}}{1 - 1.1314Z^{-1} + 0.25Z^{-2}} \tag{16}$$

In this instance, $G_p(z)$ is modeled to demonstrate the superior performance of the DAEO using the same-order system described in case 1 and the reduced-order system in case 2.

**Table 17** Parameter values of all algorithms for model 3(case 2)

| Parameters | Parameter value | | | | | | |
|---|---|---|---|---|---|---|---|
| | PSO | BA | M-ISA | LWOA | RWOA | RGWO | DAEO |
| $a_1$ | – 0.6799 | – 1.5618 | – 0.7769 | – 0.7882 | – 0.9747 | – 0.8368 | – 0.7798 |
| $a_2$ | – 0.5357 | – 0.8543 | – 0.5756 | – 0.6615 | – 0.8153 | – 0.6198 | – 0.5939 |
| $a_3$ | – 0.0378 | – 0.2499 | 0.0070 | – 0.0702 | – 0.1177 | – 0.0028 | – 0.0011 |
| $b_0$ | 1.2163 | 1.6016 | 1.0475 | 0.9102 | 1.0578 | 0.9808 | 0.9629 |
| $b_1$ | – 0.0234 | 1.2142 | – 0.0496 | – 0.1496 | 0.1455 | – 0.0337 | – 0.075 |
| $b_2$ | 0.9786 | – 0.3570 | 0.6225 | 0.6977 | 0.7091 | 0.6020 | 0.6710 |

**Table 18** MSE values of all algorithms for model 3 (case 2)

| Algorithms | MSE value | | | |
|---|---|---|---|---|
| | Best | Worst | Mean | Std |
| PSO | 1.68E– 02 | 7.12E– 02 | 4.46E– 02 | 1.36E– 02 |
| BA | 4.32E– 02 | 2.00E+00 | 3.86E– 01 | 3.95E– 01 |
| M-ISA | 3.08E– 03 | 1.10E– 02 | 7.02E– 03 | 2.43E– 03 |
| LWOA | 7.45E– 03 | 2.58E– 02 | 1.59E– 02 | 4.80E– 03 |
| RWOA | 4.00E– 03 | 4.22E– 02 | 1.05E– 02 | 7.15E– 03 |
| RGWO | 3.70E– 03 | 1.08E– 02 | 8.15E– 03 | 2.83E– 03 |
| DAEO | **2.65E– 03** | **9.00E– 03** | **4.12E– 03** | **1.65E– 03** |

**Table 19** Parameter values of all algorithms for model 4 (case 1)

| Parameters | Actual values | Parameter value | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | PSO | BA | M-ISA | LWOA | RWOA | RGWO | DAEO |
| $a_1$ | – 0.9853 | – 0.0291 | – 0.4841 | – 0.1492 | – 0.1586 | – 0.1151 | 0.1452 | 0.1407 |
| $a_2$ | – 0.9738 | 0.5316 | – 0.1169 | – 0.4186 | – 0.3379 | – 0.3208 | – 0.3622 | – 0.3364 |
| $a_3$ | – 0.3864 | 0.4575 | – 0.1338 | 0.1705 | – 0.1080 | 0.3230 | 0.3801 | 0.3688 |
| $a_4$ | – 0.1112 | – 0.0943 | 0.0986 | 0.0167 | 0.0619 | 0.0293 | – 0.0180 | – 2.5000 |
| $a_5$ | – 0.0113 | 0.1354 | – 0.1853 | 0.0079 | – 0.0699 | 0.0289 | 0.0378 | 0.0321 |
| $b_0$ | 0.1084 | – 0.3156 | – 1.3792 | 0.1096 | 0.1456 | 0.1203 | 0.1077 | 0.1079 |
| $b_1$ | 0.5419 | 0.2126 | 0.7858 | 0.4508 | 0.4468 | 0.4453 | 0.4197 | 0.4210 |
| $b_2$ | 1.0837 | – 0.2747 | 1.2736 | 0.6595 | 0.6689 | 0.6583 | 0.5251 | 0.5247 |
| $b_3$ | 1.0837 | – 0.0941 | 1.6597 | 0.3211 | 0.3538 | 0.2169 | 0.1135 | 0.1063 |
| $b_4$ | 0.5419 | 0.1508 | 0.1532 | – 0.0833 | – 0.0271 | – 0.1794 | – 0.2252 | – 0.2370 |
| $b_5$ | 0.1084 | 0.3264 | – 0.8332 | – 0.0875 | 0.0610 | – 0.2058 | – 0.1244 | – 0.1273 |

Case 1.

The second-order filter can simulate the real second-order system well. Hence, the transport function of the adaptive filter is expressed by Eq. (17).

**Table 20** MSE values of all algorithms for model 4 (case 1)

| Algorithms | MSE value | | | |
|---|---|---|---|---|
| | Best | Worst | Mean | Std |
| PSO | 1.35E−01 | 3.88E−01 | 2.43E−01 | 6.60E−02 |
| BA | 2.23E−01 | 8.02E+11 | 2.68E+10 | 1.46E+11 |
| M-ISA | 2.61E−06 | 5.58E−03 | 4.83E−04 | 1.08E−03 |
| LWOA | 8.83E−04 | 3.55E−01 | 4.09E−02 | 7.73E−02 |
| RWOA | 1.21E−04 | 8.65E−03 | 2.20E−03 | 2.18E−03 |
| RGWO | 1.15E−06 | 8.98E−04 | 2.02E−04 | 2.46E−04 |
| DAEO | **8.32E−07** | **2.30E−04** | **3.58E−05** | **5.84E−05** |

**Table 21** Parameter values of all algorithms for model 4 (case 2)

| Parameters | Parameter value | | | | | | |
|---|---|---|---|---|---|---|---|
| | PSO | BA | M-ISA | LWOA | RWOA | RGWO | DAEO |
| $a_1$ | 0.7394 | −0.3448 | 0.0188 | −0.4972 | 0.0052 | −0.0261 | −0.1143 |
| $a_2$ | 0.7371 | 0.3383 | −0.4824 | −0.4204 | −0.5614 | −0.4994 | −0.5407 |
| $a_3$ | −0.6376 | 0.9854 | 0.1656 | −0.1229 | 0.1619 | 0.1560 | 0.1135 |
| $a_4$ | −0.0385 | 0.1567 | −0.0406 | 0.1471 | −0.0421 | −0.0419 | −0.0515 |
| $b_0$ | −0.0217 | 0.2869 | 0.1027 | 0.1210 | 0.1219 | 0.1030 | 0.1067 |
| $b_1$ | 0.7162 | −0.5411 | 0.4381 | 0.4665 | 0.4347 | 0.4440 | 0.4477 |
| $b_2$ | −0.3317 | −0.0274 | 0.5832 | 0.8194 | 0.6182 | 0.6066 | 0.6533 |
| $b_3$ | −0.7868 | −0.0048 | 0.2606 | 0.4901 | 0.3022 | 0.3002 | 0.3663 |
| $b_4$ | 0.5699 | 0.3098 | −0.0684 | 0.0698 | 0.0047 | −0.0511 | −0.0050 |

**Table 22** MSE values of all algorithms for model 4 (case 2)

| Algorithms | MSE value | | | |
|---|---|---|---|---|
| | Best | Worst | Mean | Std |
| PSO | 3.99E−02 | 1.74E−01 | 9.73E−02 | 3.43E−02 |
| BA | 8.04E−02 | 8.04E+01 | 5.21E+00 | 1.56E+01 |
| M-ISA | 4.25E−05 | 2.24E−03 | 3.18E−04 | 3.91E−04 |
| LWOA | 9.23E−04 | 1.35E−01 | 1.10E−02 | 2.49E−02 |
| RWOA | 6.24E−05 | 1.75E−02 | 2.08E−03 | 3.71E−03 |
| RGWO | 4.08E−05 | 9.76E−04 | 2.85E−04 | 2.41E−04 |
| DAEO | **1.69E−05** | **1.46E−04** | **5.12E−05** | **3.35E−05** |

$$G_M(z) = \frac{b_0 - b_1 Z^{-1}}{1 - a_1 Z^{-1} - a_2 Z^{-2}} \tag{17}$$

The problem of system identification is simplified to optimize the molecule and denominator parameters $b_0$, $b_1$ and $a_1$, $a_2$, separately. The obtained parameters are

**Table 23** Parameter values of all algorithms for model 5 (case 1)

| Parameters | Actual values | Parameter value | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | PSO | BA | M-ISA | LWOA | RWOA | RGWO | DAEO |
| $a_2$ | 0.7700 | − 0.1423 | 0.1413 | 0.2818 | − 0.3091 | − 0.2218 | 0.0539 | − 0.0203 |
| $a_4$ | 0.8498 | 0.6775 | 0.1177 | 0.2776 | 0.6880 | 0.7674 | 0.8173 | 0.8335 |
| $a_6$ | − 0.6486 | − 0.1917 | 0.3717 | 0.3108 | 0.4115 | 0.2694 | 0.0063 | 0.0266 |
| $b_0$ | 1.0000 | 0.5820 | 0.8524 | 0.9797 | 0.9344 | 0.9760 | 0.9973 | 0.9958 |
| $b_2$ | − 0.4000 | 0.4619 | 0.4517 | 0.0902 | 0.6132 | 0.5820 | 0.3233 | 0.3892 |
| $b_4$ | − 0.6500 | 0.1057 | − 0.0393 | 0.098 | − 0.1143 | − 0.2065 | − 0.3550 | − 0.3518 |
| $b_6$ | 0.2600 | 0.1904 | − 0.3525 | − 0.2352 | − 0.2119 | − 0.1498 | − 0.0678 | − 0.0320 |

**Table 24** MSE values of all algorithms for model 5 (case 1)

| Algorithms | MSE value | | | |
|---|---|---|---|---|
| | Best | Worst | Mean | Std |
| PSO | 3.00E− 02 | 1.12E− 01 | 5.77E− 02 | 1.81E− 02 |
| BA | 1.36E− 02 | 4.98E+00 | 5.18E− 01 | 8.73E− 01 |
| M-ISA | 5.15E− 04 | 1.56E− 03 | 8.20E− 04 | **2.19E− 04** |
| LWOA | 2.21E− 03 | 4.41E− 02 | 1.33E− 02 | 1.01E− 02 |
| RWOA | 2.55E− 04 | 1.55E− 02 | 3.75E− 03 | 4.16E− 03 |
| RGWO | 1.50E− 04 | 2.91E− 03 | 8.55E− 04 | 4.90E− 04 |
| DAEO | **3.74E− 05** | **7.75E− 04** | **3.73E− 04** | 2.60E− 04 |

**Table 25** Parameter values of all algorithms for model 5 (case 2)

| Parameters | Parameter value | | | | | | |
|---|---|---|---|---|---|---|---|
| | PSO | BA | M-ISA | LWOA | RWOA | RGWO | DAEO |
| $a_1$ | 0.5442 | 0.9498 | 0.0402 | − 0.4705 | 0.1964 | − 0.0094 | 0.00063 |
| $a_2$ | 0.3044 | 0.1368 | 0.3150 | 0.0562 | 0.0755 | 0.0536 | 0.6398 |
| $a_3$ | 0.6206 | − 0.8105 | − 0.0357 | 0.4809 | 0.2559 | 0.1121 | 0.0529 |
| $a_4$ | − 0.1435 | 0.5154 | 0.4072 | 0.1626 | 0.1462 | 0.6248 | 0.1811 |
| $a_5$ | − 0.3715 | − 0.0806 | 0.0666 | 0.4437 | 0.1200 | − 0.0212 | − 0.0260 |
| $b_0$ | 1.2539 | 1.4369 | 1.0020 | 0.9195 | 0.9448 | 1.0022 | 0.9981 |
| $b_1$ | − 0.4733 | 0.6104 | 0.3231 | 0.7440 | 0.1403 | 0.3711 | 0.3711 |
| $b_2$ | − 0.3784 | − 0.2745 | 0.1569 | 0.5992 | 0.2964 | 0.4326 | − 0.1544 |
| $b_3$ | − 1.0655 | − 0.1902 | 0.1894 | 0.2449 | 0.0528 | 0.1554 | 0.0021 |
| $b_4$ | 0.7947 | 0.1618 | − 0.1642 | 0.1440 | 0.0073 | − 0.2794 | − 0.1193 |
| $b_5$ | 0.3012 | 0.4951 | − 0.0558 | − 0.3467 | − 0.0756 | − 0.0255 | 0.0104 |

shown in Table 7, which led to the optimal approximation to the unknown system using the swarm intelligence algorithm. As can be seen from Table 7, the values of the parameters obtained by the DAEO and M-ISA algorithms are in perfect

**Table 26** MSE values of all algorithms for model 5 (case 2)

| Algorithms | MSE value | | | |
|---|---|---|---|---|
| | Best | Worst | Mean | Std |
| PSO | 7.73E− 02 | 3.68E− 01 | 2.45E− 01 | 7.79E− 02 |
| BA | 1.79E− 01 | 4.92E+08 | 1.65E+07 | 8.97E+07 |
| M-ISA | 7.28E− 05 | 1.92E− 02 | 2.24E− 03 | 3.60E− 03 |
| LWOA | 5.29E− 03 | 3.49E− 01 | 5.06E− 02 | 6.60E− 02 |
| RWOA | 1.40E− 03 | 3.66E− 02 | 5.97E− 03 | 6.63E− 03 |
| RGWO | 9.26E− 05 | 3.88E− 03 | 8.10E− 04 | 7.83E− 04 |
| DAEO | **2.00E− 05** | **6.18E− 04** | **2.10E− 04** | **1.38E− 04** |

agreement with the actual values of the system parameters compared to the other algorithms. In Table 8, the MSE values obtained by the DAEO algorithm are all 0, which is better than those acquired by other algorithms. The best MSE values obtained are 2.69E-03, 9.72E-03, 6.07E-33, 1.05E-05, 4.88E-05, 7.47E-08, and 0 for PSO, BA, M-ISA, LWOA, RWOA, RGWO, and DAEO, respectively. From the above discussion, it can be seen that the DAEO-based system identification approach produces the best results in terms of MSE compared to PSO, BA, M-ISA, LWOA, RWOA, and RGWO.

Figure 6 shows the convergence of the mean values of MSE using different algorithms. It can be seen from Fig. 6 that DAEO requires 400 iterations to obtain the minimum value of MSE, while other algorithms fall into local optimum prematurely. Furthermore, the convergence speed of DAEO is much faster than that of the other six algorithms. From Fig. 7 and the variance values in Table 8, it can be concluded that DAEO has better robustness compared to other algorithms. In terms of the mean of MSE, the performance ranking of all algorithms is DAEO > RGWO > M-ISA > LWOA > RWOA > PSO > BA.

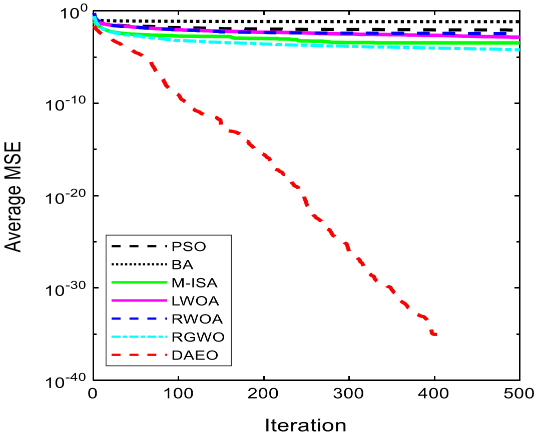
Case 2.

The second-order system can also be simulated by the first-order filter with the following transport function:

$$G_M'(z) = \frac{b_0'}{1 - a_1' Z^{-1}} \tag{18}$$

A reduced-order filter is adopted to recognize the unbeknown system. In this case, there is no accurate value but the approximate value. Here, MSE and convergence curves are used to evaluate performance metrics for the identification problem of the reduced-order system. Statistical results are used to analyze the performance of PSO, BA, M-ISA, LWOA, RWOA, RGWO, and DAEO. The MSE values obtained for all algorithms are listed in Table 10. The mean values of MSE for PSO, BA, M-ISA, LWOA, RWOA, RGWO, and DAEO are 1.61E−02, 2.06E−02, 1.13E−02, 1.46E−02, 1.31E−02, 1.20E−02, and 1.02E−02, respectively. From Table 10, it can be concluded that DAEO outperforms PSO, BA, M-ISA, LWOA, RWOA, and RGWO when modeled using the reduced-order system. From the MSE values listed

**Fig. 6** Converge diagram for model 1, case 1



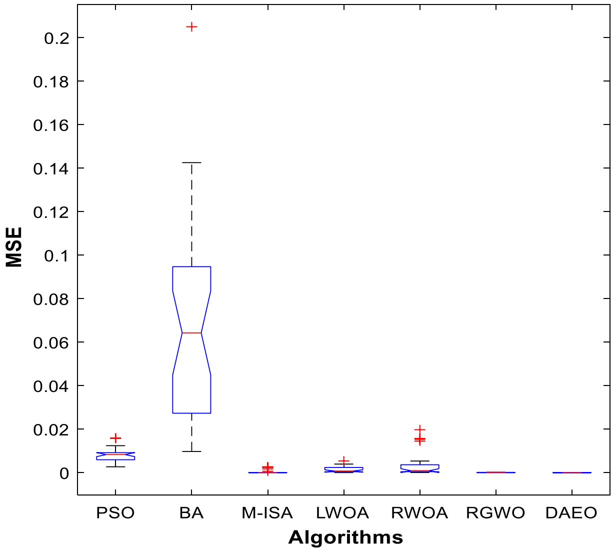
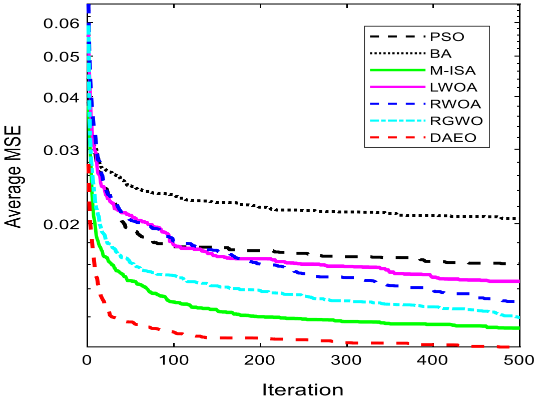


**Fig. 7** ANOVA graph for model 1, case 1

in Table 10, it can be concluded that DAEO obtains a better approximation for the second-order IIR system compared to other algorithms. Figure 8 describes the convergence behavior of the MSE values using various algorithms. As can be seen from Fig. 8, DAEO has reached convergence in 300 iterations with the mean value of about 1.02E-02. From the variance values in Table 10 and Fig. 9, it can be obtained that DAEO has a small variance value indicating that DAEO has better robustness. Therefore, in terms of the mean of MSE, the performance of these algorithms is ranked as DAEO > M-ISA > RGWO > RWOA > LWOA > PSO > BA.

**Fig. 8** Converge diagram for model 1, case 2



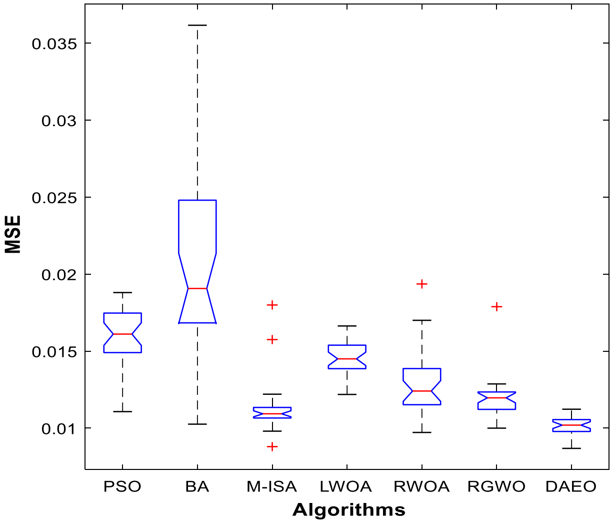


**Fig. 9** ANOVA graph for model 1, case 2

Model 2 For the second test example, the transport function of the third-order system is defined as follow:

$$G_p(z) = \frac{-0.3 + 0.4Z^{-1} - 0.5Z^{-2}}{1 - 1.2Z^{-1} + 0.5Z^{-2} - 0.1Z^{-3}} \tag{19}$$

Case 1.

The third-order filter can simulate the real third-order system well. Hence, the transport function of the filter is expressed by Eq. (20).

$$G_M(z) = \frac{b_0 + b_1 Z^{-1} + b_2 Z^{-2}}{1 - a_1 Z^{-1} - a_2 Z^{-2} - a_3 Z^{-3}} \tag{20}$$

In this example, optimization of the system parameters, $b_0$,$b_1$,$b_2$,$a_1$,$a_2$,$a_3$, is performed using PSO, BA, M-ISA, LWOA, RWOA, RGWO, and DAEO. The estimated parameters are listed in Table 11. It can be observed from Table 11 that the parameter values obtained with DAEO are in perfect agreement with the actual parameter values compared to other algorithms. Figure 10 depicts the convergence graph of the mean value of MSE for all algorithms. It can be seen that the convergence curve of DAEO keeps decreasing after the 100th iteration, while other algorithms fall into local optima at the beginning of the iteration. In addition, the MSE values obtained by all algorithms are listed in Table 12. The mean values of MSE obtained for PSO, BA, M-ISA, LWOA, RWOA, RGWO, and DAEO are 2.38E-02, 2.71E-01, 2.08E-03, 6.68E-03, 5.65E-03, 5.93E-04, and 2.13E-10, respectively. From these measurements, it can be concluded that the system identified using the DAEO method has the smallest MSE and can effectively handle the system identification problem. Based on the MSE values in Table 12, it can be finally inferred that the use of DAEO to identify the third-order IIR system model outperforms other existing comparative algorithms. The variance diagram also displays that the DAEO algorithm has better stability (Fig. 11). The performance of the algorithms adopted can be ranked as follows: DAEO > RGWO > M-ISA > RWOA > LWOA > PSO > BA.

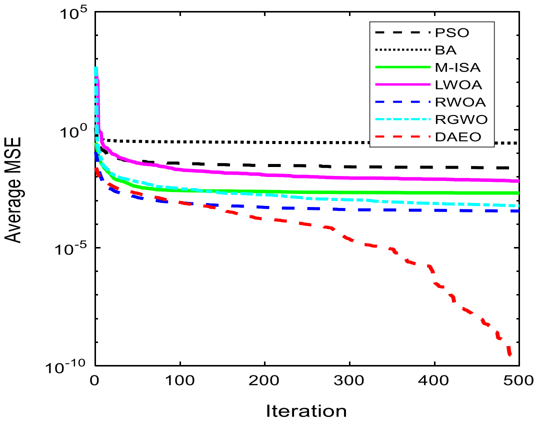
Case 2.

The third-order system can also be simulated by the second-order filter with the following transport function:

$$G'_M(z) = \frac{b'_0 + b'_1 Z^{-1}}{1 - a'_1 Z^{-1} - a'_2 Z^{-2}} \tag{21}$$

Table 13 shows the approximate solutions of the parameters obtained by all the algorithms. The MSE values of the system generated by the optimized parameters are shown in Table 14. The MSE values obtained using DAEO are better than any

**Fig. 10** Converge diagram for model 2, case 1



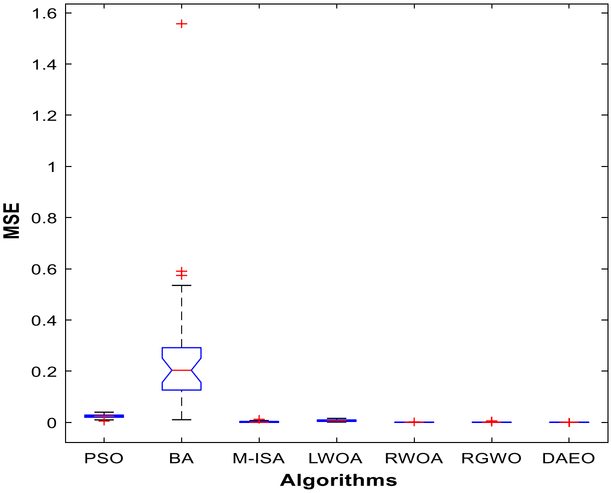


**Fig. 11** ANOVA graph for model 2, case 1

other algorithm. From the MSE values obtained in Table 14, it can be concluded that the DAEO-based system identification is superior to the other compared algorithms. The excellent performance of DAEO is also well demonstrated in Figs. 12, and 13, where DAEO has good convergence speed and accuracy compared to other algorithms, and the BA traps in the local optimal solution prematurely. It is well suited for achieving optimal MSE values in other algorithms. According to the performance of the system identification, the order of the algorithms is DAEO > RGW O > M-ISA > RWOA > LWOA > PSO > BA.

Model 3 For the third test example, the transport function of the fourth-order system is expressed by Eq. (22).

$$G_p(z) = \frac{1 - 0.9Z^{-1} + 0.81Z^{-2} - 0.729Z^{-3}}{1 + 0.04Z^{-1} + 0.2775Z^{-2} - 0.2101Z^{-3} + 0.14Z^{-4}} \tag{22}$$

Case 1.

The fourth-order filter can simulate the real fourth-order system well. Hence, the transport function of the filter is expressed by Eq. (23).

$$G_M(z) = \frac{b_0 + b_1Z^{-1} + b_2Z^{-2} + b_3Z^{-3}}{1 - a_1Z^{-1} - a_2Z^{-2} - a_3Z^{-3} - a_4Z^{-4}} \tag{23}$$

The problem of system identification is simplified to optimize the molecule and denominator parameters $b_0$,$b_1$,$b_2$, $b_3$ and $a_1$, $a_2$, $a_3$, $a_4$, separately. Table 15 shows the best parameters obtained for the optimization of the error surface of the unknown system using different evolutionary algorithms. As can be seen from Table 15, the parameters obtained by DAEO are in full agreement with the actual parameter values of the system compared to other algorithms. Table 16 shows the MSE values of the fourth-order system identification problem. The mean values of MSE obtained

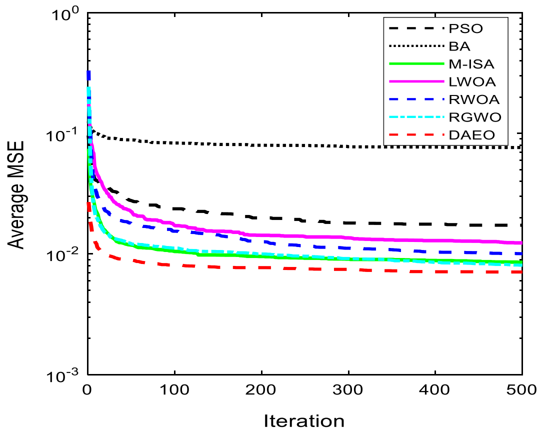


**Fig. 12** Converge diagram for model 2, case 2

are 5.73E-02, 4.97E+01, 1.28E-03, 1.28E-02, 7.37E-03, 6.10E-04, and 5.65E-05 for PSO, BA, M-ISA, LWOA, RWOA, RGWO, and DAEO, respectively. From the above numerical results, it can be seen that the proposed DAEO-based system identification approach obtains the optimal results in terms of MSE. Figure 14 shows the convergence curves of the mean values for MSE, which DAEO has a better convergence accuracy. In Fig. 15, most of the algorithms show strong robustness. Therefore, for the optimization of the parameters of the fourth-order IIR system, the performance ranking of the adopted algorithm is DAEO > RGWO > M-ISA > RWOA > LWOA > PSO > BA.

Case 2.

The fourth-order system can also be simulated by the third-order filter with the following transport function:

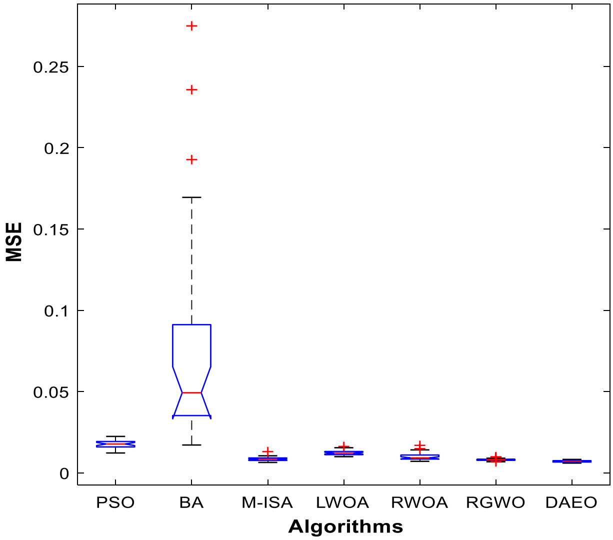
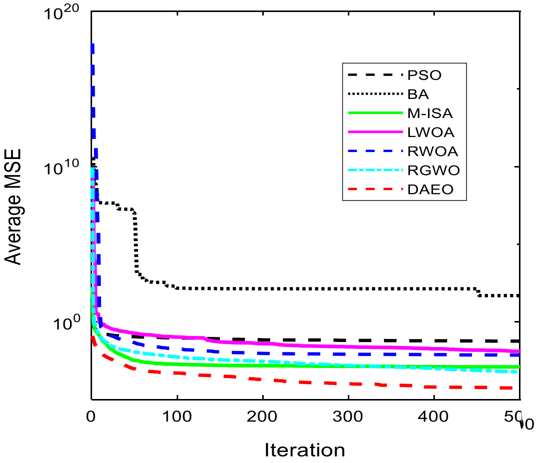


**Fig. 13** ANOVA graph for model 2, case 2

**Fig. 14** Converge diagram for model 3, case 1



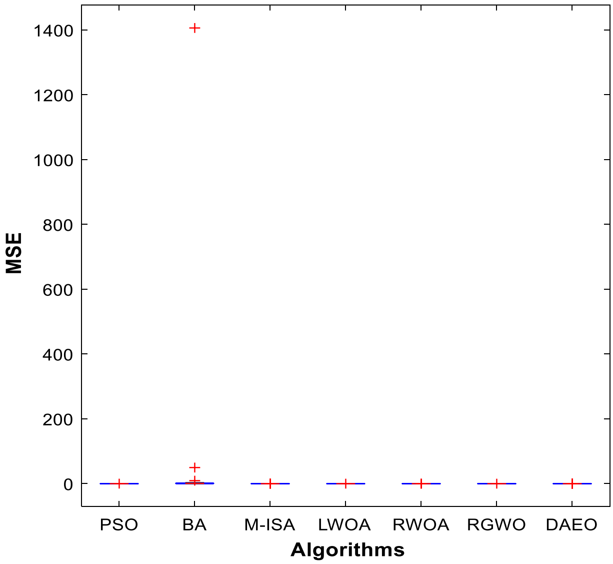


**Fig. 15** ANOVA graph for model 3, case 1

$$G'_M(z) = \frac{b'_0 + b'_1 Z^{-1} + b'_2 Z^{-2}}{1 - a'_1 Z^{-1} - a'_2 Z^{-2} - a'_3 Z^{-3}} \tag{24}$$

Table 17 shows the approximate solutions of the parameters obtained by all the algorithms. The exact parameters are not available since the reduced-order filter model is used. The MSE values are presented in Table 18. The observed mean values of MSE are 4.46E-02, 3.86E-01, 7.02E-03, 1.59E-02, 1.05E-02, 8.15E-03, and 4.12E-03 for PSO, BA, M-ISA, LWOA, RWOA, RGWO, and DAEO. From the above results, it can be inferred that the DAEO algorithm produces better effects for the IIR system identification compared to other algorithms. The convergence process of the MSE values is displayed in Fig. 16. It is obvious that all algorithms fall into local optimal solutions in early iterations. However, DAEO reaches the minimum fitness value in about 30 iterations compared to other algorithms. Figure 17 shows the variance graphs for all algorithms, where DAEO possesses smaller values, indicating that DAEO possesses good robustness. Based on the above discussion, it can be concluded that DAEO provides a good approximation for fourth-order IIR systems compared to other algorithms, which are ranked in terms of performance as DAEO > M-ISA > RGWO > RWOA > LWOA > PSO > BA.

Model 4 For the fourth test example, the transport function of the fifth-order system is expressed by Eq. (25).

$$G_p(z) = \frac{0.1084 + 0.5419Z^{-1} + 1.0837Z^{-2} + 1.0837Z^{-3} + 0.5419Z^{-4} + 0.1084Z^{-5}}{1 + 0.9853Z^{-1} + 0.9738Z^{-2} + 0.3864Z^{-3} + 0.1112Z^{-4} + 0.0113Z^{-5}} \tag{25}$$

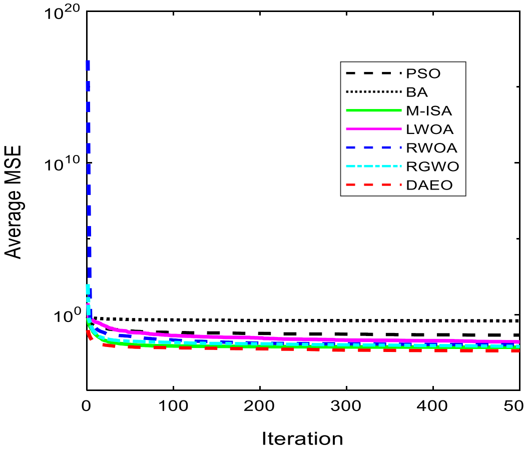
Case 1

The fifth-order filter can simulate the real fifth-order system well. Hence, the transport function of the adaptive filter is expressed by Eq. (26).

$$G_M(z) = \frac{b_0 + b_1Z^{-1} + b_2Z^{-2} + b_3Z^{-3} + b_4Z^{-4} + b_5Z^{-5}}{1 - a_1Z^{-1} - a_2Z^{-2} - a_3Z^{-3} - a_4Z^{-4} - a_5Z^{-5}} \tag{26}$$

The assessed parameter values for the same-order system are presented in Table 19, and there is no algorithm to obtain the exact parameter values. The MSE values (best, worst, mean, and variance) obtained by the various algorithms are listed in Table 20. The mean values of MSE are 2.43E-01, 2.68E+10, 4.83E-04, 4.09E-02, 2.20E-03, 2.02E-04, and 3.58E-05 for PSO, BA, M-ISA, LWOA, RWOA, RGWO, and DAEO. It can be seen that DAEO has the best approximation of the actual values for the system parameters compared to other algorithms. Figure 18

**Fig. 16** Converge diagram for model 3, case 2



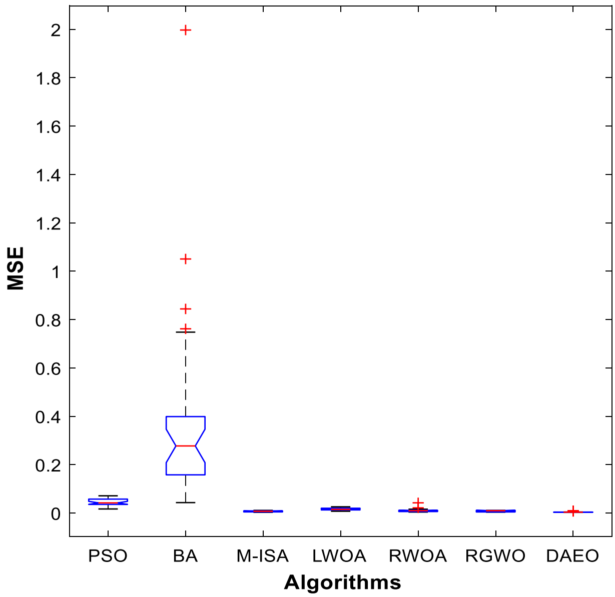


**Fig. 17** ANOVA graph for model 3, case 2

shows the convergence curve of the mean value of the MSE. As can be seen from Fig. 18, DAEO requires about 100 iterations to converge to the minimum fitness value, while BA falls prematurely into the local optimal solution. Furthermore, based on the convergence curve of DAEO, it can be inferred that the convergence rate of DAEO is higher than other algorithms. The MSE variance values in Table 20 and Fig. 19 also show that DAEO has good robustness. Based on the above values, the performance of these algorithms is ranked as DAEO > RGWO > M-ISA > RWOA > LWOA > PSO > BA.

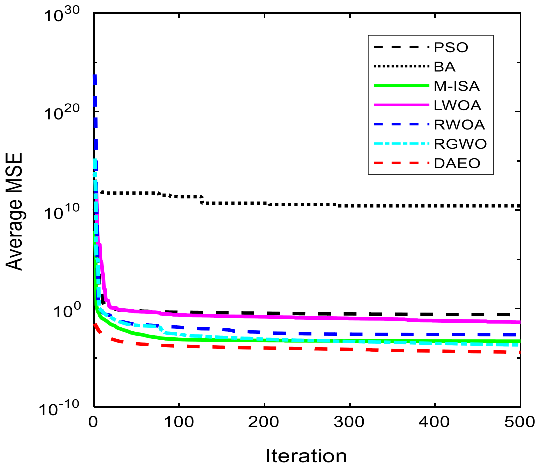
Case 2.

The fifth-order system can also be simulated by the fourth-order filter with the following transport function:

$$G'_M(z) = \frac{b'_0 + b'_1 Z^{-1} + b'_2 Z^{-2} + b'_3 Z^{-3} + b'_4 Z^{-4}}{1 - a'_1 Z^{-1} - a'_2 Z^{-2} - a'_3 Z^{-3} - a'_4 Z^{-4}} \tag{27}$$

Table 21 shows the approximate solutions of the parameters obtained by seven algorithms. Since there is no exact value for case 2, the performance of the algorithm is measured by the MSE value and the convergence rate. The MSE values are listed in Table 22. The mean values of MSE are 9.73E-02, 5.21E+00, 3.18E-04, 1.10E-02, 2.08E-03, 2.85E-04, and 5.12E-05 for PSO, BA, M-ISA, LWOA, RWOA, RGWO, and DAEO, respectively. From the above observations, it can be seen that the proposed DAEO algorithm gives the best results in terms of system identification compared to other algorithms. Figure 20 shows the convergence graphs of the different algorithms. The DAEO algorithm converges rapidly and reaches the

Fig. 18 Converge diagram for model 4, case 1



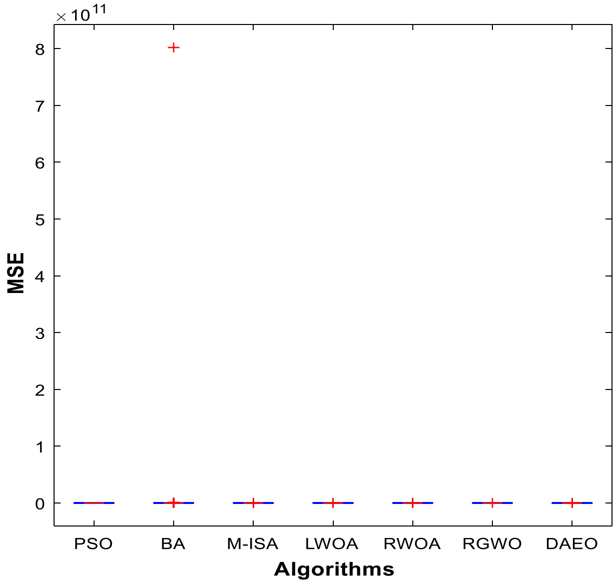
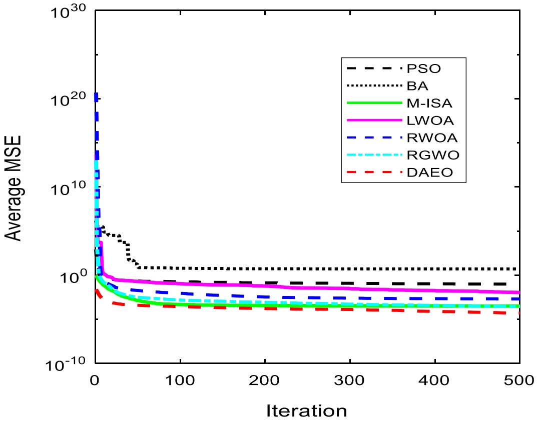


Fig. 19 ANOVA graph for model 4, case 1

minimum fitness value in about 80 iterations. From Fig. 21, the DAEO algorithm is very stable in solving the IIR model problem. In summary, in terms of its MSE value and convergence rate, the performance ranking of the adopted algorithm is DAEO > RGWO > M-ISA > RWOA > LWOA > PSO > BA.

Model 5 For the fifth test example, the transport function of a sixth-order system is expressed by Eq. (28).

**Fig. 20** Converge diagram for model 4, case 2



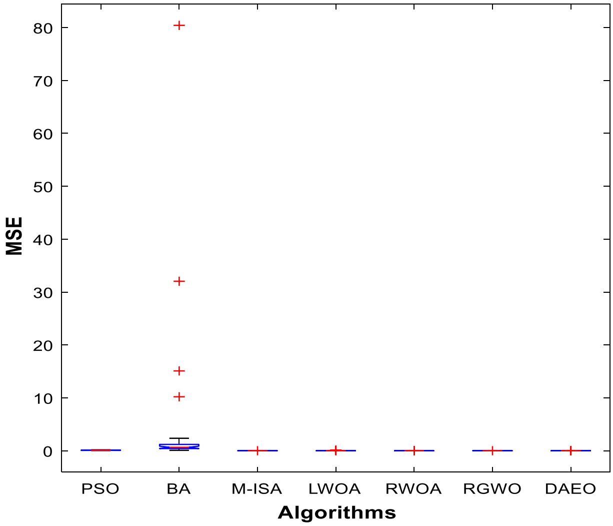


**Fig. 21** ANOVA graph for model 4, case 2

$$G_p(z) = \frac{1 - 0.4Z^{-2} - 0.65Z^{-4} + 0.26Z^{-6}}{1 - 0.77Z^{-2} - 0.8498Z^{-4} + 0.6486Z^{-6}} \tag{28}$$

Case 1.

The sixth-order filter can simulate the real sixth-order system well. Hence, the transport function of the filter is expressed by Eq. (29).

$$G_M(z) = \frac{b_0 + b_2Z^{-2} + b_4Z^{-4} + b_6Z^{-6}}{1 - a_2Z^{-2} - a_4Z^{-4} - a_6Z^{-6}} \tag{29}$$

Table 23 presents the results of the optimal parameters of the unknown system obtained by applying the metaheuristic algorithm, none of which can obtain the exact parameter values. Table 24 shows the MSE values for the sixth-order system identification problem. As can be seen from Table 24, the results obtained by DAEO are the best compared to other algorithms in terms of the best, worst, and mean values of MSE. Based on the MSE values in Table 24, it can be concluded that DAEO gives the best approximation of the parameters of the sixth-order IIR system. Figure 22 shows the convergence curves of various algorithms for the mean of the MSE. In Fig. 22, DAEO requires about 80 iterations to converge to the minimum fitness value, while BA and PSO are prone to fall into local optima at the beginning of the iterations. Based on the above discussion, the performance ranking of all algorithms in the system identification problem is DAEO > M-ISA > RGWO > RWOA > LWOA > PSO > BA (Fig. 23).

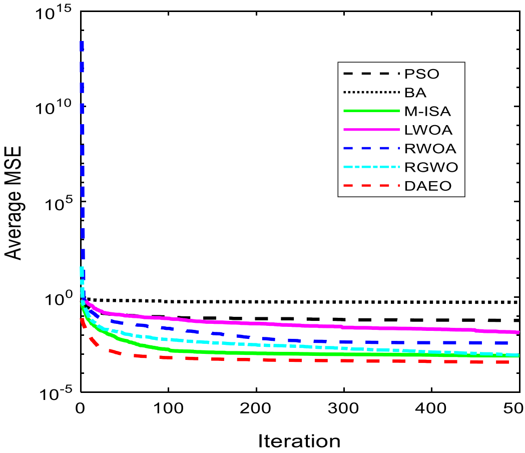
Case 2.

The sixth-order system can also be simulated by the fifth-order filter with the flowing transport function.

$$G'_M(z) = \frac{b'_0 + b'_1 Z^{-1} + b'_2 Z^{-2} + b'_3 Z^{-3} + b'_4 Z^{-4} + b'_5 Z^{-5}}{1 - a'_1 Z^{-1} - a'_2 Z^{-2} - a'_3 Z^{-3} - a'_4 Z^{-4} - a'_5 Z^{-5}} \tag{30}$$

Table 25 shows the approximate solutions of the parameters obtained by seven algorithms. Due to the use of the reduced-order system, there is no exact value. The recognition performance of the algorithm is mainly measured by the MSE value and the convergence rate. For the system identification problem, the fitness values (MSE) of all algorithms are listed in Table 26. From the MSE values in Table 26, it can be summarized that the DAEO-based system identification surpasses all other algorithms. Figure 24 shows the convergence graph of the reduced-order system, where the DAEO obtains the minimum error at the 100th iteration and has a fast convergence rate. In Fig. 25, most of the algorithms have good stability. Based on the above numerical results, the algorithms used can be ranked as DAEO > RGWO > M-ISA > RWOA > LWOA > PSO > BA.

**Fig. 22** Converge diagram for model 5, case 1



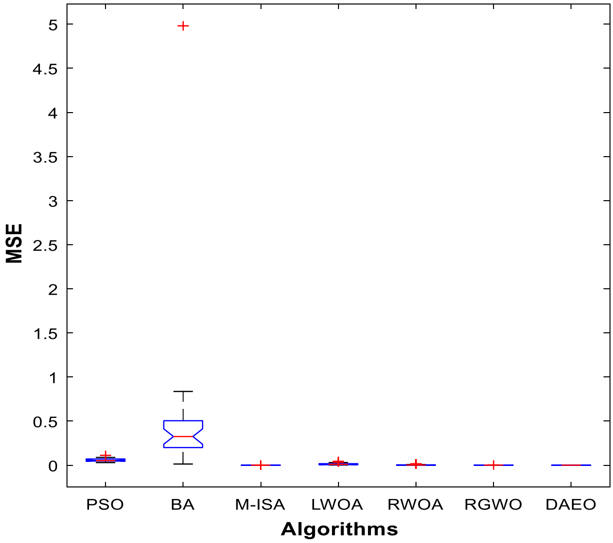
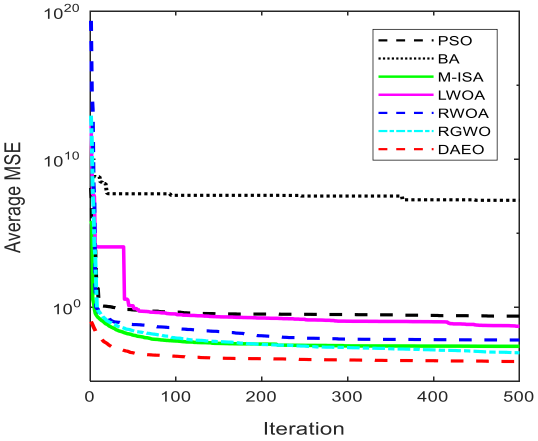


**Fig. 23** ANOVA graph for model 5, case 1

**Fig. 24** Converge diagram for model 5, case 2



## 6 Discussion analysis

DAEO is applied to function optimization and IIR system identification problems. Experimental results in Sects. 5.1 and 5.2 indicate the effectiveness and practicability of the proposed DAEO algorithm. As can be seen from Fig. 5 in experiment 1, in terms of convergence behavior, DAEO lies at the bottom on most functions and accelerates the detection of the global optimal solution. Taking the function $f_1$ as an example, DAEO can sustain good acceleration during the iterative process, while AEO has good convergence speed but is not as good as DAEO. GWO and WOA have poor convergence speed at the beginning of the iteration; although it

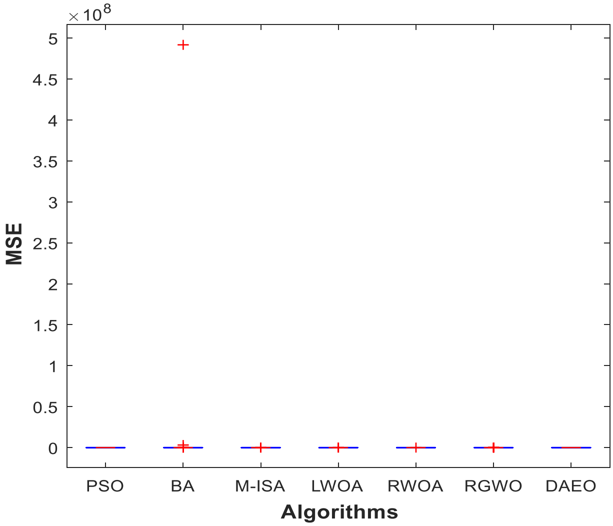


**Fig. 25** ANOVA graph for model 5, case 2

decreases later in the iterations, their convergence accuracy is also not good, while MFO, SSA, and PSOGSA fall into stagnation too early. In experiment 2, compared with other algorithms for IIR model recognition, the DAEO provides the best evaluation parameters and the best optimization accuracy in all cases. In the same-order system for models 1, 2, and 3, it is obtained from Tables 7, 11, and 15 that DAEO obtains parameter values in agreement with the actual parameter values. Although in Table 7, M-ISA also obtains parameter values in agreement with the actual values, the MSE values of M-ISA and other algorithms are not as good as the values of DAEO algorithm. In the same-order system for models 4 and 5, it can be seen from Tables 19 and 23 that none of the algorithms obtains parameter values consistent with the actual parameter values, but DAEO obtains MSE values (except for the variance values of DAEO in Table 23) better than the reported algorithms. The MSE values obtained for all algorithms in the reduced-order system also indicate that DAEO outperforms the other algorithms in terms of recognition performance, convergence, and robustness, while BA ranks the worst among the reported algorithms.

Our proposed DAEO performs better than other comparing algorithms in the two experiments due to the nonlinear weight coefficient and the dynamic opposite learning strategy. By studying the artificial ecosystem optimizer (AEO), we understand that the consumption stage represents the global search process of AEO. The producer in population is presented as the worst individual with the largest object function value, and the updated producer will guide consumers to search different regions directly or indirectly. Although the producer is updated with the participation of decomposers (the best individual), to some extent, the exploration capability of the algorithm is significantly weakened, and its search space keeps away from promising areas. Therefore, before the individuals update their position, we added the DOL strategy to form the new population. Its dynamic asymmetric search space helps the algorithm enhance the probability of obtaining good solutions and makes the global search of AEO more precise. Additionally, the DAEO always got the best population during the iteration; therefore, the searching individual could renew its position according to the optimal solution space. DAEO will have a higher probability of seeing the promising areas. Moreover, for any optimization algorithm, the poise between exploration and exploitation is essential. In the AEO, $\alpha$ controls the balance between exploration and exploitation of the algorithm. In the process of iteration, parameter $\alpha$ gets a small value in most cases, which is not conducive to the global search of the algorithm. Hence, a nonlinear weight parameter is used to make the algorithm more flexible. It enhances the algorithm's exploration capability at the iteration process, which is to achieve a good balance. Finally, DAEO maintains the simple structure of AEO, and its control parameters are less than other algorithms. Therefore, the above advantages make DAEO more suitable for practical applications and effectively deal with the function optimization and the IIR model system identification problem.

## 7 Conclusions and future directions

The paper has posted a modified version of the AEO algorithm adopting the DOL strategy and an adaptive nonlinear weight coefficient, and the novel scheme is called the DAEO. The DOL strategy forms a dynamic asymmetric search space by computing the reverse solution to the present solution to improve the overall performance of the algorithm. Since the search process in the AEO displayed an inappropriate balance relationship between exploration and exploitation. Hence, in the DAEO, an adaptive nonlinear weight coefficient was presented to build a better balance between the two stages during the search. To verify the effectiveness of the DAEO algorithm, the experiment is split into two parts. The first part is that DAEO outperforms the original AEO and other algorithms in terms of optimal global ability and convergence performances in the challenging test set. Also, DAEO is examined with the statistical analysis, i.e., the Wilcoxon nonparametric rank-sum test, and compared with the other six comparison algorithms to prove the effectiveness of the algorithm. The second part is to choose five typical IIR filter models from the related literature, which have the same order and reduced order. Simulation results manifest that the proposed algorithm has good optimization precision, convergence rate, and stability than other algorithms dealing with filter identification problems.

Although the recognition accuracy obtained by DAEO is better than that of other algorithms in the system recognition problem, there is still room for continued improvement, and the decomposition mechanism of DAEO tends to fall into local optimum in the late iteration, which reduces the mining ability of the algorithm. Therefore, effective methods and strategies will be proposed in the future to improve the performance of DAEO and further extend the application field of the algorithm.

**Acknowledgements** This work is supported by the National Key R&D Program of China, Key Technology Research and Platform Development for Cloud Manufacturing Based on Open Architecture under Grant No.: 2018YFB1702700.

## Declarations

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

## References

1. Zarai K, Cherif A (2021) Adaptive filter based on Monte Carlo method to improve the non-linear target tracking in the radar system. Aerosp Syst 4:67–74. https://doi.org/10.1007/s42401-020-00080-9
2. Li W, Xiang Z, Ren P, Li Q (2021) Adaptive filter bank multi-carrier waveform design for joint communication-radar system. Digit Sign Proc 110:102950. https://doi.org/10.1016/j.dsp.2020.102950
3. Milosavljevic MS, Corron NJ, Blakely JN (2020) Optimal communications with infinite impulse response matched filters. Chaos, Solitons Fractals 138:109822. https://doi.org/10.1016/j.chaos.2020.109822
4. Le Cong D, Zhang J, Pang Y (2021) A novel pipelined neural FIR architecture for nonlinear adaptive filter. Neurocomputing 440:220–229. https://doi.org/10.1016/j.neucom.2020.11.036
5. Samalla DK, SubbaRao SPV, Mallikarjuna Rao G, Jagadeesh BN (2021) Design and development of adaptive filter for real time signal processing. IOP Conf Ser: Mater Sci Eng 1074:012036. https://doi.org/10.1088/1757-899X/1074/1/012036
6. Zhang Q, Zhao L, Zhao L (2021) A two-step robust adaptive filtering algorithm for GNSS kinematic precise point positioning. Chin J Aeronaut. https://doi.org/10.1016/j.cja.2020.10.033
7. Chandra M (2021) Design and analysis of improved high-speed adaptive filter architectures for ECG signal denoising. Biomed Sign Proc Control. https://doi.org/10.1016/j.bspc.2020
8. Eltrass AS, Ghanem NH (2021) A new automated multi-stage system of non-local means and multi-kernel adaptive filtering techniques for EEG noise and artifacts suppression. J Neural Eng 18:036023. https://doi.org/10.1088/1741-2552/abe397
9. Nagasirisha B, Prasad VVKDV (2021) EMG SIGNAL DENOISING USING ADAPTIVE FILTERS THROUGH HYBRID OPTIMIZATION ALGORITHMS. Biomed Eng Appl Basis Commun 33:2150009. https://doi.org/10.4015/S1016237221500095
10. Czapiewska A, Luksza A, Studanski R, Zak A (2020) Application of Diversity combining with RLS adaptive filtering in data transmission in a hydroacoustic channel. Sensors 20:7255. https://doi.org/10.3390/s20247255
11. Mohammadi A, Zahiri SH, Razavi SM (2019) Infinite impulse response systems modeling by artificial intelligent optimization methods. Evol Syst. https://doi.org/10.1007/s12530-018-9218-z
12. Kumar M, Aggarwal A, Rawat TK (2016) Bat algorithm: application to adaptive infinite impulse response system identification. Arab J Sci Eng 41:3587–3604. https://doi.org/10.1007/s13369-016-2222-3
13. Luo Q, Ling Y, Zhou Y (2020) Modified Whale Optimization Algorithm for Infinitive Impulse Response System Identification. Arab J Sci Eng 45:2163–2176. https://doi.org/10.1007/s13369-019-04093-1
14. Yao L, Sethares WA (1994) Nonlinear parameter estimation via the genetic algorithm. IEEE Trans Signal Process 42:927–935. https://doi.org/10.1109/78.285655
15. Krusienski DJ, Jenkins WK (2004) Particle swarm optimization for adaptive IIR filter structures. In: Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753). IEEE, Portland, OR, USA, pp 965–970
16. Panda G, Pradhan PM, Majhi B (2011) IIR system identification using cat swarm optimization. Expert Syst Appl 38:12671–12683. https://doi.org/10.1016/j.eswa.2011.04.054
17. Mondal S, Ghoshal SP, Kar R, Mandal D (2012) Differential evolution with wavelet mutation in digital finite impulse response filter design. J Optim Theory Appl 155:315–324. https://doi.org/10.1007/s10957-012-0028-3
18. Saha SK, Kar R, Mandal D et al (2013) A new design method using opposition-based BAT algorithm for IIR system identification problem. IJBIC 5:99. https://doi.org/10.1504/IJBIC.2013.053508
19. Upadhyay P, Kar R, Mandal D et al (2014) A novel design method for optimal IIR system identification using opposition based harmony search algorithm. J Franklin Inst 351:2454–2488. https://doi.org/10.1016/j.jfranklin.2014.01.001

20. Jiang S, Wang Y, Ji Z (2015) A new design method for adaptive IIR system identification using hybrid particle swarm optimization and gravitational search algorithm. Nonlinear Dyn 79:2553–2576. https://doi.org/10.1007/s11071-014-1832-0
21. Lagos-Eulogio P, Seck-Tuoh-Mora JC, Hernandez-Romero N, Medina-Marin J (2017) A new design method for adaptive IIR system identification using hybrid CPSO and DE. Nonlinear Dyn 88:2371–2389. https://doi.org/10.1007/s11071-017-3383-7
22. Kumar M, Rawat TK, Aggarwal A (2017) Adaptive infinite impulse response system identification using modified-interior search algorithm with Lèvy flight. ISA Trans 67:266–279. https://doi.org/10.1016/j.isatra.2016.10.018
23. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Computat 1:67–82. https://doi.org/10.1109/4235.585893
24. Zhao W, Wang L, Zhang Z (2020) Artificial ecosystem-based optimization: a novel nature-inspired meta-heuristic algorithm. Neural Comput & Applic 32:9383–9425. https://doi.org/10.1007/s00521-019-04452-x
25. Yang Y, Yang B, Niu M (2018) Adaptive infinite impulse response system identification using opposition based hybrid coral reefs optimization algorithm. Appl Intell 48:1689–1706. https://doi.org/10.1007/s10489-017-1034-9
26. Durmuş B (2022) Infinite impulse response system identification using average differential evolution algorithm with local search. Neural Comput Appl 34:375–390. https://doi.org/10.1007/s00521-021-06399-4
27. Sahlol AT, Abd Elaziz M, Tariq Jamal A et al (2020) A novel method for detection of tuberculosis in chest radiographs using artificial ecosystem-based optimisation of deep neural network features. Symmetry 12:1146. https://doi.org/10.3390/sym12071146
28. Mouassa S, Jurado F, Bouktir T, Raja MAZ (2020) Novel design of artificial ecosystem optimizer for large-scale optimal reactive power dispatch problem with application to Algerian electricity grid. Neural Comput & Applic. https://doi.org/10.1007/s00521-020-05496-0
29. Essa FA (2020) Prediction of power consumption and water productivity of seawater greenhouse system using random vector functional link network integrated with artificial ecosystem-based optimization. Process Safety Environ Prot. https://doi.org/10.1016/j.psep.2020.07.044
30. Tubishat M, Idris N, Shuib L et al (2020) Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection. Expert Syst Appl 145:113122. https://doi.org/10.1016/j.eswa.2019.113122
31. Ewees AA, Abd Elaziz M, Houssein EH (2018) Improved grasshopper optimization algorithm using opposition-based learning. Expert Syst Appl 112:156–172. https://doi.org/10.1016/j.eswa.2018.06.023
32. Gupta S (2020) Opposition-based learning Harris hawks optimization with advanced transition rules: principles and analysis. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2020.113510
33. Xu Y, Yang Z, Li X et al (2020) Dynamic opposite learning enhanced teaching–learning-based optimization. Knowl-Based Syst 188:104966. https://doi.org/10.1016/j.knosys.2019.104966
34. Han X, Yue L, Dong Y et al (2020) Efficient hybrid algorithm based on moth search and fireworks algorithm for solving numerical and constrained engineering optimization problems. J Supercomput 76:9404–9429. https://doi.org/10.1007/s11227-020-03212-2
35. Goldanloo MJ, Gharehchopogh FS (2021) A hybrid OBL-based firefly algorithm with symbiotic organisms search algorithm for solving continuous optimization problems. J Supercomput. https://doi.org/10.1007/s11227-021-04015-9
36. Tang C, Zhou Y, Tang Z, Luo Q (2021) Teaching-learning-based pathfinder algorithm for function and engineering optimization problems. Appl Intell. https://doi.org/10.1007/s10489-020-02071-x
37. Wang Z, Luo Q, Zhou Y (2020) Hybrid metaheuristic algorithm using butterfly and flower pollination base on mutualism mechanism for global optimization problems. Eng Comput. https://doi.org/10.1007/s00366-020-01025-8
38. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl-Based Syst 89:228–249. https://doi.org/10.1016/j.knosys.2015.07.006
39. Mirjalili S, Gandomi AH, Mirjalili SZ et al (2017) Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. Adv Eng Softw 114:163–191. https://doi.org/10.1016/j.advengsoft.2017.07.002
40. Mirjalili S, Hashim SZM (2010) A new hybrid PSOGSA algorithm for function optimization. In: 2010 International conference on computer and information application. IEEE, Tianjin, China, pp 374–377

41. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
42. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
43. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1:3–18. https://doi.org/10.1016/j.swevo.2011.02.002
44. Zhang S, Zhou Y (2018) Grey Wolf Optimizer with Ranking-Based Mutation Operator for IIR Model Identification. Chinese J Electron 27:1071
45. Ling Y, Zhou Y, Luo Q (2017) Lévy flight trajectory-based whale optimization algorithm for global optimization. IEEE Access 5:6168–6186. https://doi.org/10.1109/ACCESS.2017.2695498

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Authors and Affiliations

**Yanbiao Niu**[1] · **Xuefeng Yan**[1] · **Yongzhen Wang**[1] · **Yanzhao Niu**[2]

✉ Xuefeng Yan
fzjm_313@nuaa.edu.cn

[1] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

[2] School of Instrumentation and Optoelectronic Engineering, Beihang University, Beijing 100191, China