



# Artificial ecosystem-based optimization: a novel nature-inspired meta-heuristic algorithm

Weiguo Zhao<sup>1,2</sup> · Liying Wang<sup>1</sup> · Zhenxing Zhang<sup>2</sup>

Received: 22 March 2019 / Accepted: 19 August 2019 / Published online: 11 September 2019  
© Springer-Verlag London Ltd., part of Springer Nature 2019

## Abstract

A novel nature-inspired meta-heuristic optimization algorithm, named artificial ecosystem-based optimization (AEO), is presented in this paper. AEO is a population-based optimizer motivated from the flow of energy in an ecosystem on the earth, and this algorithm mimics three unique behaviors of living organisms, including production, consumption, and decomposition. AEO is tested on thirty-one mathematical benchmark functions and eight real-world engineering design problems. The overall comparisons suggest that the optimization performance of AEO outperforms that of other state-of-the-art counterparts. Especially for real-world engineering problems, AEO is more competitive than other reported methods in terms of both convergence rate and computational efforts. The applications of AEO to the field of identification of hydrogeological parameters are also considered in this study to further evaluate its effectiveness in practice, demonstrating its potential in tackling challenging problems with difficulty and unknown search space. The codes are available at <https://www.mathworks.com/matlabcentral/fileexchange/72685-artificial-ecosystem-based-optimization-aeo>.

**Keywords** Artificial ecosystem-based optimization · Global optimization · Constrained problems · Optimization algorithm · Engineering design · Hydrogeological parameter

## 1 Introduction

Optimization for real-world problems is becoming an increasingly challenging field that has attracted considerable attraction from researchers for several decades. A big number of scholars explore different methods to tackle all sorts of real-world optimization problems, and most often used methods are numerical methods that usually adopt simple and ideal mathematical models. However, these methods generally need some gradient information to seek

better solutions revolving around a special point in a local region. In addition, they are very sensitive to initial points; especially when the considered problems have multiple or sharp peaks, the improper choice of initial points tends to make the search for the global optimum become difficult and unstable [1]. Recently, numerous complex optimization problems are increasingly emerging in different domains, where these problems often concern multiple decision variables, complex nonlinear constraints, and objective functions [2, 3]. Therefore, these complex problems fail to be well solved by traditional numerical methods with an acceptable time and solution precision. However, nature offers abundant sources for us including thoughts, inspiration and concepts to develop artificial intelligent methods that are used to tackle these complex problems.

Over the past years, various nature-inspired optimization approaches have been proposed. These algorithms imitate behaviors of living organisms or natural phenomena for various problems. Genetic algorithm (GA), conceived by Holland [4], evolves populations by imitating the survival of the fittest in nature; particle swarm optimization (PSO), developed by Kennedy and Eberhart [5], models the

✉ Liying Wang  
wangliying@hebeu.edu.cn

Weiguo Zhao  
zhaoweiguo@hebeu.edu.cn

Zhenxing Zhang  
zhang538@illinois.edu

<sup>1</sup> School of Water Conservancy and Hydropower, Hebei University of Engineering, Handan 056021, Hebei, China

<sup>2</sup> Illinois State Water Survey, Prairie Research Institute, University of Illinois at Urbana-Champaign, Champaign, IL 61820, USA

movement of a population of individuals (e.g., birds); ant colony optimization (ACO), introduced by Dorigo and Maniezzo [6], is motivated from the foraging process of an ant colony; simulated annealing (SA), proposed by Kirkpatrick et al. [7], is originated from the annealing process used in physical material; and differential evolution (DE), presented by Storn [8], simulates the biological processes of genetic inheritance and survival of the fittest. These approaches are very popular in scientific fields of intelligent computing since they provide a better optimization performance, especially for those non-differentiable, non-continuous, multimodal, and multidimensional problems. With their significant attention and applications, a large number of other optimization algorithms have been developed and applied to a range of fields with success [9–15]. These algorithms are roughly divided into three classes [16]: evolution-based (EB) [17], physics-based (PB) [18], and swarm-based (SB) algorithms [19].

EBs mimic natural evolution such as selection, crossover, mutation, chemotaxis, and migration [20, 21]. GA and DE can be viewed as two classic representatives in EBs. Both evolve populations by imitating the survival of the fittest in nature and are usually able to obtain high-quality solutions. Some other well-known EBs are evolutionary strategies (ES) [22], genetic programming (GP) [23], selfish gene algorithm (SGA) [24], shuffled frog leaping algorithm (SFLA) [25], biogeography-based optimization (BBO) [26], artificial algae algorithm (AAA) [27], backtracking search optimization algorithm (BSA) [28], and fruit fly optimization algorithm (FOA) [29].

In contrast to EBs, most SBs do not adopt genetic operators. They always stimulate collective behaviors of intelligent species to offer the better solutions to investigated problems. Two of the most classic SBs are PSO and ACO. For PSO, the solutions of a population are updated by the cognitive and social information. For ACO, each ant needs to find the shortest route between its nest and food, which can be achieved by pheromone trails deposited in routes. Some emerging SBs include artificial bee colony (ABC) [30, 31], hunting search (HS) algorithm [32], tree-seed algorithm (TSA) [33], whale optimization algorithm (WOA) [34], grasshopper optimization algorithm (GOA) [35], cuckoo search (CS) [36], crow search algorithm (CSA) [37], dolphin echolocation (DE) algorithm [38], firefly algorithm (FA) [39], virus colony search (VCS) [40], supply–demand-based optimization [41], and so on [42–45].

Different from EBs and SBs, PBs are motivated from physical laws in nature, and SA is famous one. SA always tends to probabilistically decide moving the system to either a randomly chosen neighboring state or the current state, and in this way, this algorithm can converge to the global optimum. There are some other promising PBs, including gravitational search algorithm (GSA) [46], central force

optimization (CFO) [47], wind driven optimization (WDO) [48], artificial physics optimization (APO) [49], electromagnetic field optimization (EFO) [50], atom search optimization (ASO) [51, 52], and so on [53–71].

All meta-heuristic optimization algorithms share the same following characteristics. (1) They are based on some fundamental theories and mathematical models. (2) They are simple and are easy to implement. (3) It is easy to develop their variants based on the existing meta-heuristics. (4) They can be viewed as black boxes by which a set of outputs can be easily obtained when given a set of inputs. (5) They are highly versatile and flexible in dealing with various optimization problems. However, one might be asking why new optimization methods are still raised despite so many existing algorithms. The answer can be found in the no-free-lunch theorem of optimization [72], which states that no optimization method can perform the best for every optimization problem. So developing new and effective swarm-inspired optimizers to tackle specific real-world problems motivates this study. In addition, most optimization algorithms have several control parameters. For a given algorithm, it is time-consuming and difficult to find a set of different parameters that can well fit different problems, largely limiting the availability of the algorithm. Therefore, developing a new optimizer with less parameters is another motivation of this study.

This paper proposes a new meta-heuristic algorithm, named artificial ecosystem-based optimization (AEO), which is motivated from the flow of energy in an ecosystem on the earth. AEO mimics production, consumption, and decomposition behaviors of living organisms. The AEO is evaluated on 31 mathematical problems and 8 real-world engineering problems. The comparative test reveals that the proposed approach is superior to those popular meta-heuristics. Eventually, the applications of AEO to the field of identification of hydrogeological parameters demonstrate its potential in other engineering fields.

The paper is structured as follows. Section 2 describes AEO algorithm and the concepts behind it in detail. Thirty-one mathematical optimization problems and 8 real-world engineering problems are utilized to test the validity of the proposed optimizer from different perspectives in Sects. 3 and 4, respectively. Additionally, Sect. 5 investigates the applications of AEO to the field of identification of hydrogeological parameters. Section 6 concludes the study and suggests several directions for future work.

## 2 Artificial ecosystem-based optimization (AEO)

In this section, the inspiration of AEO is firstly introduced. Then, its mathematical model is provided in detail.

## 2.1 Inspiration

‘Ecosystem,’ an intuitively appealing concept to most ecologists, was firstly introduced by A.G. Tansley in 1955. Owing to this appeal, the ecosystem concept has become increasingly important and popular [73]. An ecosystem is the complex of living organisms, their physical environment, and all their interrelationships in a particular unit of space [74]. An ecosystem can be classified into abiotic and biotic constituents. Abiotic constituents include sunlight, water, air, and other non-living elements. Biotic constituents include all living organisms. The flow of energy and the cycling of nutrients are chief driving forces maintaining the normal ecological order in an ecosystem, where all living organisms play a significant role. Living organisms can be categorized into producer, consumer, and decomposer.

Producers do not need to obtain energy from other organisms, and most producers are any kind of green plant that gets their food energy directly through the process of photosynthesis. During photosynthesis, carbon dioxide and water react together in the presence of sunlight to produce a sugar called glucose and oxygen. Plants use this sugar to make many things, such as leaves, fruits, wood, and roots. Producers generally provide essential food for herbivore consumers and omnivore consumers. Consumers are animals that cannot make their food; thus, they must feed on producers or other consumers to obtain energy and nutrients. Consumers can be categorized into carnivores, herbivores, and omnivores. Animals that eat only producers (plants) are called herbivores. Animals that eat both producers and other animals are called omnivores. Animals that eat only other animals are called carnivores. Decomposers are an organism that feeds on both dead plants (producers) and animals (consumers) or on the waste from living organisms. Decomposers include most bacteria and fungi. When an organism dies, decomposers break down the remains and convert them into simple molecules, such as carbon dioxide, water, and minerals. Then, these energy forms may be absorbed by producers to again make sugar and oxygen through photosynthesis, starting the cycle all over again.

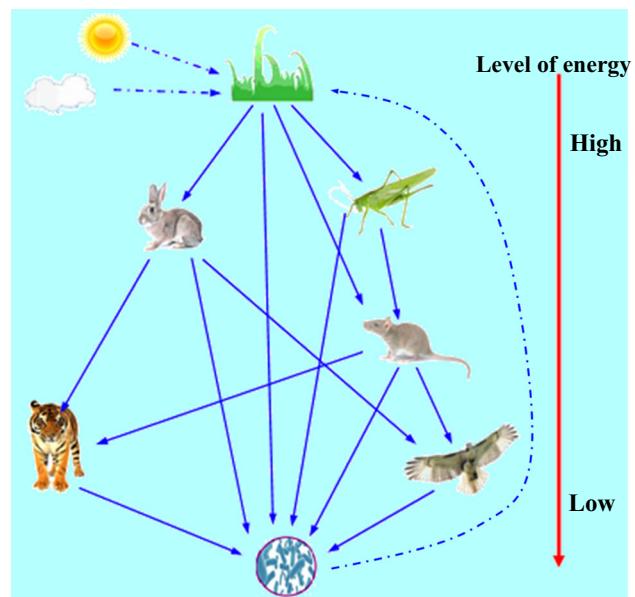
Producers, consumers, and decomposers interacting with one another compose a food chain. A food chain describes who feeds whom in an ecosystem, where every life form needs to get food energy to achieve the movement of nutrients. A food chain is a possible connected path that energy and nutrients can follow through an ecosystem, and it shows the different levels of eating within an ecosystem. Many overlapped food chains form a food web that depicts how food chains are interconnected. A food web is very complex because most consumers feed on producers or/and

different kinds of consumers and meanwhile probably be fed on other different kinds of consumers. Producers are generally at the beginning of feed webs and most food chains. Consumers are the most complex of three types of organisms. Figure 1 depicts the flow of energy in an ecosystem. The blue arrows represent the energy transfer pathway. The red arrow represents the different levels of energy decreasing from producers to decomposers. The energy always flows from the living organisms with high energy toward the living organisms with low energy. Typically, the ecosystem develops this mechanism of energy transfer as a strategy to maintain the stability of species, and it is able to keep the ecological balance over a long term. Therefore, there are reasons to believe that this mechanism of energy transfer can shape and sustain a sound and stable ecosystem.

## 2.2 Artificial ecosystem-based optimization

According to the previous discussion, our artificial ecosystem-based optimization algorithm employs three operators, including production, consumption, and decomposition. The first operator is mainly to enhance the balance between exploration and exploitation. The second operator is used to improve the exploration of the algorithm. For the third operator, it is proposed to promote the exploitation of the algorithm. AEO generally follows the following several rules to search for a solution.

1. The ecosystem as a population includes three kinds of organisms: producer, consumer, and decomposer.



**Fig. 1** Flow of energy in an ecosystem

2. There is only one producer as an individual in a population.
3. There is only one decomposer as an individual in a population.
4. The other individuals of a population are consumers, each of which is chosen as a carnivore, a herbivore, or an omnivore with the same probability.
5. The energy level of each individual in a population is evaluated by its function fitness value. The population is sorted in the descending order of function fitness value, so the higher function fitness value indicates the higher energy level for a minimization problem.

Figure 2 shows an ecosystem in AEO. In this ecosystem, all individuals are sorted in the descending order of the function fitness value, and the black arrows represent the flow direction of energy. The worst individual  $x_1$  (the highest function fitness value) and the best individual  $x_n$  (the lowest function fitness value) are a producer and a decomposer, respectively. The others are consumers, in which we suppose  $x_2$  and  $x_5$  are herbivores,  $x_3$  and  $x_7$  are omnivores, and  $x_4$  and  $x_6$  are carnivores.

### 2.2.1 Production

In an ecosystem, the producer may generate food energy with carbon dioxide, water, and sunlight, as well as the nutrition provided by the decomposer. Likewise, in AEO, the producer (the worst individual) in a population needs to be updated by the low and upper limits of search space and the decomposer (the best individual), and this updated individual will guide other individuals including herbivore and omnivore in the population to search for different regions. The production operator allows AEO to randomly produce a new individual replacing the previous one between the best individual ( $x_n$ ) and an individual randomly generated in the search space ( $x_{\text{rand}}$ ). The

mathematical model of the production operator is represented as follows:

$$x_1(t+1) = (1 - a)x_n(t) + ax_{\text{rand}}(t) \quad (1)$$

$$a = (1 - t/T)r_1 \quad (2)$$

$$x_{\text{rand}} = \mathbf{r}(U - L) + L \quad (3)$$

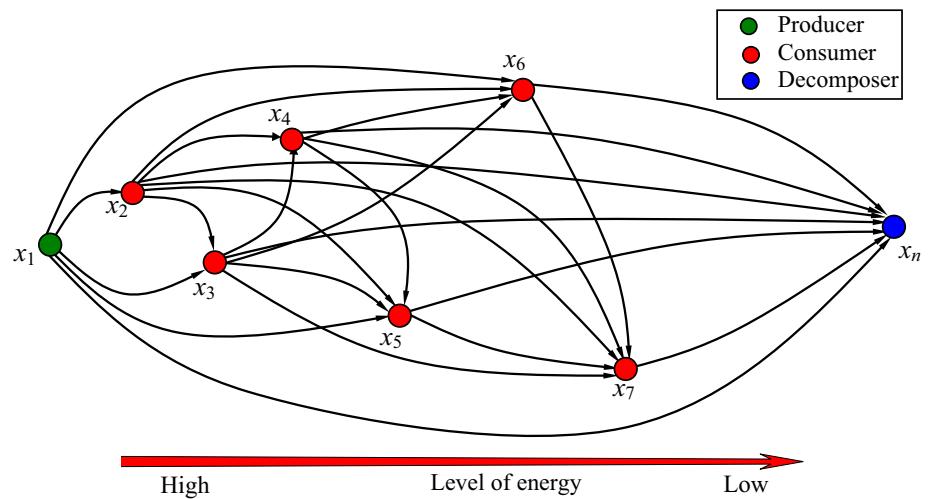
where  $n$  is the size of a population,  $T$  is the maximal number of iterations,  $L$  and  $U$  are the lower and upper limits, respectively,  $r_1$  is a random number within the range of  $[0, 1]$ ,  $\mathbf{r}$  is a random vector within the range of  $[0, 1]$ ,  $a$  is a linear weight coefficient, and  $x_{\text{rand}}$  is the position of an individual randomly produced in the search space. In Eq. (1), the weight coefficient  $a$  is used to linearly drift the individual from a position randomly produced toward the position of the best individual as the iterations increase. As Eq. (1) shows, in the early iterations,  $x_1(t+1)$  may guide the other individuals to perform an exploration in the search space extensively; in the later iterations,  $x_1(t+1)$  may guide the other individuals to perform the exploitation in a region around  $x_n$  intensively.

This production operator in 2-D and 3-D spaces using Eq. (1) can be simulated, which is illustrated in Fig. 3, and this production behavior is performed 20 iterations. As shown in the figure, this operator offered by Eq. (1) gradually moves an individual randomly generated toward the best individual. After 20 iterations, the individual reaches the position of the best individual.

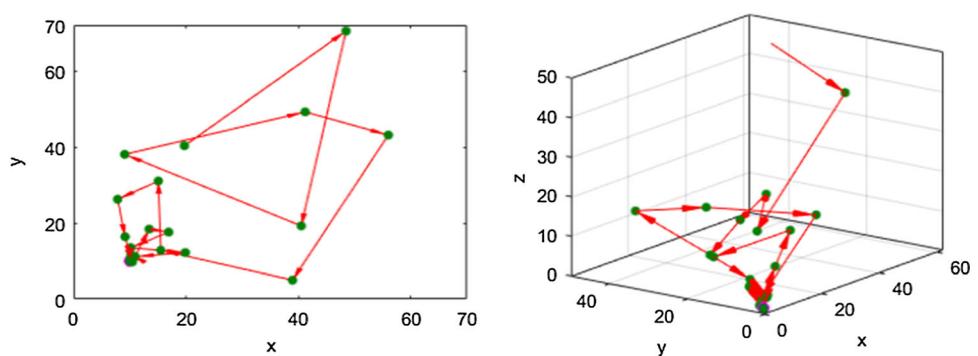
### 2.2.2 Consumption

After the producer accomplishes production operator, all the consumers may perform consumption operator. To obtain food energy, each consumer may eat either a randomly chosen consumer with the lower level of energy or a producer, or both. Levy flight, as a mathematic operator,

**Fig. 2** An ecosystem in AEO



**Fig. 3** Behaviors of production in 2-D and 3-D spaces during 20 iterations



mimics food searching of many animals like cuckoo, bumblebees, deer, and lion. Levy flight is a random walk which can efficiently explore the search space as the length of some steps is much longer in the long run, showing that it is promising in finding the global optimum. Therefore, Levy flight was often added to nature-inspired algorithms to enhance their optimization performance [75–77]. However, there are two drawbacks to this behavior, including complexity and several parameters to be tuned. At a point, a simple, parameter-free random walk with the feature of Levy flight, called consumption factor, is proposed. It is defined as follows:

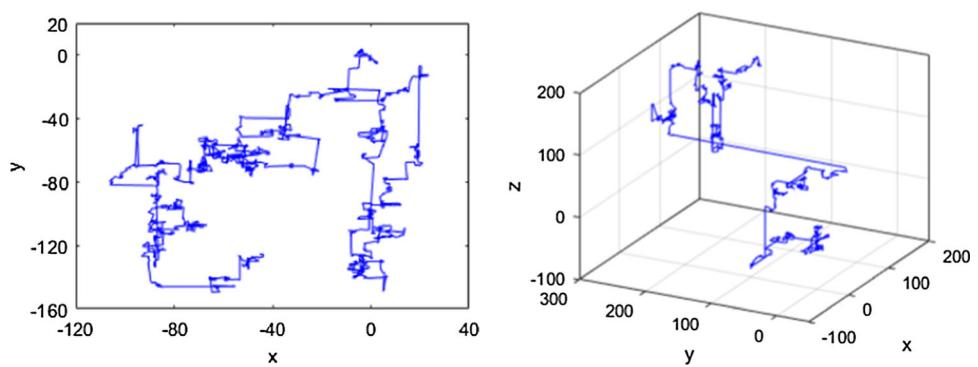
$$C = \frac{1}{2} \frac{v_1}{|v_2|} \quad (4)$$

$$v_1 \sim N(0, 1), \quad v_2 \sim N(0, 1) \quad (5)$$

where  $N(0, 1)$  is a normal distribution with the mean = 0 and the standard deviation = 1.

The sample trajectories of the consumption factor in 2-D and 3-D spaces during 2000 iterations are shown in Fig. 4. It can be found that these random walks generally get crowded around a central point and permit occasionally jump a long step far from the previous point. It allows AEO to have the opportunity to avoid local extrema and to explore the entire search space. Therefore, this consumption factor may assist each consumer to hunt for food, while different types of consumers adopt different consumption strategies.

**Fig. 4** Sample trajectories of consumption factor in 2-D and 3-D spaces during 2000 iterations



**Herbivore:** If a consumer is randomly chosen as an herbivore, it only eats the producer. As illustrated in Fig. 2, both consumers  $x_2$  and  $x_5$  are herbivores which only consume the producer  $x_1$ . To mathematically model this consumption behavior of herbivores, the following equation is presented as

$$x_i(t+1) = x_i(t) + C \cdot (x_i(t) - x_1(t)), \quad i \in [2, \dots, n] \quad (6)$$

**Carnivore:** If a consumer is randomly chosen as a carnivore, it can only eat randomly a consumer with the higher energy level. In Fig. 2, the consumer  $x_6$  is a carnivore, so it needs to randomly choose a consumer for food from the individuals  $x_2$  to  $x_5$  that all have higher energy levels than the consumer  $x_5$ . The equation modeling the consuming behavior of a carnivore is as follows:

$$\begin{cases} x_i(t+1) = x_i(t) + C \cdot (x_i(t) - x_j(t)), & i \in [3, \dots, n] \\ j = \text{randi } ([2 \dots i-1]) \end{cases} \quad (7)$$

**Omnivore:** If a consumer is randomly chosen as an omnivore, it can eat both a consumer with the higher energy level randomly and a producer. In Fig. 2, the consumer  $x_7$  is an omnivore, so it needs to eat both the producer  $x_1$  and a consumer randomly chosen from the individual  $x_2$  to  $x_6$  that have the higher energy levels than  $x_7$ . The mathematical equation modeling the consuming behavior of an omnivore can be expressed as

$$\begin{cases} x_i(t+1) = x_i(t) + C \cdot (r_2 \cdot (x_i(t) - x_1(t)) \\ \quad + (1 - r_2)(x_i(t) - x_j(t))), & i = 3, \dots, n \\ j = \text{randi}([2 \ i-1]) \end{cases} \quad (8)$$

where  $r_2$  is a random number within the range of  $[0, 1]$ .

For this consumption operator, AEO updates the position of a search individual with respect to either the worst individual or a randomly chosen individual in a population, or both. This behavior tends to emphasize the exploration and allows AEO to perform a global search.

### 2.2.3 Decomposition

Decomposition is a very vital process in terms of the functioning of an ecosystem, and it provides essential nutrients for the growth of the producer. During the decomposition, when each individual in the population dies, the decomposer will decay or break down chemically its remains. In order to mathematically model this behavior, the decomposition factor  $D$  and the weight coefficients  $e$  and  $h$  are designed. Hence, the position of the  $i$ th individual  $x_i$  in a population can be updated by the position of the decomposer  $x_n$  via these parameters  $D$ ,  $e$ , and  $h$ . It allows the next position of each individual to spread around the best individual (the decomposer), showing the exploitation to some extent. The equation expressing this decomposition behavior is as follows:

$$x_i(t+1) = x_n(t) + D \cdot (e \cdot x_n(t) - h \cdot x_i(t)), \quad i = 1, \dots, n \quad (9)$$

$$D = 3u, \quad u \sim N(0, 1) \quad (10)$$

$$e = r_3 \cdot \text{randi}([1 \ 2]) - 1 \quad (11)$$

$$h = 2 \cdot r_3 - 1 \quad (12)$$

Figure 5 shows that an individual samples 200 times in 2-D and 3-D spaces, respectively, according to Eq. (9). As shown in Fig. 5, most sampled points randomly distribute in the distance between the current individual  $x_i$  and the

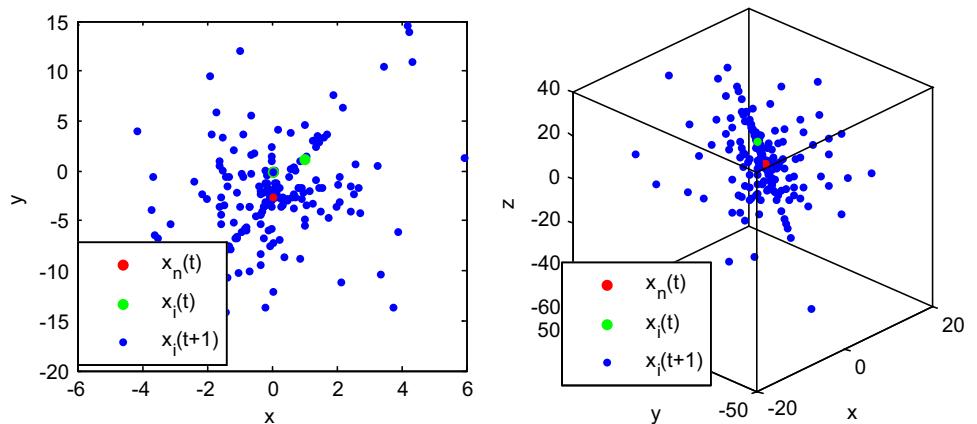
individual  $x_n$ , and the distribution of the sampled points becomes sparse as the distance between these two individuals reduces. Meanwhile, it can be seen that from this figure, a few of sampled points tend to randomly sample some regions far away from the individual  $x_n$ . This observation is able to support exploitation and avoidance of the local extrema.

AEO starts the optimization by generating a population randomly. At each iteration, the first search individual updates its position based on Eq. (1), and for the other individuals, there is a same probability to choose among Eqs. (6), (7) or Eq. (8) to update their positions. Accept it if an individual is given a better function value. Then, each individual updates its position based on Eq. (9). If an individual goes outside of the lower or upper boundaries during the updating process, it will be randomly generated in the search space. All the updates are interactively performed until the AEO algorithm is satisfied with a termination criterion. Finally, the solution of the best individual found so far is returned. The pseudocode of AEO algorithm is described in Fig. 6.

With the above formulation of AEO algorithm and the observation of its optimization performance, the following remarks are made.

1. AEO is motivated from three energy transfer mechanism in an ecosystem, including production, consumption, and decomposition, which are effective in improving the optimizing performance of AEO from different aspects.
2. The production assists AEO to produce a candidate solution drifting from a randomly generated position to the best position with the increase in iterations, and this solution will guide other individuals to perform consumption operator during the consuming process. This behavior contributes greatly to the balance between the explorative and exploitative search.
3. The consumption allows AEO to update the solutions of individuals with respect to either the solution offered

**Fig. 5** Behaviors of consumption in 2-D and 3-D spaces during 200 samples



**Fig. 6** Pseudocode of AEO algorithm

```

Randomly initialize an ecosystem  $X_i$  (solutions) and calculate the
fitness  $Fit_i$ , and  $X_{best}$ =the best solution found so far.

While the stop criterion is not satisfied do
    //Production//
    For individual  $X_1$ , update its solution using equation (1).

    //Consumption//
    For individual  $X_i$  ( $i=2,\dots,n$ ),
        // Herbivore //
        If  $rand < 1/3$  then update its solution using equation (6),
        // Omnivore //
        Else If  $1/3 \leq rand \leq 2/3$  then update its solution using equation (7),
        // Carnivore //
        Else update its solution using equation (8),
        End If.
    End If.

    Calculate the fitness of each individual.

    Update the best solution found so far  $X_{best}$ .
    // Decomposition//
    Update the position of each individual using equation (9).
    Calculate the fitness of each individual.

    Update the best solution found so far  $X_{best}$ .
End While.

Return  $X_{best}$ .

```

by the production process or the solution of the randomly chosen individual with higher energy level, or both. It can improve the exploration of AEO.

4. The consumption factor encourages AEO to perform a global search.
5. In consumption process, each consumer is randomly chosen as a carnivore, a herbivore, or an omnivore with the same probability. A herbivore updates its solution based on the solution offered by the production process. A carnivore updates its solution based on the randomly chosen individual with the higher energy level. An omnivore updates its solution based on between both the solution offered by the production process and the randomly chosen solution with higher energy level.
6. The decomposition enables AEO to update the solutions of individuals based on the best solution in the

population via three key coefficients. It can enhance the exploration of AEO.

7. AEO is very simple to implement and does not require any other parameter to be adjusted besides both the number of population and the maximal iteration.

## 2.3 Convergence analysis of the algorithm

### 2.3.1 Markov model of AEO

AEO is a meta-heuristic method. Hence, the theory of the Markov model can be employed to analyze how AEO can converge to the global optimum. To illustrate the Markov model of AEO, the related mathematical descriptions and definitions are provided as follows.

**Definition 1** [78] (*individual state and individual state space*) The individual states of life organisms are

composed of their positions  $X$ ,  $X \in A$ , and  $A$  is the feasible solution. The set of all possible individual states constitute the state space of individuals,  $X = \{X | X \in A\}$ .

**Definition 2** [78] (*population state and population state space*) Population states of life organisms are composed of all the individual states in the population;  $s = (x_1, x_2, \dots, x_n)$ ,  $x_i$  is the state of the  $i$ th individual. The set of all possible population states constitute the population state space of individuals  $S = \{s = (x_1, x_2, \dots, x_n) | x_i \in X, 1 \leq i \leq n\}$ .

**Definition 3** (*state transition*) For  $\forall x_i \in s, \forall x_j \in s$ , the state transition of an individual is expressed by  $T_s(x_i) = x_j$  which signifies that the individual state  $x_i$  is transferred to  $x_j$ .

**Theorem 1** In AEO, the probability of the individual state  $x_i$  transferred to  $x_j$  is expressed by  $P(T_s(x_i) = x_j)$

$$P(T_s(x_i) = x_j) = \begin{cases} P_p(T_s(x_i) = x_j) & \text{Production} \\ P_c(T_s(x_i) = x_j) & \text{Consumption} \\ P_d(T_s(x_i) = x_j) & \text{Decomposition} \end{cases} \quad (13)$$

**Proof** Because the position updates of life organisms corresponding to three operations are different, their state transition probabilities are also different. The population positions of organisms can be viewed as a set of points in hyperspace. Therefore, their position updates are the transition of a set of points in hyperspace. According to Definition 3 and the geometric property of AEO, the transition probability of a producer is expressed by

$$P_p(T_s(x_i) = x_j) = \begin{cases} \frac{1}{|x_n - x_{\text{rand}}|} p_0(x_i, x_j) & x_j \in [x_n, x_{\text{rand}}] \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

where

$$p_0(x_i, x_j) = \begin{cases} 1 & f(x_j) < f(x_i) \\ 0 & f(x_j) \geq f(x_i) \end{cases} \quad (15)$$

The transition probability of a consumer is expressed by

$$P_c(T_s(x_i) = x_j) = \begin{cases} \frac{1}{|(C \cdot (x_i(t) - x_1(t))|)} p_0(x_i, x_j) & x_j \in [x_i, x_i(t) + C \cdot (x_i(t) - x_1(t))] \\ \frac{1}{|(C \cdot (x_i(t) - x_j(t))|)} p_0(x_i, x_j) & x_j \in [x_i, x_i(t) + C \cdot (x_i(t) - x_j(t))] \\ \frac{1}{|C \cdot (r_2 \cdot (x_i(t) - x_1(t)) + (1 - r_2)(x_i(t) - x_j(t))|)} p_0(x_i, x_j) & x_j \in [x_i, x_i(t) + C \cdot (r_2 \cdot (x_i(t) - x_1(t)) + (1 - r_2)(x_i(t) - x_j(t)))] \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

The transition probability of a decomposer is expressed by

$$P_d(T_s(x_i) = x_j) = \begin{cases} \frac{1}{|((e+1) \cdot x_n - 2r_3 \cdot x_i)|} p_0(x_i, x_j) & x_j \in [x_i, x_n(t) + D \cdot (e \cdot x_n(t) - h \cdot x_i(t))] \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

□

**Definition 4** (*transition probability of population state*) In AEO, for  $\forall s_i \in S, \forall s_j \in S$ , the transition probability of population state is expressed by  $T_s(s_i) = s_j$  which signifies that the population state  $s_i$  is transferred to  $s_j$ . The transition probability of the population state  $s_i$  transferred to  $s_j$  is expressed by

$$P(T_s(s_i) = s_j) = \prod_{k=1}^n P(T_s(x_{ik}) = s_{jk}) \quad (18)$$

Note that the transition probability of population state  $s_i$  transferred to  $s_j$  is the transition probabilities of all the individual states  $x_i$  transferred to  $x_j$ .

**Theorem 2** In BOA, the state sequence of the population  $\{s(t) : t \geq 0\}$  is a finite homogeneous Markov chain.

**Proof** For  $\forall s(t) \in S$  and  $\forall s(t+1) \in S$ , according to Definition 4, the transition probability  $P(T_s(s(t)) = s(t+1))$  is determined by the transition probabilities of all the individual states  $P(T_s(x(t)) = x(t+1))$ . According to Theorem 1, the transition probability of any individual state  $P(T_s(x(t)) = x(t+1))$  in the population is related to the state  $x(t)$  at the time  $t$ , parameters  $C$  and  $a$ , the producer  $x_1$ , the decomposer  $x_n$ , and the rand number  $r$ . Thus,  $P(T_s(x(t)) = x(t+1))$  is only related to the individual states of life organisms at the time  $t$ , not to the time  $t$ . Therefore, the state sequence of population is homogeneous Markov chain. Because the locations  $x$  of life organisms are finite, the state space of life organisms  $X$  is finite. The number of every population  $s$  is finite, which means the state space of population  $S$  is also finite. Therefore, the state sequence of the population  $\{s(t) : t \geq 0\}$  is a finite homogeneous Markov chain. □

According to [79], stochastic search algorithms are globally convergent and AEO belongs to this category, so we use the convergence criterion of stochastic search algorithms to determine whether AEO is convergent.

### 2.3.2 Convergence analysis of AEO

**Definition 5** (*set of the optimal state*) Assume that  $g$  is the optimal solution of the optimization problem  $\langle A, f \rangle$ . The set of the optimal state of AEO is defined by

$$\begin{aligned} G &= \{s = (x_1, \dots, x_i, \dots, x_n) | f(x_i) = f(g), \quad x_i \in s, s \in S\}, \\ G &\subset S \end{aligned} \quad (19)$$

**Theorem 3** *The set of the optimal state  $\mathbf{G}$  is a closed set on the state space  $S$ .*

**Proof** For  $\forall s_i \in \mathbf{G}$ ,  $s_j \notin \mathbf{G}$  and  $s_i \in S$ , the transition probability of  $T_s(x_i) = x_j$  is

$$P(T_s(s_i) = s_j) = \prod_{k=1}^n P(T_s(x_{ik}) = s_{jk}) \quad (20)$$

Suppose  $\exists x_{i0k} \in \mathbf{G}$  and  $g = x_{i0k}$ , we can get  $P(T_s(x_{i0k}) = x_{jk})$ . Also because  $P(T_s(s_i) = s_j)$ , the set of the optimal state  $\mathbf{G}$  is a closed set on the state space  $S$ .  $\square$

**Theorem 4** *There does not exist the closed set  $M$  in the state space of the population, which makes  $M \cap \mathbf{G} = \emptyset$ .*

**Proof** Apagoge. Assume that there exists the closed set  $M$  in the state space of the population, which makes  $M \cap \mathbf{G} = \emptyset$ . Hypothesize  $s_i = (g, \dots, g, \dots, g) \in G$ ,  $\forall s_j = (x_{j1}, \dots, x_{jk}, \dots, x_{jd}) \in M$ , and  $f(x_{ik}) = f(g)$ . Based on Chapman–Kolmogorov equation, the transition probability of the population state from  $s_i$  to  $s_j$  by  $l$  steps can be given as

$$\begin{aligned} P_{si,sj}^l &= \sum_{s_{r1} \in s} \cdots \sum_{s_{rl-1} \in s} P(T_s(s_i) = s_{r1}) \cdot P(T_s(s_{r1}) = s_{r2}) \\ &\cdots P(T_s(s_{rl-1}) = s_j) \end{aligned} \quad (21)$$

AEO satisfies Eqs. (14)–(17) in Theorem 1 after a finite number of iterations; when the step size  $l$  is the large enough, the transition probability of each term in Eq. (21) by one step is

$$P(T_s(s_{rc+j}) = s_{rc+j+1}) > 0 \quad (22)$$

That is to say  $P_{si,sj}^l > 0$ , we can draw a conclusion that the set  $M$  is not a closed set, which contradicts the premise. Therefore, the Markov chain of life organisms in AEO is irreducible and the state space  $S$  does not contain closed sets except  $\mathbf{G}$ .  $\square$

**Theorem 5** [80] *Assume there is a non-empty closed set  $E$  in the Markov chain and there is no other non-empty closed set  $O$  which can satisfy  $E \cap O = \emptyset$ , then*

$$\lim_{k \rightarrow \infty} P(x_k = j) \begin{cases} \pi_j & j \in E \\ 0 & j \notin E \end{cases} \quad (23)$$

**Theorem 6** *In AEO, when the number of iterations approaches infinity, the state sequence of the population must enter the optimal state set  $\mathbf{G}$ .*

**Proof** Based on Theorems 3–5, the conclusion can be drawn.  $\square$

**Theorem 7** *AEO can converge to the optimal solution globally.*

**Proof** AEO is a heuristic search algorithm, which satisfies global search convergence theorem (Hypothesis 1) [80]. Based on Theorem 6, the probability that AEO fails to find the global optimal solution in continuous infinite search is 0, so

$$\prod_{k=0}^{\infty} P(1 - u_k[B]) = 0 \quad (24)$$

where  $u_k[B]$  is the probability measure of the  $k$ th search solution of AEO on the set  $B$ , which satisfies the global convergence criterion of heuristic algorithms in Hypothesis 2 [80]. During the iterations in AEO, the best position of each individual obtained so far is retained; when  $k$  approaches infinity, the probability measure of the solution in AEO is expressed by

$$\lim_{k \rightarrow \infty} P(x_k \in R_{e,M}) = 1 \quad (25)$$

where  $\{x_k\}_{k=0}^{\infty}$  is the solution sequence produced by AEO during the iterations. According to the sufficient and necessary conditions of heuristic search algorithms, AEO can converge to the global optimal solution.

### 3 Results and discussion

The numerical efficiency of AEO is analyzed on 31 popular benchmark functions to extensively evaluate its performance. These functions are described as unimodal functions, multimodal functions, low-dimensional functions, and composite function, which are listed in Tables 1, 2, 3, and 4, respectively. The unimodal functions are used to verify the exploitation of algorithms. The multimodal and low-dimensional functions are used to check the exploration of algorithms. The composite functions are constructed by the basic and hybrid functions, so they tend to check local optima avoidance of algorithms. The details of these composite functions from the CEC 2014 special session are available in [81]. Meanwhile, all these benchmark functions can be used to analyze the convergence behaviors of algorithms.

For these benchmark functions, the performance of AEO is compared with those of six stochastic optimizers, namely GA [4], PSO [5], DE [8], CS [36], ABC [30, 31], and GSA [46]. Although a number of variants based on them or entirely new algorithms have been developed, they are not well explained in the literature and are documented with incomplete parameter settings, causing the exact replication of experiments and results nearly infeasible [82]. Therefore, the standard versions of optimization algorithms

**Table 1** Unimodal test functions

Name	Function	D	Range	$f_{\text{opt}}$
Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	30	$[-100, 100]^D$	0
Schwefel 2.22	$f_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	30	$[-10, 10]^D$	0
Schwefel 1.2	$f_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30	$[-100, 100]^D$	0
Schwefel 2.21	$f_4(x) = \max_i\{ x_i , 1 \leq i \leq n\}$	30	$[-100, 100]^D$	0
Rosenbrock	$f_5(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i)^2 + (x_i - 1)^2)$	30	$[-30, 30]^D$	0
Step	$f_6(x) = \sum_{i=1}^n (x_i + 0.5)^2$	30	$[-100, 100]^D$	0
Quartic	$f_7(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1)$	30	$[-1.28, 1.28]^D$	0

**Table 2** Multimodal test functions

Name	Function	D	Range	$f_{\text{opt}}$
Schwefel	$f_8(x) = -\sum_{i=1}^n (x_i \sin(\sqrt{ x_i }))$	30	$[-500, 500]^D$	-12,569.5
Rastrigin	$f_9(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)^2$	30	$[-5.12, 5.12]^D$	0
Ackley	$f_{10}(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i) + 20 + e$	30	$[-32, 32]^D$	0
Griewank	$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^n (x_i - 100)^2 - \prod_{i=1}^n \cos(\frac{x_i - 100}{\sqrt{i}}) + 1$	30	$[-600, 600]^D$	0
Penalized	$f_{12}(x) = \frac{\pi}{n} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_i + 1)] + (y_n - 1)^2 \right\} + \sum_{i=1}^{30} u(x_i, s10, 100, 4)$	30	$[-50, 50]^D$	0
Penalized2	$f_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^{29} (x_i - 1)^2 p [1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2 [1 + \sin^2(2\pi x_{30})] \right\} + \sum_{i=1}^{30} u(x_i, 5, 10, 4)$	30	$[-50, 50]^D$	0

**Table 3** Low-dimensional test functions

Name	Function	D	Range	$f_{\text{opt}}$
Foxholes	$f_{14}(x) = \left[ \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{j=1}^2 (x_i - a_{ij})^6} \right]^{-1}$	2	$[-65.536, 65.536]^D$	0.998
Kowalik	$f_{15}(x) = \sum_{i=1}^{11} \left  a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right ^2$	4	$[-5, 5]^D$	$3.075 \times 10^{-4}$
Six-Hump Camel	$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]^D$	-1.0316
Branin	$f_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos x_1 + 10$	2	$[-5, 10] \times [0, 15]$	0.398
Goldstein-Price	$f_{18}(x) = \left[ 1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2) \right] \times \left[ 30 + (2x_1 + 1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2) \right]$	2	$[-2, 2]^D$	3
Hartman 3	$f_{19}(x) = -\sum_{i=1}^4 \exp \left[ -\sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2 \right]$	3	$[0, 1]^D$	-3.86
Hartman 6	$f_{20}(x) = -\sum_{i=1}^4 \exp \left[ -\sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2 \right]$	6	$[0, 1]^D$	-3.322
Shekel 5	$f_{21}(x) = -\sum_{i=1}^5 \left  (x_i - a_i)(x_i - a_i)^T + c_i \right ^{-1}$	4	$[0, 10]^D$	-10.1532
Shekel 7	$f_{22}(x) = -\sum_{i=1}^7 \left  (x_i - a_i)(x_i - a_i)^T + c_i \right ^{-1}$	4	$[0, 10]^D$	-10.4028
Shekel 10	$f_{23}(x) = -\sum_{i=1}^{10} \left  (x_i - a_i)(x_i - a_i)^T + c_i \right ^{-1}$	4	$[0, 10]^D$	-10.5364

**Table 4** Composition test functions

Function	Name	D	Range	$f_{\text{opt}}$
$f_{24}(x)$	Composition function 1 ( $N = 5$ )	30	$[-100, 100]^D$	2300
$f_{25}(x)$	Composition function 2 ( $N = 3$ )	30	$[-100, 100]^D$	2400
$f_{26}(x)$	Composition function 3 ( $N = 3$ )	30	$[-100, 100]^D$	2500
$f_{27}(x)$	Composition function 4 ( $N = 5$ )	30	$[-100, 100]^D$	2600
$f_{28}(x)$	Composition function 5 ( $N = 5$ )	30	$[-100, 100]^D$	2700
$f_{29}(x)$	Composition function 6 ( $N = 5$ )	30	$[-100, 100]^D$	2800
$f_{30}(x)$	Composition function 7 ( $N = 3$ )	30	$[-100, 100]^D$	2900
$f_{31}(x)$	Composition function 8 ( $N = 3$ )	30	$[-100, 100]^D$	3000

**Table 5** Comparisons of results for unimodal functions

Function	Index	AEO	PSO	GA	DE	CS	GSA	ABC
$f_1(x)$	Mean	0.000E+00	2.148E−04	9.247E−03	3.636E−14	9.675E−03	2.195E−17	2.363E−03
	SD	0.000E+00	2.247E−04	3.778E−03	6.064E−14	4.518E−03	6.379E−18	1.525E−03
	Best	0.000E+00	2.985E−06	3.206E−03	1.264E−15	3.270E−03	1.146E−17	7.092E−04
$f_2(x)$	Mean	5.248E−289	2.957E−04	2.126E−02	4.385E−08	1.404E+00	2.283E−08	2.317E−04
	SD	0.000E+00	2.307E−04	5.609E−03	2.530E−08	5.609E−01	3.490E−09	1.530E−04
	Best	3.509E−301	4.155E−05	1.264E−02	1.494E−08	6.309E−01	1.647E−08	8.106E−05
$f_3(x)$	Mean	0.000E+00	2.844E+03	1.053E+03	5.694E+00	4.725E+02	2.216E+02	9.562E+03
	SD	0.000E+00	1.343E+03	3.489E+02	3.910E+00	1.096E+02	7.067E+01	1.750E+03
	Best	0.000E+00	1.143E+03	3.967E+02	9.258E−01	2.900E+02	8.526E+01	5.926E+03
$f_4(x)$	Mean	2.958E−291	1.736E+01	1.046E+00	9.172E+00	3.248E+00	3.451E−09	2.453E+01
	SD	0.000E+00	3.624E+00	2.897E−01	3.999E+00	8.547E−01	7.445E−10	2.284E+00
	Best	1.433E−304	1.089E+01	5.211E−01	2.184E+00	1.597E+00	2.258E−09	1.981E+01
$f_5(x)$	Mean	1.920E+01	9.473E+01	9.856E+01	3.000E+01	3.861E+01	2.669E+01	5.470E+02
	SD	5.460E−01	7.896E+01	5.845E+01	1.765E+01	1.029E+01	2.671E+00	2.099E+02
	Best	1.743E+01	7.618E+00	1.299E+01	4.013E+00	2.872E+01	2.571E+01	2.397E+02
$f_6(x)$	Mean	0.000E+00	1.333E−01	0.000E+00	1.333E−01	0.000E+00	0.000E+00	0.000E+00
	SD	0.000E+00	4.342E−01	0.000E+00	4.342E−01	0.000E+00	0.000E+00	0.000E+00
	Best	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
$f_7(x)$	Mean	6.319E−05	5.644E−02	4.473E−02	2.146E−01	3.085E−02	1.913E−02	9.528E−02
	SD	6.815E−05	2.032E−02	1.355E−02	7.238E−02	7.932E−03	6.870E−03	2.385E−02
	Best	3.848E−06	1.915E−02	1.510E−02	1.158E−01	1.381E−02	7.731E−03	5.285E−02

are used for comparison. As we know, it is difficult for each function to search for a set of appropriate parameters to fit each algorithm. Therefore, for simplicity, for each algorithm, a set of fixed parameters are used to analyze its optimization performance on all benchmark functions. The initial parameters of different algorithms are provided as follows.

**PSO** Inertia weight linearly reduces from 0.9 to 0.2, acceleration coefficients  $c_1 = 2$  and  $c_2 = 2$ .

**GA** Decreasing coefficient  $\gamma = 20$ , mutation rate  $p_m = 0.2$ , crossover rate  $p_c = 0.8$  and crossover adopts roulette wheel method.

**DE** Mutation factor  $F = 0.5$  and crossover rate  $C = 0.5$ .

**CS** Mutation probability  $p_a = 0.25$ .

**GSA** Initial gravitational constant  $G_0 = 100$  and decreasing coefficient  $a = 20$ .

### 3.1 Analysis of exploitation capability

Each of unimodal functions in Table 1 only has one global optimum and has no local optima, so they are generally used to test the exploitation of algorithms. Table 5 shows the comparisons of optimization results offered by different optimizers on unimodal functions given in Table 1. ‘Mean’ indicates the mean of best-so-far solution, ‘SD’ indicates the standard deviation of best-so-far solution, and ‘Best’ indicates the minimum of best-so-far solution. According to Table 5, the results of AEO are superior to those of the other met-heuristic algorithms. AEO has obtained the best

results in terms of the mean, standard deviation, and minimum of best-so-far solution for all the unimodal functions. In particular, for the exploitation, CS provides the worst results on functions  $f_1$  and  $f_2$ , ABC provides the worst results on functions  $f_3$ ,  $f_4$ , and  $f_5$ , and PSO and DE perform the worst for functions  $f_6$  and  $f_7$ , respectively. It is clear that AEO shows the best exploitation capability of all the investigated optimizers. This merit results from the exploitation behavior in AEO previously discussed.

### 3.2 Analysis of exploration capability

Differing from unimodal functions, multimodal functions have an exponentially increasing number of local optima. Therefore, these functions are more suitable to test the exploration capability of algorithms. The comparisons of optimization results offered by different optimizers for multimodal functions are listed in Table 6. Throughout the paper, the best results are italicized. As this table shows, AEO performs the best of all the algorithms for all multimodal functions but function  $f_{13}$ . However, GSA provides the worst results on functions  $f_8$  and  $f_{11}$ . ABC performs the worst on functions  $f_{12}$  and  $f_{13}$ . In addition, DE and CS also perform the worst for functions  $f_8$  and  $f_9$ , respectively. Table 7 shows the comparisons of optimization results offered by different optimizers on low-dimensional multimodal functions. The results of AEO are not inferior to those of the other algorithms except for functions  $f_{15}$  and

$f_{20}$ . Although AEO fails to achieve the best results for these two functions, it only ranks behind CS and DE for function  $f_{15}$  and performs better than PSO for function  $f_{20}$ . These results discover that AEO provides a very good exploration capability. This is owing to the fact that the exploration operator is integrated into AEO.

### 3.3 Analysis of avoidance of local optima

Functions  $f_{24}$ – $f_{31}$  are composite functions, and their different variables are randomly separated into various subdivisions constructed by the basic and hybrid functions. This allows these functions to generate numerous local optima. Also, the global optimum is able to randomly shift from one position to another position during the iterations. Therefore, composite functions are very suitable for evaluating local optima avoidance which is able to well balance exploration and exploitation. The comparisons of optimization results offered by different optimizers on composite functions are provided in Table 8. As this table shows, the results of AEO are superior to all other algorithms for half of composite functions. This reveals that AEO has high local optima avoidance resulting from an appropriate balance between exploration and exploitation. Such capability originates from the fact the production operator guides individuals to search from a randomly generated region far away from the best individual to a region close to the best individual as the iterations increase.

**Table 6** Comparisons of results for multimodal functions

Function	Index	AEO	PSO	GA	DE	CS	GSA	ABC
$f_8(x)$	Mean	<i>-1.146E+04</i>	-5.139E+03	-6.820E+03	-5.310E+03	-8.692E+03	-2.639E+03	-5.124E+03
	SD	3.089E+02	5.777E+02	5.769E+02	6.617E+02	2.352E+02	4.351E+02	4.607E+02
	Best	-1.200E+04	-6.951E+03	-7.832E+03	-7.387E+03	-9.032E+03	-4.189E+03	-6.527E+03
$f_9(x)$	Mean	<i>0.000E+00</i>	3.088E+01	1.240E+01	1.650E+02	8.323E+01	1.585E+01	1.578E+02
	SD	0.000E+00	8.884E+00	2.875E+00	1.751E+01	1.310E+01	3.848E+00	2.130E+01
	Best	0.000E+00	1.593E+01	7.967E+00	1.282E+02	5.889E+01	7.960E+00	1.111E+02
$f_{10}(x)$	Mean	<i>8.882E-16</i>	7.599E-03	2.018E-02	5.413E-08	4.152E+00	3.443E-09	5.329E-02
	SD	0.000E+00	8.229E-03	4.771E-03	2.618E-08	1.487E+00	5.587E-10	4.047E-02
	Best	8.882E-16	9.453E-04	1.408E-02	1.812E-08	2.016E+00	2.655E-09	1.242E-02
$f_{11}(x)$	Mean	<i>0.000E+00</i>	1.477E-02	2.158E-02	2.054E-03	9.626E-02	4.264E+00	1.631E-01
	SD	0.000E+00	1.361E-02	1.021E-02	3.888E-03	4.183E-02	1.588E+00	1.152E-01
	Best	0.000E+00	1.010E-04	6.839E-03	7.994E-15	2.609E-02	1.893E+00	8.277E-03
$f_{12}(x)$	Mean	<i>2.016E-18</i>	3.758E-01	2.977E-05	6.911E-03	1.101E+00	3.399E-02	1.501E+01
	SD	4.097E-18	7.257E-01	3.014E-05	2.630E-02	3.172E-01	5.361E-02	4.671E+00
	Best	4.302E-20	1.435E-04	7.861E-06	1.335E-16	5.652E-01	5.162E-20	7.760E+00
$f_{13}(x)$	Mean	1.831E-03	1.909E-01	6.310E-04	5.306E-02	1.343E-01	<i>2.041E-18</i>	3.626E+01
	SD	4.165E-03	3.879E-01	4.234E-04	2.886E-01	6.310E-02	5.131E-19	1.819E+01
	Best	1.800E-17	2.751E-03	3.591E-05	3.896E-15	6.292E-02	1.328E-18	1.294E+01

**Table 7** Comparisons of results for low-dimensional functions

Function	Index	AEO	PSO	GA	DE	CS	GSA	ABC
$f_{14}(x)$	Mean	9.980E−01	9.980E−01	3.855E+00	9.980E−01	9.980E−01	3.473E+00	9.980E−01
	SD	0.000E+00	0.000E+00	2.766E+00	0.000E+00	0.000E+00	2.529E+00	6.916E−06
	Best	9.980E−01						
$f_{15}(x)$	Mean	3.543E−04	3.666E−04	1.175E−03	3.075E−04	3.075E−04	2.358E−03	5.097E−04
	SD	2.351E−04	1.908E−04	1.501E−03	1.420E−19	3.917E−08	1.143E−03	4.988E−05
	Best	3.075E−04	3.075E−04	3.207E−04	3.075E−04	3.075E−04	9.913E−04	4.036E−04
$f_{16}(x)$	Mean	−1.032E+00						
	SD	6.775E−16	6.775E−16	4.277E−10	6.775E−16	6.775E−16	5.532E−16	6.649E−16
	Best	−1.032E+00						
$f_{17}(x)$	Mean	3.979E−01						
	SD	0.000E+00	0.000E+00	1.225E−08	0.000E+00	0.000E+00	0.000E+00	6.751E−10
	Best	3.979E−01						
$f_{18}(x)$	Mean	3.000E+00						
	SD	2.070E−15	2.188E−15	1.837E−07	2.030E−15	1.882E−15	1.690E−15	2.861E−11
	Best	3.000E+00						
$f_{19}(x)$	Mean	−3.863E+00						
	SD	2.710E−15	2.710E−15	1.454E−09	2.710E−15	2.710E−15	2.449E−15	1.668E−15
	Best	−3.863E+00						
$f_{20}(x)$	Mean	−3.259E+00	−3.254E+00	−3.298E+00	−3.286E+00	−3.322E+00	−3.322E+00	−3.322E+00
	SD	6.033E−02	6.042E−02	4.837E−02	5.542E−02	1.917E−13	1.355E−15	1.525E−15
	Best	−3.322E+00						
$f_{21}(x)$	Mean	−1.015E+01	−6.131E+00	−6.993E+00	−9.985E+00	−1.015E+01	−6.976E+00	−1.011E+01
	SD	7.227E−15	2.836E+00	3.700E+00	9.224E−01	3.803E−14	3.549E+00	1.683E−01
	Best	−1.015E+01						
$f_{22}(x)$	Mean	−1.040E+01	−8.218E+00	−9.225E+00	−1.040E+01	−1.040E+01	−1.040E+01	−1.040E+01
	SD	1.512E−15	2.979E+00	2.686E+00	1.745E−15	2.795E−14	0.000E+00	3.529E−14
	Best	−1.040E+01						
$f_{23}(x)$	Mean	−1.054E+01	−7.717E+00	−9.340E+00	−1.054E+01	−1.054E+01	−1.054E+01	−1.054E+01
	SD	1.745E−15	3.251E+00	2.731E+00	1.807E−15	4.200E−12	1.682E−15	8.095E−14
	Best	−1.054E+01						

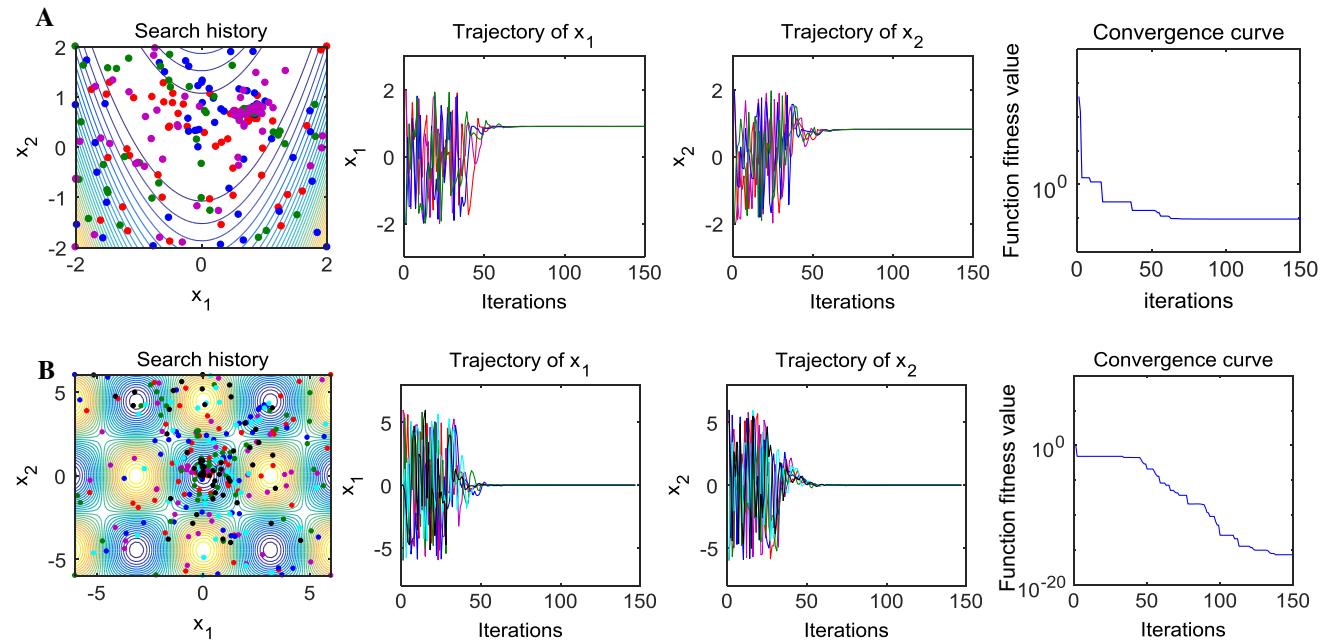
### 3.4 Analysis of convergence behavior

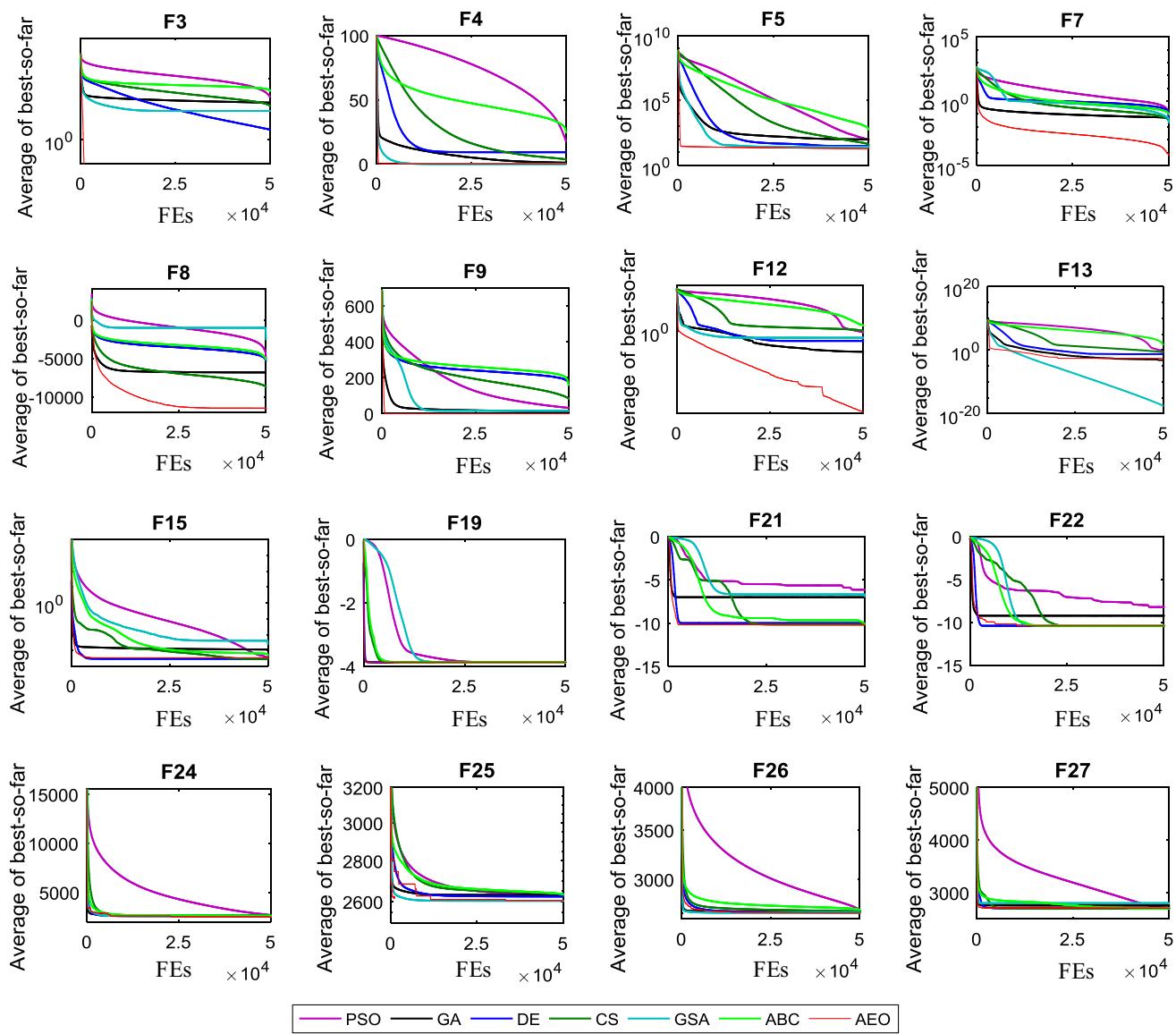
In this section, two functions with two dimensions are firstly chosen to investigate the convergence behavior of AEO. The first function is Rosenbrock function in Table 1. This is a unimodal function, whose global optimum is inside a long, narrow, parabolic-shaped flat valley, hence enabling many algorithms difficult to search for the global optimum. The other one is Griewank function in Table 2. It is a multimodal function with multiple local extrema. These local extrema generated by product terms and tall barriers around the global optimum are regularly distributed, and it allows algorithms to sink into local extrema easily [83]. Hence, these two functions are widely used to evaluate the convergence performance of algorithms.

For Rosenbrock function, four individuals searching for the global optimum simultaneously start at  $(-2, -2)$ ,  $(-2, 2)$ ,  $(2, -2)$ , and  $(2, 2)$  with 150 iterations. For Griewank function, six individuals searching for the global optimum simultaneously start at  $(-6, -6)$ ,  $(-6, 6)$ ,  $(6, -6)$ ,  $(0, -6)$ ,  $(0, 6)$ , and  $(6, 6)$  with 150 iterations. The search history, trajectory, and convergence curve are depicted in Fig. 7. As shown in the figure, all the history positions of search individuals during iterations are clearly provided, and the distribution of sampled points in the search space can discover that how AEO performs exploration or exploitation. The sparse sampled points show the process of exploring the promising regions, and the dense sampled points indicate the process of exploiting a local region. It can be found that the sampled points are sparse in

**Table 8** Comparisons of results for composition functions

Function	Index	AEO	PSO	GA	DE	CS	GSA	ABC
$f_{24}(x)$	Mean	2.529E+03	2.645E+03	2.616E+03	2.615E+03	2.615E+03	2.535E+03	2.617E+03
	SD	1.510E+02	8.508E+00	4.168E-01	2.968E-05	3.295E-03	6.435E+01	4.374E-01
	Best	2.501E+03	2.630E+03	2.616E+03	2.615E+03	2.615E+03	2.506E+03	2.616E+03
$f_{25}(x)$	Mean	2.605E+03	2.631E+03	2.631E+03	2.624E+03	2.633E+03	2.608E+03	2.633E+03
	SD	4.811E+00	5.665E+00	4.335E+00	4.393E+00	3.840E+00	4.734E+00	1.750E+00
	Best	2.600E+03	2.621E+03	2.627E+03	2.602E+03	2.629E+03	2.600E+03	2.629E+03
$f_{26}(x)$	Mean	2.701E+03	2.721E+03	2.714E+03	2.704E+03	2.711E+03	2.702E+03	2.727E+03
	SD	1.429E+00	6.793E+00	3.014E+00	6.924E-01	1.631E+00	3.260E+00	2.056E+00
	Best	2.700E+03	2.712E+03	2.706E+03	2.703E+03	2.709E+03	2.700E+03	2.722E+03
$f_{27}(x)$	Mean	2.702E+03	2.709E+03	2.757E+03	2.704E+03	2.706E+03	2.789E+03	2.705E+03
	SD	1.606E+01	2.500E+01	5.025E+01	1.819E+01	1.793E+01	2.416E+01	1.202E-01
	Best	2.700E+03	2.700E+03	2.700E+03	2.700E+03	2.700E+03	2.722E+03	2.700E+03
$f_{28}(x)$	Mean	3.655E+03	3.361E+03	3.495E+03	3.049E+03	3.122E+03	4.508E+03	3.350E+03
	SD	3.844E+02	2.339E+02	3.693E+02	5.072E+01	5.888E+01	3.007E+02	6.293E+01
	Best	2.908E+03	3.135E+03	3.108E+03	3.000E+03	2.815E+03	3.643E+03	3.186E+03
$f_{29}(x)$	Mean	7.013E+03	7.051E+03	6.796E+03	3.663E+03	3.837E+03	5.546E+03	4.138E+03
	SD	2.217E+03	8.762E+02	8.252E+02	6.744E+01	6.050E+01	1.051E+03	5.010E+01
	Best	3.009E+03	5.422E+03	5.437E+03	3.482E+03	3.734E+03	3.724E+03	4.066E+03
$f_{30}(x)$	Mean	9.370E+07	1.016E+07	4.107E+03	3.963E+03	7.298E+03	3.440E+03	1.600E+05
	SD	5.659E+07	1.528E+07	1.481E+02	2.820E+02	2.115E+03	1.040E+03	8.275E+04
	Best	4.152E+04	3.778E+03	3.903E+03	3.445E+03	5.374E+03	3.100E+03	5.690E+04
$f_{31}(x)$	Mean	1.198E+06	1.162E+05	1.331E+04	4.210E+03	7.445E+03	2.190E+05	2.671E+04
	SD	6.479E+05	7.905E+04	3.082E+03	4.568E+02	7.818E+02	9.727E+04	6.191E+03
	Best	1.992E+05	3.205E+04	8.230E+03	3.533E+03	5.631E+03	7.016E+04	1.724E+04

**Fig. 7** Convergence behaviors of AEO on functions, **a** Rosenbrock function, **b** Griewank function



**Fig. 8** Comparisons of convergence curves for some benchmark functions

the region further from the global optimum and are dense in the region closer to the global optimum.

The trajectory curves can effectively depict the explorative and exploitative search of AEO, which may trace the positions of all individuals in different dimensions during iterations. In the figure, the trajectory of each variable is depicted, respectively. It can be seen that all the trajectories abruptly change in early iterations, and it allows AEO to explore the entire search space extensively. Then, these abrupt changes gradually reduce to the global optimum in later iterations, and it assists AEO to intensively exploit the local space. These trajectories change in the same as what we see in the search history. In addition, we may observe that AEO tends to perform explorative search before exploitative search in the trajectory curves.

The convergence curves of all the investigated optimizers for some benchmark functions are shown in Fig. 8 to observe their convergence performance. The mean of best-so-far in this figure denotes the mean of the best solutions obtained so far from each algorithm over 30 independent runs. The convergence curves reported in Fig. 8 reveal that AEO converges faster than all other optimizers in most benchmark functions and is much promising in the other cases. It can be concluded that AEO has an excellent convergence capability for these benchmark functions.

### 3.5 Statistical significance analysis

Wilcoxon signed-rank test (WSRT) can effectively assess the overall performance of algorithms. WSRT is a

**Table 9** Statistical comparisons of WSRT for AEO versus GA, PSO, and DE

Function	AEO versus PSO				AEO versus GA				AEO versus DE			
	p value	T-	T+	Winner	p value	T-	T+	Winner	p value	T-	T+	Winner
$f_1(x)$	1.73E–06	0	465	–	1.73E–06	0	465	–	1.73E–06	0	465	–
$f_2(x)$	1.73E–06	0	465	–	1.73E–06	0	465	–	1.73E–06	0	465	–
$f_3(x)$	1.73E–06	0	465	–	1.73E–06	0	465	–	1.73E–06	0	465	–
$f_4(x)$	1.73E–06	0	465	–	1.73E–06	0	465	–	1.73E–06	0	465	–
$f_5(x)$	3.18E–06	6	459	–	1.92E–06	1	464	–	0.00049	63	402	–
$f_6(x)$	0.25	0	465	=	1	0	465	=	0.25	0	465	=
$f_7(x)$	1.73E–06	0	465	–	1.73E–06	0	465	–	1.73E–06	0	465	–
$f_8(x)$	1.73E–06	0	465	–	1.73E–06	0	465	–	1.73E–06	0	465	–
$f_9(x)$	1.73E–06	0	465	–	1.73E–06	0	465	–	1.73E–06	0	465	–
$f_{10}(x)$	1.73E–06	0	465	–	1.73E–06	0	465	–	1.73E–06	0	465	–
$f_{11}(x)$	1.73E–06	0	465	–	1.73E–06	0	465	–	1.73E–06	0	465	–
$f_{12}(x)$	1.73E–06	0	465	–	1.73E–06	0	465	–	1.73E–06	0	465	–
$f_{13}(x)$	2.13E–06	2	463	–	0.057096	140	325	=	0.813017	221	244	=
$f_{14}(x)$	1	0	465	=	1.22E–05	0	465	–	1	0	465	=
$f_{15}(x)$	0.000283	56	409	–	2.16E–05	26	439	–	0.001626	341	124	+
$f_{16}(x)$	1	0	465	=	0.125	0	465	=	1	0	465	=
$f_{17}(x)$	1	0	465	=	0.003906	0	465	–	1	0	465	=
$f_{18}(x)$	0.328125	7	458	=	9.80E–05	1	464	–	0.148438	35	430	=
$f_{19}(x)$	1	0	465	=	3.79E–06	0	465	–	1	0	465	=
$f_{20}(x)$	0.082328	117	348	=	0.643517	255	210	=	0.175346	114	351	=
$f_{21}(x)$	9.90E–06	0	465	–	1.73E–06	0	465	–	1	0	465	=
$f_{22}(x)$	7.06E–05	6	459	–	1.73E–06	0	465	–	0.039063	40	425	–
$f_{23}(x)$	0.000237	1.5	463.5	–	1.73E–06	0	465	–	0.5	0	465	=
$f_{24}(x)$	0.135908	305	160	=	0.023038	122	343	–	0.021827	121	344	–
$f_{25}(x)$	1.73E–06	0	465	–	1.73E–06	0	465	–	2.13E–06	2	463	–
$f_{26}(x)$	1.73E–06	0	465	–	1.73E–06	0	465	–	1.49E–05	22	443	–
$f_{27}(x)$	1.02E–05	18	447	–	0.382034	275	190	=	6.98E–05	168	297	–
$f_{28}(x)$	0.000664	398	67	+	0.165027	300	165	=	0.07432	305	160	=
$f_{29}(x)$	0.813017	244	221	=	0.571646	260	205	=	1.36E–05	444	21	+
$f_{30}(x)$	4.73E–06	455	10	+	1.73E–06	465	0	+	1.73E–06	465	0	+
$f_{31}(x)$	1.73E–06	465	0	+	1.73E–06	465	0	+	1.73E–06	465	0	+

nonparametric test which can be used to check for statistical significance difference between two algorithms. To determine whether the one outperforms the other one with certain statistical significance, the size of the ranks provided by WSRT is examined. When using WSRT, the  $R+$  and  $R-$  related to the comparisons between two algorithms can be calculated and their  $p$  values can be obtained. The test results of WSRT on 31 benchmark functions in 30 runs are presented in Tables 9 and 10, where ‘=’ indicates the case in which there is no statistically significant difference between AEO and the comparative algorithm, ‘+’ indicates the case in which the null hypothesis will be rejected and AEO shows a better

performance than the comparative algorithm at 95% significance level ( $\alpha = 0.05$ ) and ‘–’ vice versa. The sum of ‘+’, ‘–’, and ‘=’ for each function in 30 runs is summarized, and the corresponding statistical results are listed in Table 11. The results demonstrate that compared with its counterparts, AEO shows a better performance at 95% significance level for the majority of functions; although no significant difference is found between AEO and any of other optimizers on function  $f_6$ , its mean of best-so-far solution equals the global optimum. From Table 11, the statistical analysis shows the fact that AEO considerably outperforms the other optimizers on four typical test functions.

**Table 10** Statistical comparisons of WSRT for AEO versus CS, GSA, and ABC

Function	AEO versus CS				AEO versus GSA				AEO versus ABC			
	p value	T-	T+	Winner	p value	T-	T+	Winner	p value	T-	T+	Winner
$f_1(x)$	1.73E–06	0	465	–	1.73E–06	0	465	–	1.73E–06	0	465	–
$f_2(x)$	1.73E–06	0	465	–	1.73E–06	0	465	–	1.73E–06	0	465	–
$f_3(x)$	1.73E–06	0	465	–	1.73E–06	0	465	–	1.73E–06	0	465	–
$f_4(x)$	1.73E–06	0	465	–	1.73E–06	0	465	–	1.73E–06	0	465	–
$f_5(x)$	1.73E–06	0	465	–	1.73E–06	0	465	–	1.73E–06	0	465	–
$f_6(x)$	1	0	465	=	1	0	465	=	1	0	465	=
$f_7(x)$	1.73E–06	0	465	–	1.73E–06	0	465	–	1.73E–06	0	465	–
$f_8(x)$	1.73E–06	0	465	–	1.73E–06	0	465	–	1.73E–06	0	465	–
$f_9(x)$	1.73E–06	0	465	–	1.72E–06	0	465	–	1.73E–06	0	465	–
$f_{10}(x)$	1.73E–06	0	465	–	1.73E–06	0	465	–	1.73E–06	0	465	–
$f_{11}(x)$	1.73E–06	0	465	–	1.73E–06	0	465	–	1.73E–06	0	465	–
$f_{12}(x)$	1.73E–06	0	465	–	0.571534	205	260	=	1.73E–06	0	465	–
$f_{13}(x)$	1.73E–06	0	465	–	1.73E–06	465	0	+	1.73E–06	0	465	–
$f_{14}(x)$	1	0	465	=	2.56E–06	0	465	–	1.73E–06	0	465	–
$f_{15}(x)$	0.000359	59	406	–	1.73E–06	0	465	–	3.11E–05	30	435	–
$f_{16}(x)$	1	0	465	=	2.21E–05	0	465	–	0.5	0	465	=
$f_{17}(x)$	1	0	465	=	1	0	465	=	1.73E–06	0	465	–
$f_{18}(x)$	0.15332	67	398	=	4.82E–06	0	465	–	1.73E–06	0	465	–
$f_{19}(x)$	1	0	465	=	2.21E–05	0	465	–	5.99E–07	0	465	–
$f_{20}(x)$	0.008728	360	105	+	8.72E–05	168	297	–	0.00047	232	233	–
$f_{21}(x)$	9.57E–07	0	465	–	1.71E–06	0	465	–	1.71E–06	0	465	–
$f_{22}(x)$	9.25E–06	7	458	–	4.59E–06	0	465	–	1.72E–06	0	465	–
$f_{23}(x)$	1.67E–06	0	465	–	7.85E–07	0	465	–	1.73E–06	0	465	–
$f_{24}(x)$	0.021827	121	344	–	0.093132	278	187	=	0.023038	122	343	–
$f_{25}(x)$	1.73E–06	0	465	–	9.32E–05	302	163	–	1.73E–06	0	465	–
$f_{26}(x)$	1.73E–06	0	465	–	0.749871	217	248	=	1.73E–06	0	465	–
$f_{27}(x)$	0.07387	145	320	=	0.002255	84	381	–	0.07306	205	260	=
$f_{28}(x)$	0.10891	236	229	=	1.73E–06	0	465	–	0.08319	205	260	=
$f_{29}(x)$	1.64E–05	442	23	+	0.05667	146	319	=	3.11E–05	435	30	+
$f_{30}(x)$	1.73E–06	465	0	+	1.73E–06	465	0	+	1.92E–06	464	1	+
$f_{31}(x)$	1.73E–06	465	0	+	1.73E–06	465	0	+	1.73E–06	465	0	+

**Table 11** Statistical results of WSRT obtained by AEO

Function type	AEO versus GA (+/-/-)	AEO versus PSO (+/-/-)	AEO versus DE (+/-/-)	AEO versus CS (+/-/-)	AEO versus GSA (+/-/-)	AEO versus ABC (+/-/-)
Unimodal	6/1/0	6/1/0	6/1/0	6/1/0	6/1/0	6/1/0
Multimodal	6/0/0	5/1/0	5/1/0	6/0/0	4/1/1	6/0/0
Low dimension	4/6/0	8/2/0	1/8/1	4/5/1	9/1/0	9/1/0
Composition	3/2/3	3/3/2	4/1/3	3/2/3	3/3/2	3/2/3
Total	19/9/3	22/7/2	16/11/4	19/8/4	22/6/3	24/4/3

**Table 12** MAE provided by various algorithms for 31 test functions

Function	AEO	PSO	GA	DE	CS	GSA	ABC
$f_1(x)$	0	2.150E−04	9.250E−03	3.640E−14	9.680E−03	2.200E−17	2.360E−03
$f_2(x)$	5.250E−289	2.960E−04	2.130E−02	4.390E−08	1.400E+00	2.280E−08	2.320E−04
$f_3(x)$	0	2840	1050	5.69	473	222	9560
$f_4(x)$	2.960E−291	17.4	1.05	9.17	3.25E+00	3.450E−09	24.5
$f_5(x)$	19.2	94.7	98.6	30	38.6	26.7	547
$f_6(x)$	0	0.133	0	0.133	0	0	0
$f_7(x)$	6.320E−05	0.0564	4.470E−02	0.215	0.0309	1.910E−02	9.530E−02
$f_8(x)$	1069.5	7429.5	5.750E+03	7259.5	3879.5	9929.5	7449.5
$f_9(x)$	0	30.9	12.4	165	83.2	15.9	158
$f_{10}(x)$	8.880E−16	0.0076	2.020E−02	5.410E−08	4.150E+00	3.440E−09	5.330E−02
$f_{11}(x)$	0	0.0148	2.160E−02	2.050E−03	0.0963	4.26	0.163
$f_{12}(x)$	2.020E−18	3.760E−01	2.980E−05	6.910E−03	1.1	3.400E−02	15
$f_{13}(x)$	1.830E−03	0.191	6.310E−04	0.0531	1.34E−01	2.040E−18	36.3
$f_{14}(x)$	0	0	2.862	0	0	2.472	0
$f_{15}(x)$	4.600E−05	5.900E−05	8.720E−04	0	0	2.052E−03	2.020E−04
$f_{16}(x)$	1.600E−03	1.600E−03	1.600E−03	1.600E−03	0.0016	1.600E−03	1.600E−03
$f_{17}(x)$	0	0	0	0	0	0	0
$f_{18}(x)$	0	0	0	0	0	0	0
$f_{19}(x)$	0	0	0	0	0	0	0
$f_{20}(x)$	6.200E−02	0.072	0.022	3.200E−02	0.002	2.000E−03	2.000E−03
$f_{21}(x)$	4.680E−02	4.0232	3.1632	1.632E−01	0.0468	3.1732	0.0532
$f_{22}(x)$	2.800E−03	2.1828	1.1728	2.800E−03	0.0028	2.800E−03	2.800E−03
$f_{23}(x)$	3.640E−02	2.8164	1.1964	3.640E−02	0.0364	3.640E−02	0.0364
$f_{24}(x)$	230	350	320	320	320	240	320
$f_{25}(x)$	210	230	230	220	230	210	230
$f_{26}(x)$	200	220	210	200	210	200	230
$f_{27}(x)$	100	110	160	100	110	190	110
$f_{28}(x)$	960	660	800	350	420	1810	650
$f_{29}(x)$	4210	4250	4000	860	1040	2750	1340
$f_{30}(x)$	9.370E+07	1.020E+07	1210	1060	4400	540	157,100
$f_{31}(x)$	1.197E+06	1.130E+05	10,300	1210	4450	216,000	23,700

### 3.6 Quantitative rank analysis

In order to quantitatively rank the 7 algorithms on the 31 considered test functions, the mean absolute error (MAE) of the optimal solutions of these test functions provided by the algorithms is employed. The MAE is a valid statistical index that can effectively indicate how far the optimal solutions provided by the algorithms from the actual optimal solutions [82]. The MAE is calculated as:

$$\text{MAE} = \frac{\sum_{i=1}^N |f_i - f^*|}{N} \quad (26)$$

where  $f^*$  is the actual optimal value of a function,  $f_i$  is the optimal value provided by an algorithm at  $i$ th run, and  $N$  is the number of runs an algorithm performs for each function.

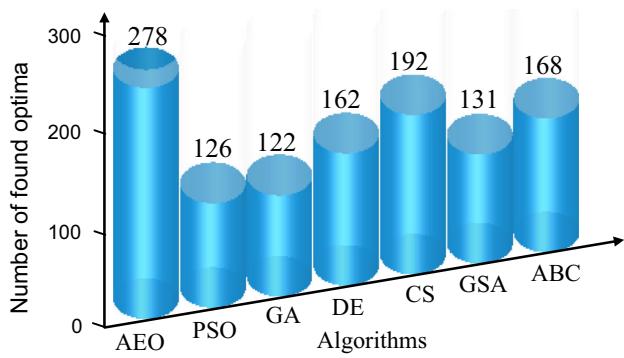
The MAE obtained from different algorithms on 31 test functions is given in Table 12. Table 13 shows the overall rank of all algorithms on these functions, and the last row of the table lists the mean of rank of each algorithm. It can be observed that AEO is ranked the first on average since AEO shows the best MAE for most of test functions. Also the comparisons of the algorithms in finding the optima out of 930 runs (30 runs for 31 test functions) are depicted in Fig. 9. As shown in this figure, AEO is able to reach the global optima 278 times out of 930 runs, demonstrating its best optimization performance in all cases.

### 3.7 Sensitivity analysis

AEO algorithm has two control parameters: the maximal number of iterations and the size of population. The

**Table 13** Rank of algorithms for 31 test functions using MAE

Function	AEO	PSO	GA	DE	CS	GSA	ABC
$f_1(x)$	1	4	6	3	7	2	5
$f_2(x)$	1	5	6	3	7	2	4
$f_3(x)$	1	6	5	2	4	3	7
$f_4(x)$	1	6	3	5	4	2	7
$f_5(x)$	1	5	6	3	4	2	7
$f_6(x)$	1	6	2	7	3	4	5
$f_7(x)$	1	5	4	7	3	2	6
$f_8(x)$	1	5	3	4	2	7	6
$f_9(x)$	1	4	2	7	5	3	6
$f_{10}(x)$	1	4	5	3	7	2	6
$f_{11}(x)$	1	3	4	2	5	7	6
$f_{12}(x)$	1	5	2	3	6	4	7
$f_{13}(x)$	3	6	2	4	5	1	7
$f_{14}(x)$	1	2	7	3	4	6	5
$f_{15}(x)$	3	4	6	1	2	7	5
$f_{16}(x)$	1	2	3	4	5	6	7
$f_{17}(x)$	1	2	3	4	5	6	7
$f_{18}(x)$	1	2	3	4	5	6	7
$f_{19}(x)$	1	2	3	4	5	6	7
$f_{20}(x)$	6	7	4	5	1	2	3
$f_{21}(x)$	1	7	5	4	2	6	3
$f_{22}(x)$	1	7	6	2	3	4	5
$f_{23}(x)$	1	7	6	2	3	4	5
$f_{24}(x)$	1	7	3	4	5	2	6
$f_{25}(x)$	1	4	5	3	6	2	7
$f_{26}(x)$	1	6	4	2	5	3	7
$f_{27}(x)$	1	3	6	2	4	7	5
$f_{28}(x)$	6	4	5	1	2	7	3
$f_{29}(x)$	6	7	5	1	2	4	3
$f_{30}(x)$	7	6	3	2	4	1	5
$f_{31}(x)$	7	5	3	1	2	6	4
Mean of rank	2	4.774	4.194	3.290	4.097	4.065	5.581

**Fig. 9** Comparisons of algorithms in finding the optima out of 930 runs

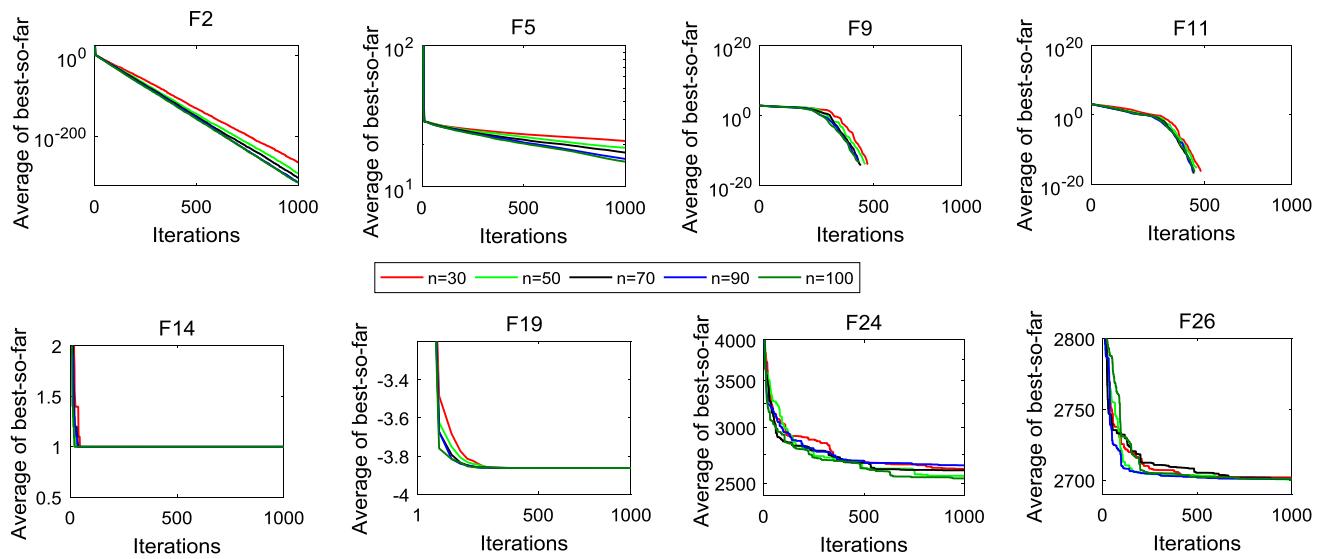
sensitivity of the control parameters is investigated by changing each value of both parameters while keeping the other one unchanged. Two functions chosen from each of the four types of test functions are used for sensitivity analysis.

The size of population: To explore the effect of the population size on the performance of AEO, the size of population is set as 30, 50, 70, 90, and 100, and the maximal number of iterations is set as 1000. AEO algorithm is conducted for the benchmark functions under these different population sizes. For each considered function, AEO algorithm independently runs 30 times for each population size and the results are summarized as the average performance of these runs. Figure 10 and Table 14 provide the convergence curves and optimal results of various types of functions with different population sizes. For all the considered functions, the results obtained by AEO are able to offer very high precision for varying population sizes. Varying the population size does not result in a significant change in precision. AEO obtains better convergence toward the global optimum for varying population sizes.

The maximal number of iterations: The maximal number of iterations is set as 300, 500, 700, 900, and 1000, respectively, and the size of population is fixed as 50. Figure 11 and Table 15 provide the convergence curves and optimal results of various types of functions for different maximal number of iterations. AEO can converge toward the global optimum as the number of iterations increases. AEO provides good convergence rate for different maximal number of iterations.

### 3.8 Scalability analysis

Many optimization problems in the real world include multiple decision variables. It is very necessary for us to investigate the effect of scalability on the problems using AEO algorithm. In this section, unimodal and multimodal functions are used for test by varying the dimensionality of their decision variables, which varies from 45 to 300 with the step 15. The performance comparisons of AEO and other algorithms on these scalable functions are provided in Fig. 12. As shown in the figure, the performance of AEO is less degraded than that of other algorithms on most considered functions as the dimensionality of decision variables increases. For function  $f_{13}$ , although the degradation rate of AEO is worse than that of GA when the dimensionality is less than 120, this degradation of AEO is remarkably improved afterward. The results demonstrate that increasing dimensionality of decision variables has the least effect on the performance of AEO.



**Fig. 10** Convergence curves of various types of functions for different population sizes

**Table 14** Obtained results of various types of functions for different population sizes

n	Index	Functions							
		$f_2$	$f_5$	$f_9$	$f_{11}$	$f_{14}$	$f_{19}$	$f_{24}$	$f_{26}$
30	Mean	8.251E–266	21.16289	0.000E+00	0.000E+00	0.998004	– 3.86278	2620.277	2701.832
	SD	0.000E+00	0.324479	0.000E+00	0.000E+00	0.000E+00	9.367E–16	135.5623	2.175827
50	Mean	7.592E–293	18.94141	0.000E+00	0.000E+00	0.998004	– 3.86278	2561.131	2700.384
	SD	0.000E+00	0.829209	0.000E+00	0.000E+00	0.000E+00	9.363E–16	84.18277	0.366186
70	Mean	7.671E–304	17.52624	0.000E+00	0.000E+00	0.998004	– 3.86278	2608.156	2700.895
	SD	0.000E+00	0.858311	0.000E+00	0.000E+00	0.000E+00	6.326E–16	133.9838	0.868738
90	Mean	2.322E–315	15.80084	0.000E+00	0.000E+00	0.998004	– 3.86278	2651.014	2700.814
	SD	0.000E+00	0.71403	0.000E+00	0.000E+00	0.000E+00	1.431E–16	131.5206	0.936715
100	Mean	1.342E–316	15.10545	0.000E+00	0.000E+00	0.998004	– 3.86278	2539.661	2700.93
	SD	0.000E+00	1.071787	0.000E+00	0.000E+00	0.000E+00	5.381E–16	46.11724	0.661825

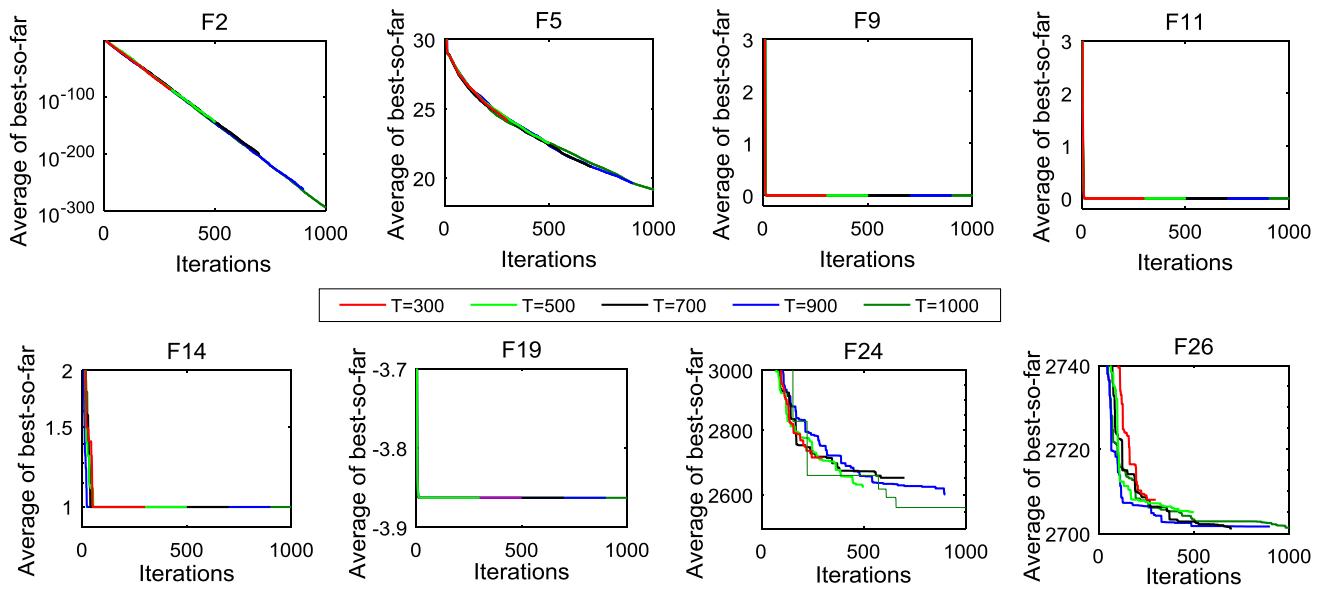
#### 4 Constrained optimization using AEO algorithm

Eight constrained engineering design problems in this part, including welded beam design, three-bar truss design, tension/compression spring design, cantilever beam design, pressure vessel design, speed reducer design, rolling element bearing design, and multiple disk clutch brake design, are utilized to the validity of this proposed algorithm. However, these problems are all constraint optimization problems that involve inequality and equality constraints. For simplicity, penalty functions are used to handle these constraint problems in AEO. When using penalty functions, every search individual who violates any constraint with any level is assigned a big function fitness value. In this way, these constrained problems can be converted into

unconstrained problems by introducing penalty functions. This method is very effective and is easy to implement for AEO without the need for modification of the algorithm. AEO does not need to adjust any other parameter except for the size of population and the maximal number of function evaluations, indicating the consistence of AEO performance among different problems.

##### 4.1 Three-bar truss design

This case, taken from [84] depicted in Fig. 13, is minimization of volume of a statically loaded three-bar truss while satisfying three constraints on stress, deflection, and buckling. This problem needs to optimize two variables ( $x_1$  and  $x_2$ ) to adjust the sectional areas. The optimization



**Fig. 11** Convergence curves of various types of functions for different number of iterations

**Table 15** Obtained results of various types of functions for different number of iterations

T	Index	Functions							
		$f_2$	$f_5$	$f_9$	$f_{11}$	$f_{14}$	$f_{19}$	$f_{24}$	$f_{26}$
300	Mean	4.710E-89	24.49525	0.000E+00	0.000E+00	0.998004	-3.86278	2704.102	2701.304
	SD	5.412E-85	0.515526	0.000E+00	0.000E+00	0.000E+00	1.817E-16	109.782	1.249189
500	Mean	2.827E-142	22.61685	0.000E+00	0.000E+00	0.998004	-3.86278	2611.548	2702.749
	SD	1.903E-148	21.39999	0.000E+00	0.000E+00	0.000E+00	4.71E-16	101.4317	3.695647
700	Mean	2.252E-206	0.443078	0.000E+00	0.000E+00	0.998004	-3.86278	2604.432	2701.462
	SD	9.627E-206	0.858311	0.000E+00	0.000E+00	0.000E+00	3.377E-16	94.99693	1.580398
900	Mean	2.385E-264	20.06242	0.000E+00	0.000E+00	0.998004	-3.86278	2555.139	2701.354
	SD	2.793E-265	0.776597	0.000E+00	0.000E+00	0.000E+00	1.292E-16	55.3923	1.694863
1000	Mean	3.201E-301	18.80743	0.000E+00	0.000E+00	0.998004	-3.86278	2538.878	2700.888
	SD	3.296E-302	0.894924	0.000E+00	0.000E+00	0.000E+00	8.312E-16	62.28167	1.071801

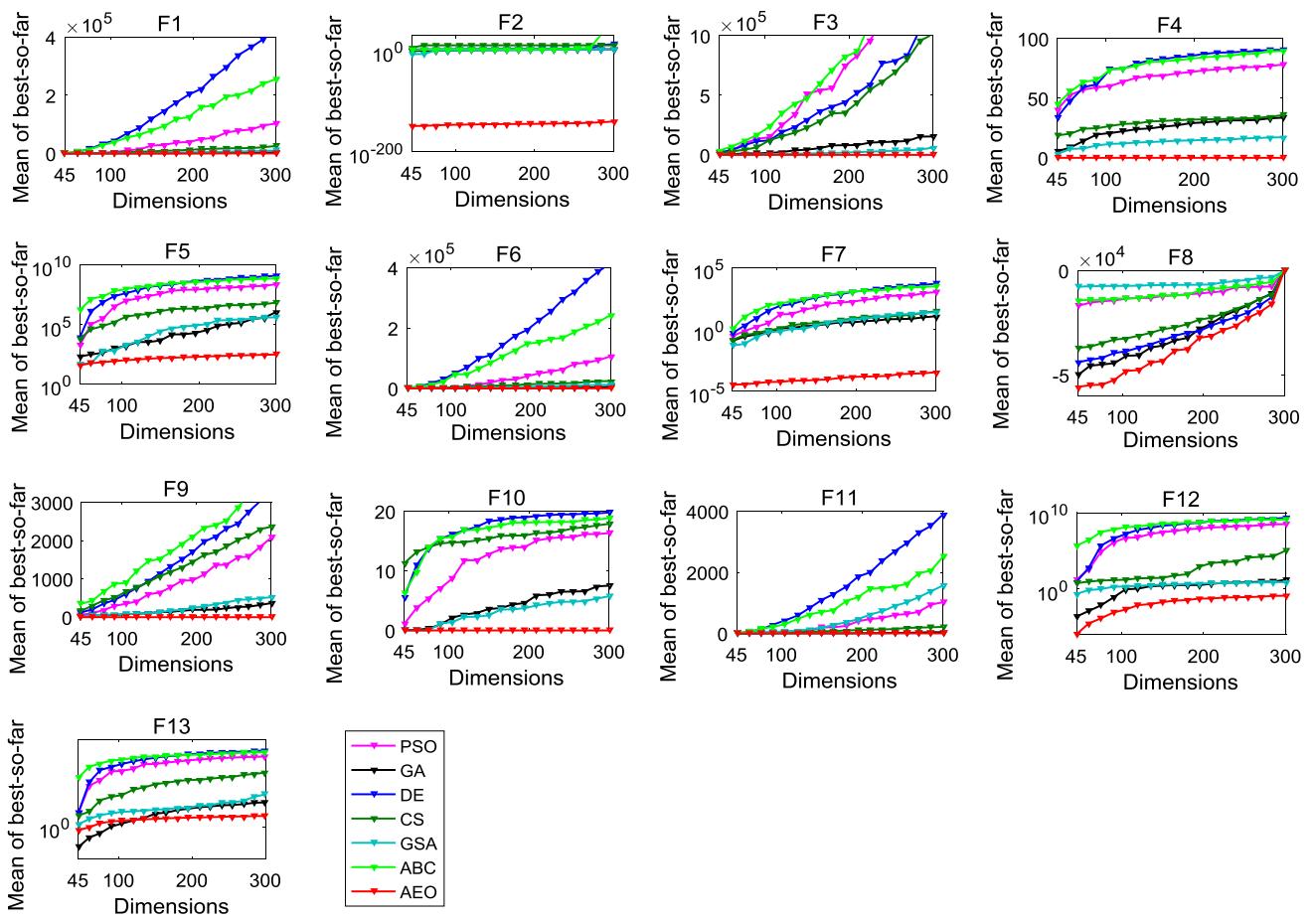
equation for this problem is given in “[Three-bar truss design](#)” in Appendix.

This problem is solved by AEO compared with different approaches in the literature, including SC, PSO-DE, DEDS, HEAA, and CS. Table 16 provides the comparisons of their best results among these methods. A set of statistical results is offered by AEO for 15,000 function evaluations (FEs) in the table. Obviously, AEO provides very competitive results in terms of different indices compared to the other algorithms. The ‘Best’ index of AEO is superior to that of SC and CS; meanwhile, this result is as good as those of PSO-DE, DEDS, and HEAA. In terms of the ‘Worst,’ ‘Mean,’ and ‘SD’ indices, the results offered by AEO are also very competitive. The comparisons of the best solutions offered by the reported algorithms are

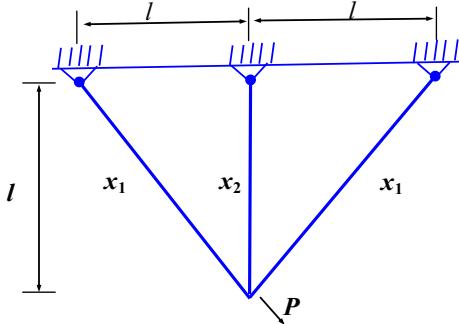
presented in Table 17. The convergence curve of AEO is depicted in Fig. 14. For this case, as shown in the figure, AEO possesses a high convergence rate since it tends to rapidly find a promising region with the global optimum in less than 1500 FEs.

#### 4.2 Cantilever beam design

This problem proposed by [90] is illustrated in Fig. 15, in which the cantilever beam is composed of five hollow square blocks with constant thickness. The beam is rigidly supported at the first block, and there is a vertical force acting at the free end of the fifth block. So this case needs to minimize the weight of the beam and meanwhile only meet the constraint requirement on an upper limit on the



**Fig. 12** Scalability comparisons of algorithms for unimodal and multimodal functions



**Fig. 13** Three-bar truss design problem

vertical displacement of the free end. There are five decision variables that are, respectively, lengths of the different blocks. The optimization equation for this problem is given in “[Cantilever beam design](#)” in Appendix.

This problem is solved using our method, and many reported meta-heuristic approaches, such as SOS, CS, MMA, GCA-I, GCA-II, and MFO, and the comparisons of their statistical results are provided in Table 18. It can be observed that for the ‘Best’ index, AEO offers better results

than the rest of investigated approaches. For the ‘Mean’ index, the results of AEO are as same as those of the other algorithms. Additionally, the lower SD’ index of AEO indicates its higher robustness. The comparisons of best solutions offered by reported algorithms are presented in Table 19. The convergence curve of AEO is depicted in Fig. 16. From this figure, AEO can easily discover a promising region in less than 3500 FEs. This may be owing to the searching mechanism as well as the constraint handling method for AEO, where it firstly tends to investigate promising regions of the problem space extensively and then to exploit the promising region rapidly and intensively.

### 4.3 Tension/compression spring design

This problem, originated from [93], needs to minimize the weight of a tension/compression spring and meet the constraint requirements on shear stress, deflection, and surge frequency as shown in Fig. 17. There are three decision variables: wire diameter ( $d$ ), mean coil diameter ( $D$ ), and

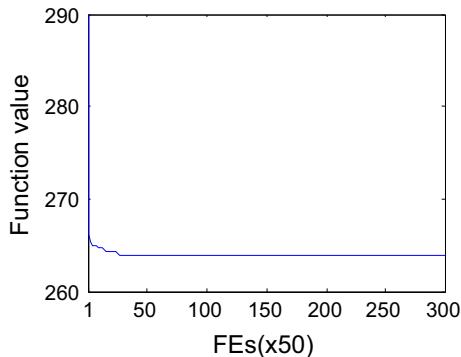
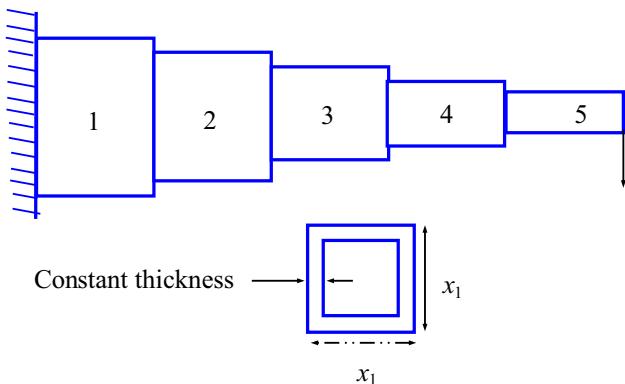
**Table 16** Comparisons of statistical results using reported optimizers in the literature for three-bar truss design

Methods	Worst	Mean	Best	SD	FES
SC [85]	263.969756	263.903356	263.895846	1.3E–02	17,610
PSO-DE [86]	263.895843	263.895843	263.895843	4.5E–10	17,600
DEDS [87]	263.895849	263.895843	263.895843	9.7E–07	15,000
HEAA [88]	263.896099	263.895865	263.895843	4.9E–05	15,000
CS [89]	NA	264.0669	263.97156	9.0E–05	15,000
AEO	263.895892	263.895861	263.895843	5.8E–06	15,000

NA not available

**Table 17** Comparisons of best solutions offered by reported optimizers for three-bar truss design

	SC	PSO-DE	DEDS	HEAA	CS	AEO
$x_1$	0.788621	0.7886751	0.788675	0.788680	0.788670	0.788707
$x_2$	0.408401	0.4082482	0.408248	0.408234	0.409020	0.408159
$g_1$	NA	– 5.29E–11	– 1.77E–08	NA	– 0.00029	– 4.49E–13
$g_2$	NA	– 1.4637475	– 1.464101	NA	– 0.26853	– 1.464104
$g_3$	NA	– 0.5362524	– 0.535898	NA	– 0.73176	– 0.535896
$f_1$	263.895847	263.895843	263.895843	263.895843	263.971623	263.895843

**Fig. 14** Convergence curve of AEO for three-bar truss design**Fig. 15** Cantilever beam design problem

number of active coils ( $N$ ). The optimization equation for this problem is given in “Tension/compression spring design” in Appendix.

The comparisons of statistical results for AEO with the reported methods in the literature, including GA2, GA3, CA, CPSO, HPSO, PSO2, QPSO, UPSO, CDE, SSB, and  $(\mu + \lambda)$ ES, are provided in Table 20. In Table 20, the two sets of statistical results are obtained using AEO for 2000 FEs and 25,000 FEs, respectively. As seen in the table, compared with the other algorithms, AEO under pre-defined maximum function evaluations is very promising. More specifically, for the second set statistical results, AEO is far superior to all the other meta-heuristic methods in terms of four different indices using less number of FEs. The comparisons of best solutions offered by investigated optimizers are presented in Table 21.

Figure 18 illustrates the convergence curve and each constraint value with respect to FEs, respectively. As observed in this figure, the individuals obtain the infeasible solutions violating two constraints in their early iterations, resulting in a higher function value. With the increase in iterations, the individuals quickly obtain feasible solutions satisfying all the constraints, thus rapidly decreasing the function value. Obviously, AEO performs good convergence rate since it tends to reach the best solution in almost 500 iterations. This may be due to a special search mechanism and constraint handling strategy of AEO in which it initially explores the entire problem space and then rapidly approaches the optimal solution.

#### 4.4 Pressure vessel design

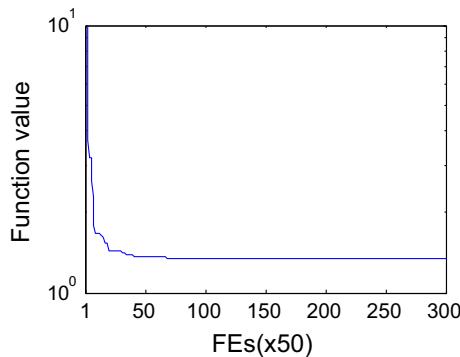
This case, proposed by [103], needs to minimize the fabrication cost of a cylindrical pressure vessel and meet the constraint requirements on buckling load end deflection of

**Table 18** Comparisons of statistical results using reported optimizers in the literature for cantilever beam design

Methods	Worst	Mean	Best	SD	FEs
SOS [91]	NA	1.33997	1.33996	1.1E–05	15,000
CS [89]	NA	NA	1.33999	NA	125,000
MMA [90]	NA	NA	1.3400	NA	NA
GCA-I [90]	NA	NA	1.3400	NA	NA
GCA-II [90]	NA	NA	1.3400	NA	NA
MFO [92]	NA	NA	3.399880	NA	NA
AEO	1.340089	1.339970	1.339965	8.25E–06	15,000

**Table 19** Comparisons of best solution offered by reported optimizers for cantilever beam design

	SOS	CS	MMA	GCA-I	GCA-II	MFO	AEO
$x_1$	6.01878	6.0089	6.0100	6.0100	6.0100	5.984871	6.028850
$x_2$	5.30344	5.3049	5.3000	5.3000	5.3000	5.316726	5.316521
$x_3$	4.49587	4.5023	4.4900	4.4900	4.4900	4.497332	4.462649
$x_4$	3.49896	3.5077	3.4900	3.4900	3.4900	3.513616	3.508455
$x_5$	2.15564	2.1504	2.1500	2.1500	2.1500	2.1616200	2.157761
$g_1$	NA	NA	NA	NA	NA	NA	–1.26E–13
$f_2$	1.33996	1.33999	1.3400	1.3400	1.3400	1.339988	1.339965



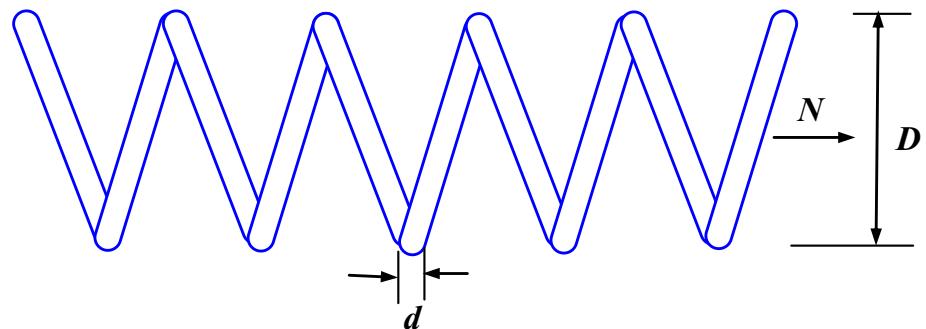
**Fig. 16** Convergence curve of AEO for cantilever beam design

the beam, shear stress and bending stress in the beam. A cylindrical vessel is capped at both ends by hemispherical heads as depicted in Fig. 19. This problem includes four decision variables: thickness of weld ( $T_s$ ), length of the clamped bar ( $T_h$ ), height of the bar ( $R$ ), and thickness of the

bar ( $L$ ). The optimization equation for this problem is given in “Pressure vessel design” in Appendix.

This problem is handled using many reported metaheuristic approaches, such as GA2, GA3, CPSO, HPSO, PSO-DE, PSO2, CDE, ABC, QPSO,  $(\mu + \lambda)$ ES, and CSA. Table 22 shows the statistical results obtained from these approaches with those found by the proposed one, and the two sets of statistical results are obtained using AEO for 8000 FEs and 30,000 FEs, respectively. From this table, for the first set of results, with the same FEs, the results of AEO are superior to those of PSO2 and QPSO in terms of four different indices. For the second set of results, in terms of the ‘Best’ index, with less number of FEs, the results offered by AEO outperform the results offered by all the other algorithms. Meanwhile, in terms of the ‘Worst’ and ‘Mean’ indices, the performance of AEO is promising. The comparisons of best solutions offered by reported optimizers are represented in Table 23.

**Fig. 17** Tension/compression string design problem



**Table 20** Comparisons of statistical results using reported optimizers in the literature for tension/compression spring design

Methods	Worst	Mean	Best	SD	FEs
GA2 [94]	0.0128221	0.0127690	0.0127047	3.9390E–05	900,000
GA3 [95]	0.0129730	0.0127420	0.0126810	5.9000E–05	80,000
CA [96]	0.0151156	0.0135681	0.0127210	8.4215E–04	50,000
CPSO [97]	0.0129240	0.0127300	0.0126747	5.1985E–04	200,000
HPSO [98]	0.0127191	0.0127072	0.0126652	1.5824E–05	75,000
PSO2 [99]	0.0718020	0.0195550	0.0128570	0.0116620	2000
QPSO [99]	0.0181270	0.0138540	0.0126690	1.3410E–03	2000
UPSO [100]	0.0503651	0.0229478	0.0131200	7.2057E–03	10,000
CDE [101]	0.0127900	0.0127030	0.0126702	2.7000E–05	240,000
SSB [85]	0.0167173	0.0129227	0.0126692	5.9000E–04	25,167
$(\mu + \lambda)$ ES [102]	NA	0.0131650	0.0126890	3.9000E–04	30,000
AEO	0.0142060	0.0133186	0.0127324	3.4611E–04	2000
AEO	0.0127271	0.0127271	0.0126662	2.5401E–05	25,000

**Table 21** Comparisons of best solutions offered by reported optimizers for tension/compression spring design

	GA3	CA	CPSO	HPSO	CDE	$(\mu + \lambda)$ ES	AEO
$x_1(d)$	0.051989	0.050000	0.051728	0.051706	0.051609	0.052836	0.051897
$x_2(D)$	0.363965	0.317395	0.357644	0.357126	0.354714	0.384942	0.361751
$x_3(N)$	10.890522	14.031795	11.244543	11.265083	11.410831	9.807729	10.879842
$g_1$	– 0.000013	0	– 8.25E–04	– 3.06E–06	– 0.000039	– 0.000001	– 1.26E–05
$g_2$	– 0.000021	– 0.000075	– 2.52E–05	– 1.39E–06	– 0.000183	0	– 6.75E–08
$g_3$	– 4.061338	– 3.967960	– 4.051306	– 4.054583	– 4.048627	– 4.106146	– 4.063548
$g_4$	– 0.722698	– 0.755070	– 0.727085	– 0.727445	– 0.729118	– 0.708148	– 0.72427
$f_3$	0.0126810	0.0127210	0.0126747	0.0126652	0.0126702	0.012689	0.0126662

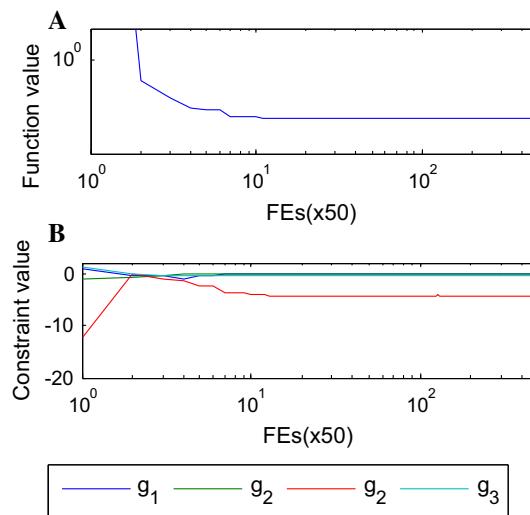
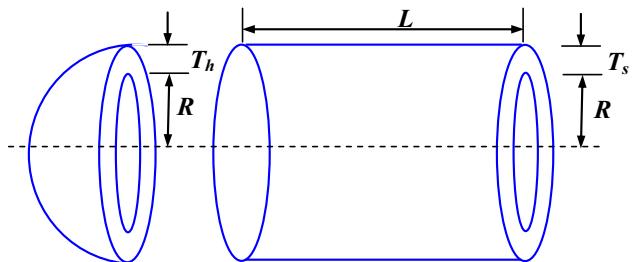
**Fig. 18** **a** Function value versus FEs, **b** constraint value versus FEs

Figure 20 illustrates the convergence curve and each constraint value with respect to FEs, respectively. As shown in this figure, during the initial stage of iterations,

**Fig. 19** Pressure vessel design problem

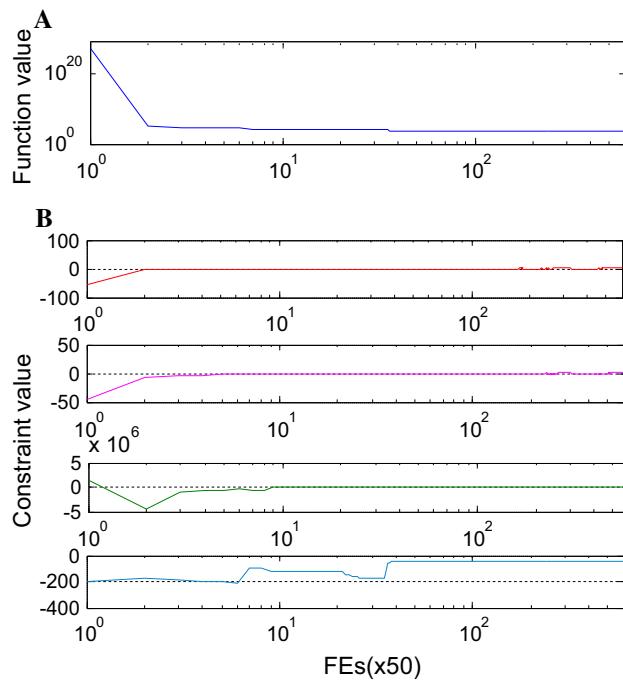
the solutions of individuals are in the infeasible region since the solutions violate the third constraint. As 50 FEs are done, the infeasible solutions are transformed into the feasible ones that are likely to converge to near the global optimum in about 2000 FEs. This characteristic observed in other engineering problems can be viewed as a merit of the proposed approach.

**Table 22** Comparisons of statistical results using reported optimizers in the literature for pressure vessel design

Methods	Worst	Mean	Best	SD	FEs
GA2 [94]	6308.4970	6293.8432	6288.7445	7.4133	900,000
GA3 [95]	6469.3220	6177.2533	6059.9463	130.9297	80,000
CPSO [97]	6363.8041	6147.1332	6061.0777	86.4500	240,000
HPSO [98]	6288.6770	6099.9323	6059.7143	86.2000	81,000
PSO-DE [86]	NA	6059.7140	6059.714	NA	42,100
PSO2 [99]	14,076.3240	8756.6803	6693.7212	1492.5670	8000
QPSO [99]	8017.2816	6440.3786	6059.7209	479.2671	8000
CDE [101]	6371.0455	6085.2303	6059.7340	43.0130	204,800
ABC [104]	NA	6245.3080	6059.714339	205.0000	30,000
$(\mu + \lambda)$ ES [102]	6820.397461	6379.938037	6059.701610	210.0000	30,000
CSA [105]	7332.8410	6342.4990	6059.7140	384.9450	250,000
AEO	7110.1964	6615.7111	6030.0039	216.7381	8000
AEO	6820.8007	6136.3019	59,945.0695	161.2901	30,000

**Table 23** Comparisons of best solutions offered by reported optimizers for pressure vessel design

	GA3	CPSO	HPSO	CDE	ABC	$(\mu + \lambda)$ ES	AEO
$x_1(T_s)$	0.812500	0.812500	0.8125	0.812500	0.8125	0.8125	0.8374205
$x_2(T_h)$	0.437500	0.437500	0.4375	0.437500	0.4375	0.4375	0.413937
$x_3(R)$	42.097398	42.091266	42.0984	42.098411	42.098446	42.098446	43.389597
$x_4(L)$	176.654047	176.746500	176.6366	176.637690	176.636596	76.636596	161.268592
$g_1$	– 0.000020	– 0.000139	– 8.80E–07	– 6.677E–07	0	0	– 1.3666E–6
$g_2$	– 0.035891	– 0.035949	– 3.58E–02	– 0.035881	– 0.035881	– 0.035880	5.3169E–7
$g_3$	– 27.886075	– 116.382700	– 3.12260	– 3.683016	– 0.000226	0	– 2.641622
$g_4$	– 63.345953	– 63.253500	– 63.36340	– 63.36231	– 63.363404	63.363404	– 78.731407
$f_4$	6059.946341	6061.0777	6059.7143	6059.7340	6059.714339	6059.7143	59,945.0695

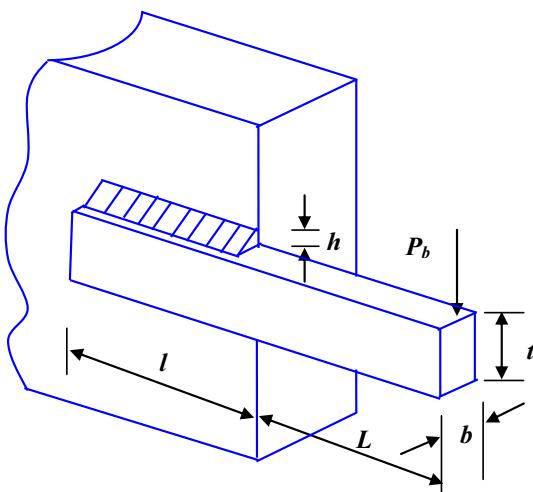


**Fig. 20** **a** Function value versus FEs, **b** constraint value versus FEs

#### 4.5 Welded beam design

In this problem [94], the fabrication cost of a welded beam is to be minimized subject to constraints on shear stress, bending stress in the beam, buckling load on the bar, end deflection of the beam and side constraints. There are four decision variables as shown in Fig. 21, including thickness of weld ( $h$ ), length of attached part of bar ( $l$ ), height of the bar ( $t$ ), and thickness of the bar ( $b$ ). The optimization equation for this problem is given in “[Welded beam design](#)” in Appendix.

The statistical results of some competitive metaheuristic methods, including GA2, GA3, CPSO, HPSO, PSO-DE, WOA, EPSO, ABC,  $(\mu + \lambda)$ ES, and SC, are compared with the proposed approach, and their comparative results are listed in Table 24. In this table, two sets of statistical results found by AEO with 9900 FEs and 15,000 FEs are provided, respectively. As the table shows, for the first set of results, with the same FEs, AEO performs better than WOA in terms of the ‘Mean’ and ‘SD’ indices. For the second set of results, AEO provides more promising results in terms of four indices using much less number of FEs



**Fig. 21** Pressure vessel design problem

than those provided by all the other methods but PSO-DE. However, the results obtained by AEO are slightly inferior to the results given by PSO-DE in terms of the ‘Worst’ and ‘Mean’ indices. The comparisons of best solutions offered by investigated meta-heuristic methods are represented in Table 25. Figure 22 depicts the function value and each constraint value with respect to FEs for the welded beam design problem.

#### 4.6 Speed reducer design

In this problem [102], the weight of speed reducer is to be minimized subject to constraints on bending stress of the gear teeth, surface stress, transverse deflections of the shafts and stresses in the shafts. There are seven decision variables as shown in Fig. 23: face width ( $b$ ), module of teeth ( $m$ ), number of teeth in the pinion ( $z$ ), length of the first shaft between bearings ( $l_1$ ), length of the second shaft

between bearings ( $l_2$ ), and diameter of first ( $d_1$ ) and second shafts ( $d_2$ ). The optimization equation for this problem is given in “Speed reducer design” Appendix.

The results of the proposed algorithm are compared with those of the other considered algorithms in terms of different statistical indexes. These reported methods include SC, DELC, HEAA, PSO-DE, DEDS, ABC,  $(\mu + \lambda)$ ES, and MDE. The comparisons of statistical results are presented in Table 26. As this table shows, the results offered by AEO are as competitive as those by DELC and DEDS in terms of the ‘Worst,’ ‘Mean’ and ‘Best’ indices, which outperform the other considered algorithms except ABC. However, AEO can reach the optimum solution much faster with considerably less computational efforts than all other reported methods. The comparisons of best solutions offered by reported optimizers are given in Table 27. Figure 24 shows the function value and each constraint value with respect to FEs for speed reducer design problem.

#### 4.7 Rolling element bearing design

In this problem [110, 111], the dynamic load-carrying capacity of rolling element bearing, as depicted in Fig. 25, is minimized subject to constraints on the geometry and kinematics. There are ten decision variables: pitch diameter ( $D_m$ ), ball diameter ( $D_b$ ), number of balls ( $Z$ ), inner and outer raceway curvature coefficients ( $f_i$  and  $f_o$ ), and other internal geometry parameters ( $K_{D\min}$ ,  $K_{D\max}$ ,  $\varepsilon$ ,  $e$ , and  $\zeta$ ) that only appear in constraints. For these ten variables, one is the discrete and the others are continuous. The optimization equation for this problem is given in “Rolling element bearing design” in Appendix.

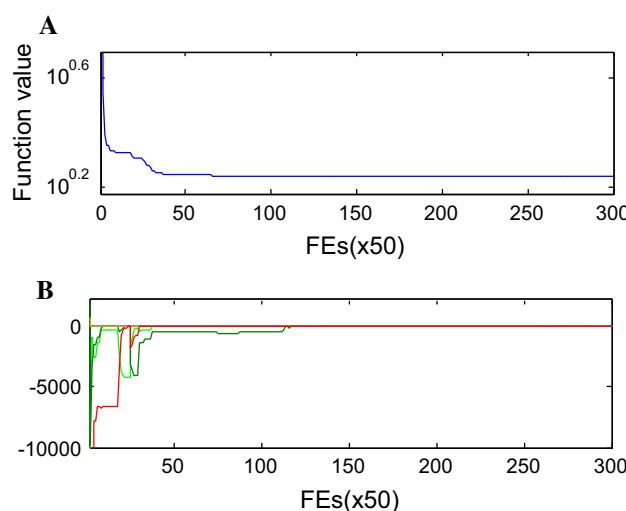
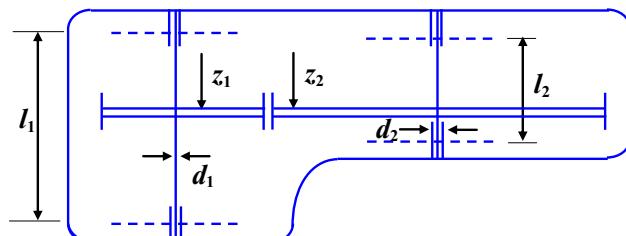
This case was tackled by some researchers with meta-heuristic optimizers, such as GA4, TLBO, ABC, and MBA, and the statistical results of these approaches and AEO are

**Table 24** Comparisons of statistical results using reported optimizers in the literature for welded beam design

Methods	Worst	Mean	Best	SD	FES
GA2 [94]	1.7858350	1.7719730	1.7483090	1.1200E-02	900,000
GA3 [95]	1.9934080	1.7926540	1.7282260	7.4700E-02	80,000
CPSO [97]	1.7821430	1.7488310	1.7280240	1.2900E-02	240,000
HPSO [98]	1.8142950	1.7490400	1.7248520	4.0100E-02	81,000
PSO-DE [86]	1.7248520	1.7248520	1.7248520	6.7000E-16	66,600
CDE [100]	1.8241050	1.7681580	1.7334620	2.2194E-02	204,800
WOA [106]	NA	1.7320000	NA	0.0226000	9900
EPSO [107]	1.7472200	1.7282190	1.7248530	5.6200E-03	50,000
ABC [104]	NA	1.7419130	1.7248520	3.100E-02	30,000
$(\mu + \lambda)$ ES [102]	NA	1.7776920	1.7248520	8.800E-02	30,000
SC [85]	6.3996780	3.0025883	2.3854347	9.600E-01	33,095
AEO	1.7383026	1.7274068	1.7248686	4.39E-03	9900
AEO	1.7255664	1.7250057	1.7248520	2.4763E-04	15,000

**Table 25** Comparisons of best solutions offered by reported optimizers for welded beam design

	GA3	CPSO	HPSO	CDE	ABC	$(\mu + \lambda)$ ES	AEO
$x_1(T_s)$	0.205986	0.202369	0.20573	0.203137	0.205730	0.205730	0.2057296
$x_2(T_h)$	3.471328	3.544214	3.470489	3.542998	3.470489	3.470489	3.4704886
$x_3(R)$	9.020224	9.048210	9.036624	9.033498	9.036624	9.036624	9.0366239
$x_4(L)$	0.206480	0.205723	0.20573	0.206179	0.205730	0.205729	0.2057296
$g_1$	− 0.074092	− 12.839796	− 0.025399	− 44.578568	0	0	− 1.82E−11
$g_2$	− 0.266227	− 1.247467	− 0.053122	− 44.663534	0.000002	0.000002	− 2.18E−11
$g_3$	− 0.000495	− 0.001498	0	− 0.003042	0	0	− 1.23E−12
$g_4$	− 3.430043	− 3.429347	− 3.432981	− 3.423726	− 3.432984	− 3.432984	− 3.4329837
$g_5$	− 0.080986	− 0.079381	− 0.08073	− 0.078137	− 0.08073	− 0.08073	− 0.0807296
$g_6$	− 0.235514	− 0.235536	− 0.235540	− 0.235557	− 0.235540	− 0.235540	− 0.2355403
$g_7$	− 58.666440	− 11.681355	− 0.031555	− 38.028268	0.000001	0.000001	0
$f_5$	1.728226	1.728024	1.724852	1.733462	1.724852	1.724852	1.7248520

**Fig. 22** **a** Function value versus FEs, **b** constraint value versus FEs**Fig. 23** Speed reducer design

compared in Table 28. In terms of the ‘Worst,’ ‘Mean,’ and ‘Best’ indices, AEO provides the best results of all the investigated algorithms using the least number of FEs for this case. The comparisons of best solutions offered by reported optimizers are represented in Table 29. Figure 26 shows the convergence curves for three used algorithms. As Fig. 25 shows, the convergence rate of ABC is nearly

the same as that of TLBO, with a slight advantage for TLBO, and both methods do not seem to achieve the optimum solution in 5000 FEs. Nevertheless, along with a fast convergence rate, AEO can obtain the optimum solution in less than 3500 FEs.

#### 4.8 Multiple disk clutch brake design

This problem, taken from [114] as shown in Fig. 27, needs to minimize the mass of the multiple disk clutch brake using five discrete decision variables: inner radius ( $r_i = 60, 61, 62, \dots, 80$ ), outer radius ( $r_o = 90, 91, 92, \dots, 110$ ), thickness of the disk ( $t = 1, 1.5, 2, 2.5, 3$ ), actuating force ( $F = 600, 610, 620, \dots, 1000$ ), and number of friction surfaces ( $Z = 2, 3, 4, 5, 6, 7, 8, 9$ ). The optimization equation for this problem is given in “[Multiple disc clutch brake design](#)” Appendix.

The comparisons of statistical results for the reported optimizers including NSGA-II, TLBO, and ABC are shown in Table 30. As this table shows, in terms of all the four indices, with less computational efforts, the developed algorithm offers its best results of all the considered optimizers for this case. The comparisons of best solutions offered by reported optimizers are also represented in Table 31. Figure 28 shows the convergence curves for three used algorithms. As observed in Fig. 28a, the convergence rate of TLBO is faster than that of ABC in the early iterations, but with the increase in iterations, the convergence of both algorithms becomes almost same [112]. From Fig. 28b, AEO provides a faster convergence rate than the two optimizers that offer the best solution after about 600 FEs. However, our approach can find the same solution only using 500 FEs.

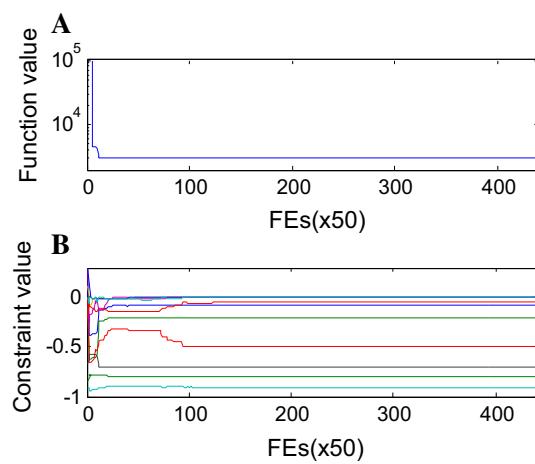
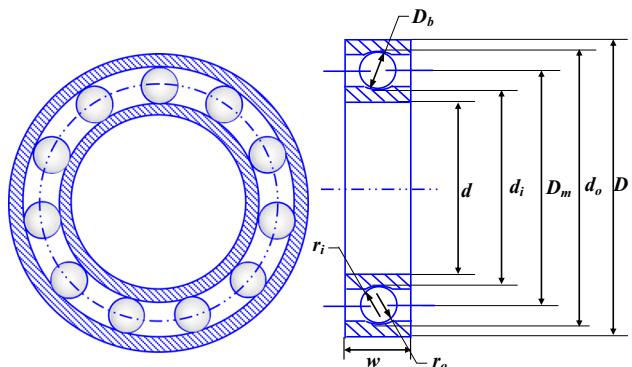
These overall tests suggest that the proposed AEO can be viewed as a penitential alternative in solving various optimization problems. On the one hand, the experimental

**Table 26** Comparisons of statistical results using reported optimizers in the literature for speed reducer design

Methods	Worst	Mean	Best	SD	FEs
SC [85]	3009.9647360	3001.7582640	2994.7442410	4	54,456
DELC [108]	2994.4710660	2994.4710660	2994.4710660	1.9000E–12	30,000
HEAA [88]	2994.7523110	2994.6133680	2994.4991070	7.0000E–02	40,000
PSO–DE [86]	2996.3482040	2996.3481740	2996.3481670	6.4000E–06	54,350
DEDS [87]	2994.4710660	2994.4710660	2994.4710660	3.6000E–12	30,000
ABC [104]	NA	2997.0580000	2997.0580000	0	30,000
$(\mu + \lambda)$ ES [102]	NA	2996.3480940	2996.3480940	0	30,000
MDE [109]	NA	2996.3672200	2996.3566890	8.2000E–03	24,000
AEO	2994.4710667	2994.4710662	2994.4710661	1.2391E–07	22,000

**Table 27** Comparisons of best solutions offered by reported optimizers for speed reducer design

	SC	HEAA	DEDS	ABC	$(\mu + \lambda)$ ES	MDE	AEO
$x_1(b)$	3.50000	3.5000228993	3.5000000000	3.499999	3.499999	3.500010	3.5
$x_2(m)$	0.70000	0.7000003924	0.7000000000	0.7	0.699999	0.700000	0.7
$x_3(z)$	17	17.0000128592	17	17	17	17	17
$x_4(l_1)$	7.327602	7.3004277414	7.3000000000	7.3	7.300000	7.300156	7.3
$x_5(l_2)$	7.715321	7.7153774494	7.7153199115	7.8	7.800000	7.800027	7.7153199
$x_6(d_1)$	3.350267	3.3502309666	3.3502146661	3.350215	3.350215	3.350221	3.3502146
$x_7(d_2)$	5.286655	5.2866636970	5.2866544650	5.287800	5.286683	5.286685	5.2866545
$g_1$	–0.073915	NA	NA	–0.073915	–0.073915	–0.073918	–0.0739152
$g_2$	–0.197999	NA	NA	–0.197999	–0.197998	–0.198001	–0.1979985
$g_3$	–0.493501	NA	NA	–0.499172	–0.499172	–0.499144	–0.4991722
$g_4$	–0.904644	NA	NA	–0.901555	–0.901472	–0.901471	–0.9046439
$g_5$	0	NA	NA	0	0	–0.000005	1.4322E–13
$g_6$	0.000633	NA	NA	0	0	–0.000001	5.9175E–13
$g_7$	–0.7025	NA	NA	–0.702500	–0.702500	–0.702500	–0.7025
$g_8$	0	NA	NA	0	0	–0.000003	6.8002E–13
$g_9$	–0.583333	NA	NA	–0.583333	–0.583333	–0.583332	–0.7958333
$g_{10}$	–0.054889	NA	NA	–0.051326	–0.051325	–0.051345	–0.0513257
$g_{11}$	0	NA	NA	–0.010695	–0.010852	–0.010856	9.0150E–14
$f_6$	2994.744241	2994.499107	2994.471066	2997.058412	2996.348094	2996.356689	2994.4710661

**Fig. 24** **a** Function value versus FEs, **b** constraint value versus FEs**Fig. 25** Rolling element bearing design

**Table 28** Comparisons of statistical results using reported optimizers in the literature for rolling element bearing design

Methods	Worst	Mean	Best	SD	FEs
GA4 [110]	NA	NA	81,843.3000	NA	225,000
ABC [112]	78,897.8100	81,496.0000	81,859.7416	0.6900	10,000
TLBO [112]	80,807.8551	81,438.9870	81,859.7400	0.6600	10,000
MBA [113]	84,440.1948	85,321.4030	85,535.9611	211.5200	15,100
AEO	85,516.5915643	85,541.5549651	85,549.0559072	10.6305	10,000

**Table 29** Comparisons of best solutions offered by reported optimizers for rolling element bearing design

	GA4	TLBO	MBA	AEO
$x_1(D_m)$	125.7171	125.7191	125.7153	125.7189368
$x_2(D_b)$	21.423	21.42559	21.4223300	21.4255649
$x_3(Z)$	11	11	11.000	11.3951818
$x_4(f_i)$	0.515	0.515	0.515000	0.5150000
$x_5(f_o)$	0.515	0.515	0.515000	0.5150009
$x_6(K_{D\min})$	0.4159	0.424266	0.488805	0.4102007
$x_7(K_{D\max})$	0.651	0.633948	0.627829	0.6384093
$x_8(\varepsilon)$	0.300043	0.3	0.300149	0.3000036
$x_9(e)$	0.0223	0.068858	0.097305	0.0469690
$x_{10}(\zeta)$	0.751	0.799498	0.646095	0.6095004
$g_1$	0.000821	0	0	1.0278E–09
$g_2$	13.732999	13.15257	8.630183	14.1370792
$g_3$	2.724000	1.5252	1.101429	1.8375246
$g_4$	3.606000	0.719056	2.040448	12.4611902
$g_5$	0.717000	16.49544	0.715366	11.0233166
$g_6$	4.857899	0	23.611002	0.7189368
$g_7$	0.003050	0	0.000480	4.0017E–07
$g_8$	– 0.000007	– 2.559363	0	– 3.1405502
$g_9$	0.000007	0	0	9.7829E–10
$g_{10}$	0.000005	0	0	9.0920E–07
$f_7$	81,843.3	81,859.74	85,535.9611	85,549.0559072

results on a range of unconstrained benchmark functions reveal the superiority of AEO in terms of exploration, exploitation, and exploration–exploitation trade-off. On the other hand, the comparisons on eight classical constrained engineering problems demonstrate that AEO may offer a faster convergence rate and high-quality solutions with less computational efforts.

## 5 Applications to identification of hydrogeological parameters

### 5.1 Confined aquifer

For a confined aquifer, the transmissibility is the seepage flow under the conditions of unit hydraulic gradient, unit time, and unit width, which signifies a capability of aquifer's water conductivity. The storage coefficient is

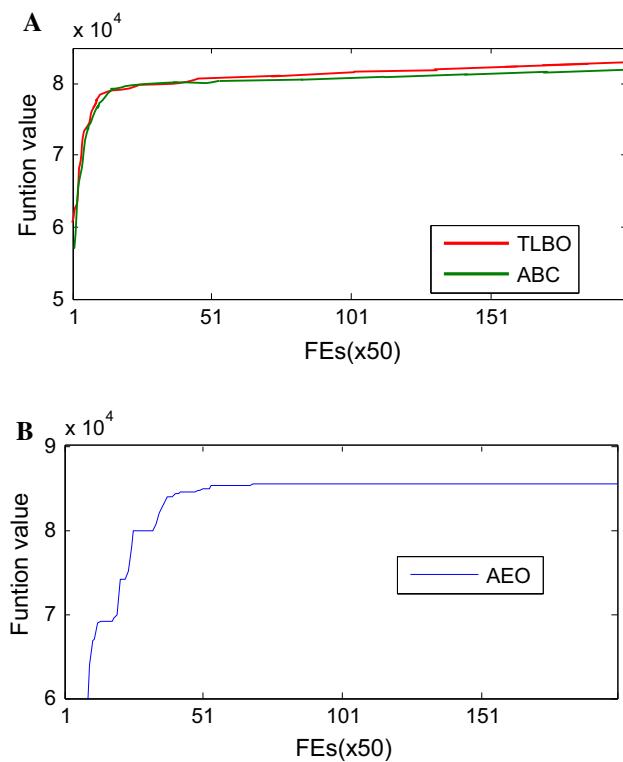
generally the water volume that is stored or released from unit area aquifer when the waterhead has a unit change, which signifies the water storage capacity. These hydrogeological parameters are calculated according to the following Theis formula [116]

$$s = \frac{Q}{4\pi T} W(u) \quad (27)$$

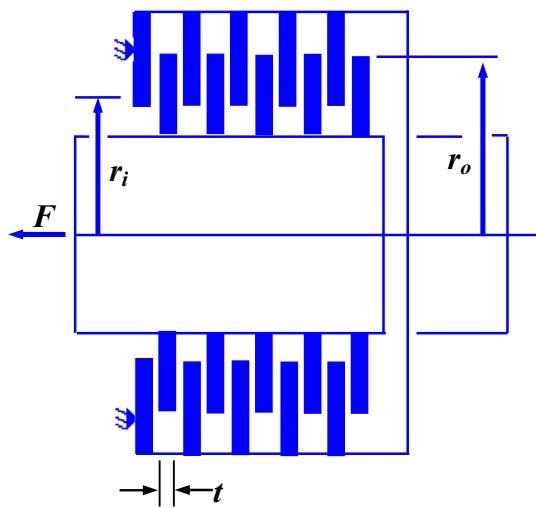
$$u = \frac{r^2 \mu^*}{4Tt} \quad (28)$$

where  $s$ : drawdown in an observation well,  $t$ : time since pumping has started,  $T$ : transmissibility of the aquifer,  $Q$ : pumping flow rate,  $r$ : distance from the pumping well to the observation well,  $W$ : well function, and  $\mu^*$ : storage coefficient of the aquifer.

By rearranging Eqs. (27) and (28) and taking logarithms of both sides, the following relationships are obtained



**Fig. 26** Convergence curves of algorithms for rolling element bearing problem, **a** TLBO and ABC, **b** AEO



**Fig. 27** Multiple disk clutch brake design

**Table 30** Comparisons of statistical results using reported optimizers in the literature for multiple disk clutch brake design

Methods	Worst	Mean	Best	SD	FES
ABC [112]	0.352864	0.324751	0.313657	0.54	600
TLBO [112]	0.392071	0.3271662	0.313657	0.67	600
AEO	0.3332601	0.3216842	0.3136566	0.008154	500

$$\lg s = \lg W(u) + \lg \frac{Q}{4\pi T} \quad (29)$$

$$\lg \frac{t}{r^2} = \lg \frac{1}{u} + \lg \frac{\mu^*}{4T} \quad (30)$$

If the values of  $W(u)$  and  $1/u$  are shown on a log–log paper, then a type curve for the relationship between  $s$  and  $t/r^2$  is produced. The values of  $s$  versus  $t/r^2$  can be shown on a transparent log–log paper using the same scale as the type curve, and its curve may be similar to the type curve, but would be displaced by the term  $Q/(4\pi T)$  on the  $s$  and  $W(u)$  axes, and by the term  $\mu^*/(4T)$  on the  $t/r^2$  and  $1/u$  axes. The plot of  $s$  versus  $t/r^2$  could be carefully slid on the type curve, keeping the axes parallel, until it matches a segment of the type curve. When the match is finished, the match point is any intersecting line set on the overlay curve, and choosing different match points will produce the similar results, so it is frequently chosen  $W(u) = 1$  and  $1/u = 10$  in  $W(u) - 1/u$  coordinate system, and the values of  $s$  and  $t/r^2$  are read in  $s - t/r^2$  coordinate system. Figure 29 shows the curve fitting procedure; the blue plot is  $W(u) - 1/u$  coordinate including the type curve and the red is  $s - t/r^2$  coordinate, the curve fitting procedure is the motion of  $s - t/r^2$  coordinate relative to  $W(u) - 1/u$  coordinate horizontally and vertically. When the match is finished, the match point can be obtained. Then, the transmissivity and storage coefficient may be calculated by

$$T = \frac{Q}{4\pi s} W(u) \quad (31)$$

$$\mu^* = \frac{4Ttu}{r^2} \quad (32)$$

The objective function which can well measure the fitting error between  $W(u) - 1/u$  coordinate system and  $s - t/r^2$  coordinate system is established for the confined aquifer, and then, the following objective function is considered to determine the optimal parameters

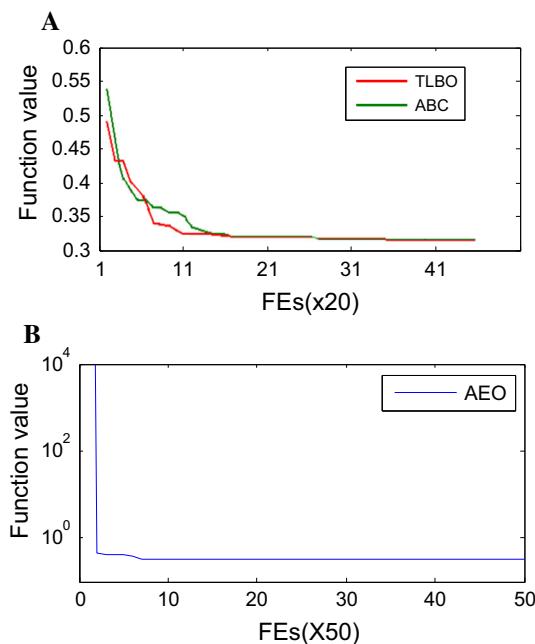
$$\min f_1(x, y) = \frac{\sum_{i=1}^N \left( \log_{10}(s_{obs,i} \cdot 10^y) - \log_{10} W\left(\frac{1}{t_i \cdot 10^x}\right) \right)^2}{\sum_{i=1}^N \left( \log_{10} W\left(\frac{1}{t_i \cdot 10^x}\right) - \frac{1}{N} \sum_{i=1}^N \log_{10} W\left(\frac{1}{t_i \cdot 10^x}\right) \right)^2} \quad (33)$$

$$W(u) = \begin{cases} -\ln u + \sum_{i=0}^5 a_i u^i & u \leq 1 \\ \frac{1}{ue^u} \sum_{i=0}^4 b_i u^i & u \geq 1 \end{cases} \quad (34)$$

$$\text{subject to } \begin{cases} -\log(t_1) \leq x \leq 4 - \log(t_N) \\ -1 - \log(s_1) \leq y \leq 1 - \log(s_N) \end{cases} \quad (35)$$

**Table 31** Comparisons of best solutions offered by reported optimizers for multiple disk clutch brake design

	NSGA-II [115]	TLBO [112]	ABC	PSO	CS	GSA	AEO
$x_1(r_i)$	70.0000	70	70	70	70	72	70
$x_2(r_o)$	90.0000	90	90	90	90	92	90
$x_3(t)$	15.0000	1	1	1	1	2	1
$x_4(F)$	1000.0000	810	790	860	810	815	810
$x_5(Z)$	3.0000	3	3	3	3	3	3
$g_1$	0.0000	0	0	0	0	0	0
$g_2$	22.0000	24.000000	24	24	24	22	24
$g_3$	0.9005	0.919427	0.9214172	0.9144542	0.9194278	0.9213930	0.9194278
$g_4$	9.7906	9830.371000	9.8345595	9.8199002	9.8303711	9.8304134	9.8303711
$g_5$	7.8947	7894.696500	7.8946966	7.8946966	7.8946966	7.8426028	7.8946966
$g_6$	3.3527	0.702013	0.3510866	1.5099273	0.7020132	1.0372174	0.7020132
$g_7$	60.6250	37,706.250000	35.2937500	43.7375000	37.7062500	40.1239024	37.7062500
$g_8$	11.6473	14.297986	14.6489134	13.4900727	14.2979868	13.9627826	14.2979868
$f_8$	0.4704	0.3136566	0.3136570	0.3136566	0.3136566	0.3175771	0.3136566

**Fig. 28** Convergence curves of algorithms for multiple disk clutch brake design problem, **a** TLBO and ABC, **b** AEO

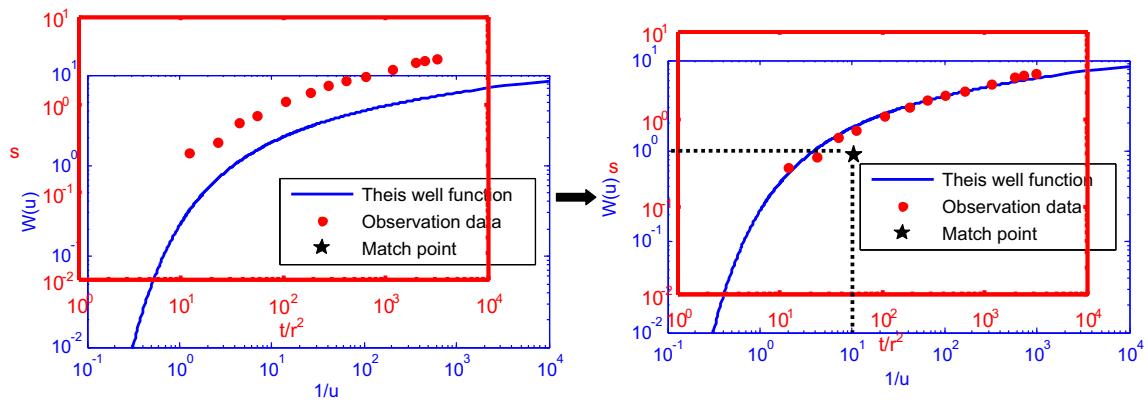
where the rounded-off values of the constants are given by  $a_0 = -0.57722$ ,  $a_1 = 0.99999$ ,  $a_2 = -0.24991$ ,  $a_3 = 0.05519$ ,  $a_4 = -0.00976$ ,  $a_5 = -0.00976$ ,  $a_6 = 0.00108$ ,  $b_0 = 0.26777$ ,  $b_1 = 8.63476$ ,  $b_2 = 18.05902$ ,  $b_3 = 8.57333$ ,  $b_4 = 1$ ,  $c_0 = 3.95850$ ,  $c_1 = 21.09965$ ,  $c_2 = 25.63296$ ,  $c_3 = 9.57332$ ,  $c_4 = 1$ .  $x$  and  $y$  are the displacements of  $s - t/r^2$  coordinate relative to  $W(u) - 1/u$  coordinate horizontally and vertically, respectively,  $s_{obs,i}$  is the observed drawdown at time  $i$ ,  $t_i (i = 1, \dots, N)$  is the  $i$ th

time from the beginning of discharge to observation,  $N$  is the size of the drawdown data,  $s_{obs,i}$  is the  $i$ th drawdown in an observation well, and  $x$  and  $y$ , respectively, signify the displacements of the  $s - t$  coordinate relative to the  $W(u) - 1/u$  coordinate both horizontally and vertically.

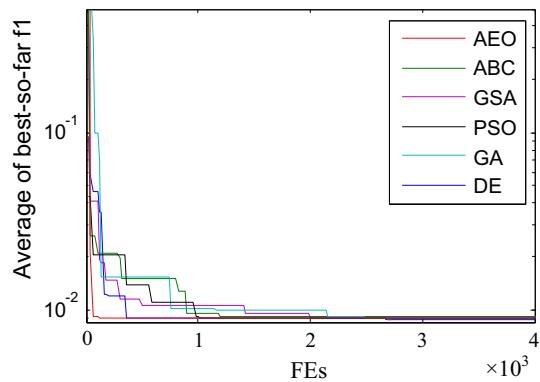
In this study, pumping test data of multiple wells in a confined aquifer are provided, and the observation data of multiple wells are summarized in Table 32, which is obtained from the data analysis [117]. The water inflow  $Q$  is  $60 \text{ m}^3/\text{h}$ . The distances between the pumping well and the observation wells 1, 2, 10, and 15, are, respectively, 780, 43, 510, and 140 m.

The above meta-heuristics and the proposed method are used to handle this problem, and these algorithms use the same parameters as the above section. The size of population and the maximal number of iterations are set to 20 and 200, respectively. Each algorithm performs 30 runs, and the results are based on the average performance of these runs. To fully compare the obtained results, two numerical methods, such as the Jacob linear graphic method and Theis type-curve method, are also applied to this problem.

The convergence comparisons of all the investigated optimizers for parameter identification of confined aquifer are shown in Fig. 30. From this figure, AEO is far superior to the other meta-heuristic methods with respect to convergence rate. Besides, the comparisons of optimization results offered by different optimizers are given in Table 33. As the table shows, AEO and ABC offer the same good results in terms of the ‘Mean’ and ‘Best’ indices, which are better than those of other optimizers, and the ‘SD’ index obtained by AEO is slightly better

**Fig. 29** Fitting curve procedure of confined aquifer**Table 32** Pumping test data of a confined aquifer

Observation well	Distance $r$ (m)	Time $t$ (min)	Drawdown $s$ (m)	Observation well	Distance $r$ (m)	Time $t$ (min)	Drawdown $s$ (m)
1	780	210	0.16	15	140	20	0.48
		330	0.34			100	1.12
		400	0.42			210	1.55
		645	0.71			400	1.89
		870	0.87			870	2.38
		1185	1.06			1185	2.54
2	43	10	0.74	10	510	100	0.20
		20	1.28			210	0.40
		60	1.96			400	0.65
		100	2.28			870	1.14
		210	2.77			1185	1.35
		400	3.2				
		870	3.68				
		1185	3.85				

**Fig. 30** Convergence comparisons of investigated optimizers for parameter identification of confined aquifer

than that obtained by ABC. The solutions, match points, hydrogeological parameters, and fitting errors offered by the reported optimizers are given in Table 34. The hydrogeological parameters obtained by different

considered methods can be used to plot their curve fitting charts between the Theis well function and the observation data, which are depicted in Fig. 31. Visually, the fitting effect of AEO is very satisfactory. Nevertheless, the fitting effect of the Jacob linear graphic method is barely satisfactory.

## 5.2 Leaky aquifer

A leaky aquifer is essentially a confined aquifer except that its ceiling and/or bottom are semipervious so that water may leak into or out of the aquifer. So a hydrogeological parameter representing the leakage characteristic of aquitard, called the leakage factor, is added to the well function. The hydrogeological parameters of a leaky aquifer can be obtained by the following Hantush–Jacob formula [118–121]

**Table 33** Performance comparisons of algorithms for parameter identification of confined aquifer

Index	AEO	ABC	GSA	PSO	GA	DE
Mean	7.4152E–03	7.4152E–03	7.7132E–03	7.8750E–03	8.0304E–03	7.4154E–03
SD	3.2742E–18	2.1581E–10	2.4581E–04	4.3272E–06	1.3272E–05	2.1581E–10
Best	7.4152E–03	7.4152E–03	7.4201E–03	7.4381E–03	7.5156E–03	7.4152E–03

**Table 34** Result comparisons of different methods for parameter identification of confined aquifer

Method	Solution ( $x, y$ )	Match point ( $W, 1/u, s, t/r^2$ )	Hydrogeological parameters ( $T, \mu^*$ )	Fitting error ( $f_i$ )
AEO	(3.5963, 0.2975)	(1, 10, 0.5041, 2.5333E–03)	(2.2733E+02, 1.5997E–04)	7.4152E–03
ABC	(3.5963, 0.2975)	(1, 10, 0.5041, 2.5333E–03)	(2.2733E+02, 1.5997E–04)	7.4152E–03
GSA	(3.5973, 0.2914)	(1, 10, 0.5112, 2.5278E–03)	(2.2418E+02, 1.5714E–04)	7.7132E–03
PSO	(3.6161, 0.3117)	(1, 10, 0.4879, 2.4204E–03)	(2.3486E+02, 1.5791E–04)	7.8750E–03
GA	(3.5836, 0.2994)	(1, 10, 0.5019, 2.6086E–03)	(2.2834E+02, 1.6544E–04)	8.0304E–03
DE	(3.5967, 0.2978)	(1, 10, 0.5037, 2.5311E–03)	(2.2751E+02, 1.5995E–04)	7.4154E–03
Type-curve [117]	–	(1, 10, 0.54, 2.5E–03)	(2.123E+02, 1.47E–04)	1.4985E–02
Jacob [117]	–	–	(1.94E+02, 2.78E–04)	0.1332

$$\lg s = \lg W(u, z) + \lg \frac{Q}{4\pi T} \quad (36)$$

$$\lg t = \lg \frac{1}{u} + \lg \frac{r^2 \mu^*}{4T} \quad (37)$$

where  $B$  is the leakage factor and  $z = r/B$ .

The curve fitting procedure of a leaky aquifer is similar to that of a confined aquifer, and the difference between them is that there is a leakage factor determining the shape of the well function, which makes it difficult to find accurate parameters of a leaky aquifer. Figure 32 depicts the curve fitting procedure of a leaky aquifer. After the match point is determined, the curve shape of the Hantush well function is obtained, and then the storage coefficient, the leaky factor, and the transmissivity are obtained by

$$T = \frac{Q}{4\pi s} W(u, z) \quad (38)$$

$$\mu^* = \frac{4Ttu}{r^2} \quad (39)$$

$$B = \frac{r}{z} \quad (40)$$

Therefore, the objective function which can well measure the fitting error between  $W(u, z) - 1/u$  coordinate system and  $s - t$  coordinate system is established for a leaky aquifer, and then the following objective function is considered to determine the optimal parameters [52, 118]

$$\min f_2(x, y, z) = \frac{\sum_{i=1}^N \left( \log_{10}(s_{\text{obs},i} \cdot 10^y) - \log_{10} W\left(\frac{1}{t_i \cdot 10^x}, z\right) \right)^2}{\sum_{i=1}^N \left( \log_{10} W\left(\frac{1}{t_i \cdot 10^x}, z\right) - \frac{1}{N} \sum_{i=1}^N \log_{10} W\left(\frac{1}{t_i \cdot 10^x}, z\right) \right)^2} \quad (41)$$

$$W\left(u, \frac{r}{B}\right) = \begin{cases} W\left(0, \frac{r}{B}\right) - W\left(\frac{r^2}{4B^2}u, 0\right) & u < u_{\min} \\ 0.5 \cdot W(0, u) \cdot \text{erfc}(zX + \beta X^3) & u_{\min} \leq u \leq u_{\max} \\ W(u, 0) & u > u_{\max} \end{cases} \quad (42)$$

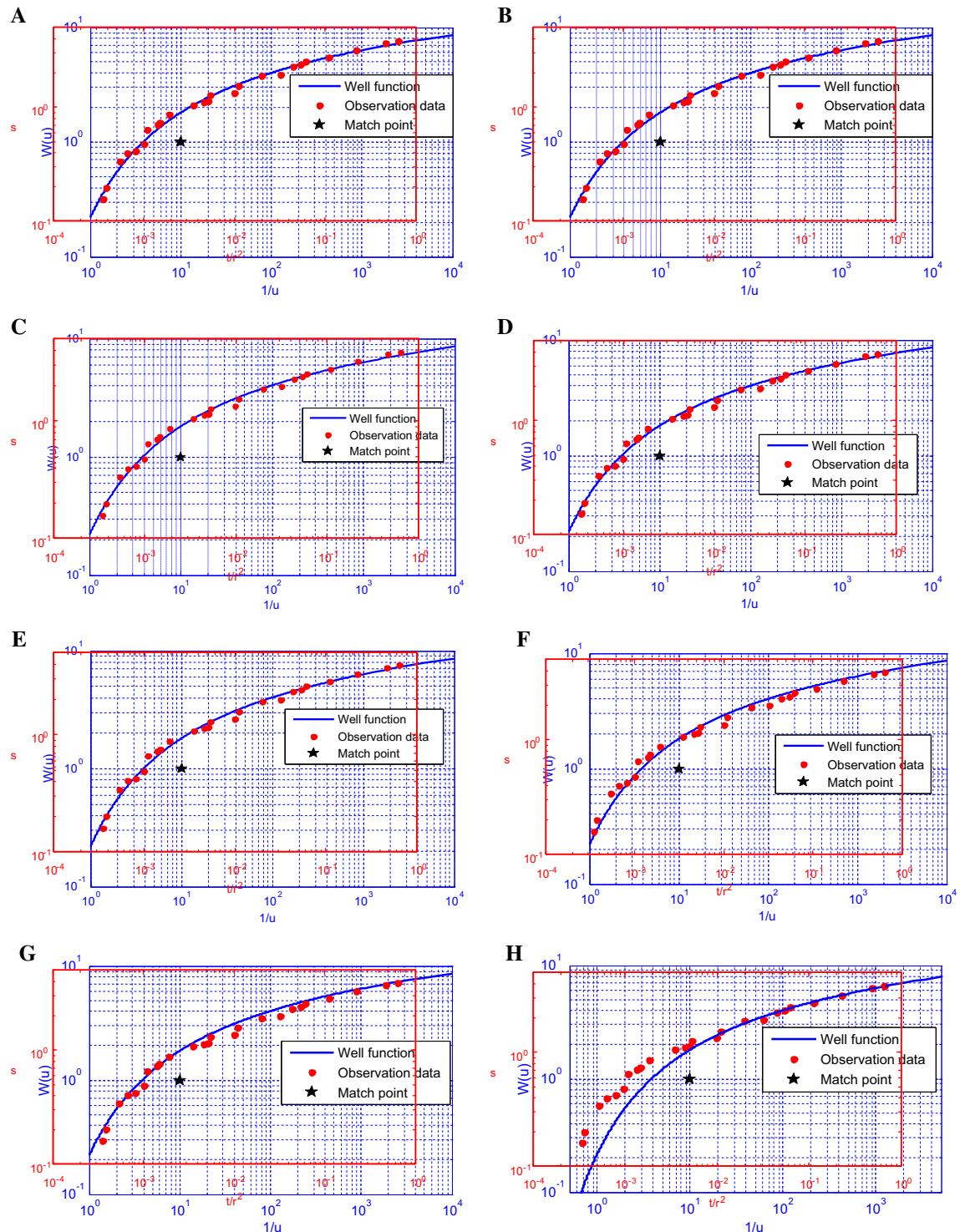
$$\begin{cases} X = \lg \frac{2uB}{r} \\ z = 0.7708 + 0.3457 \lg\left(\frac{r}{B}\right) + 0.09128 \lg^2\left(\frac{r}{B}\right) + 0.09937 \lg^3\left(\frac{r}{B}\right) \\ \beta = 0.02796 + 0.01023 \lg\left(\frac{r}{B}\right) \\ u_{\min} = \begin{cases} \max(0.06541v^{0.2763}, 0.02985) & \frac{r}{B} < 0.5 \\ 0.1192v^{1.142} & \frac{r}{B} \geq 0.5 \end{cases} \\ u_{\max} = \max(39.93v^{2.391}, 0.02985) \end{cases} \quad (43)$$

$$W(0, v) = \begin{cases} (1 + 0.2062v^2) \cdot \lg(-2\lg(0.5772)/v)^2 + 0.5579v^2 & v < 1 \\ 1/(v \cdot \exp(v) \cdot \sqrt{2\pi/v} \cdot (v - 0.09173)) & v \geq 1 \end{cases} \quad (44)$$

$$W(u, 0) = \begin{cases} -\ln u + \sum_{i=0}^5 a_i u^i & u \leq 1 \\ \frac{1}{ue^u} \sum_{i=0}^4 b_i u^i & u \geq 1 \end{cases} \quad (45)$$

$$\text{Subject to } \begin{cases} -1 - \log_{10}(t_1) \leq x \leq 4 - \log_{10}(t_N) \\ -2 - \log_{10}(s_1) \leq y \leq 1 - \log_{10}(s_N) \\ 0 \leq z \leq 10 \end{cases} \quad (46)$$

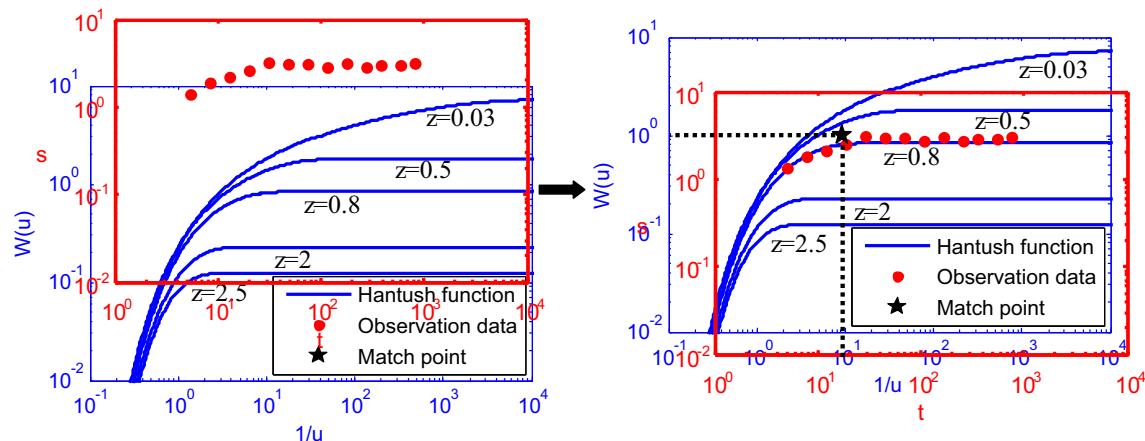
where  $x$  and  $y$  signify the horizontal and vertical displacements of  $s - t$  relative to  $W(u, z) - 1/u$ , respectively.  $z$  signifies the curve shape of the Hantush well function.



**Fig. 31** Fitting curves between Theis well function and observation data obtained by different methods, **a** AEO, **b** ABC, **c** GSA, **d** PSO, **e** GA, **f** DE, **g** Theis type-curve method, **h** Jacob linear graphic method

In this study, pumping test data in a leaky aquifer are provided in Table 35, which is selected for data analyses [117]. This problem is solved by the reported algorithms whose parameters are as same as those of the algorithms

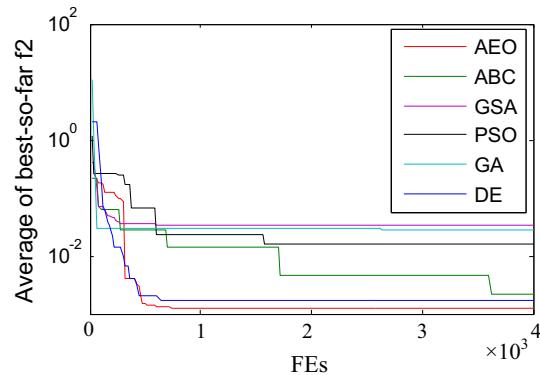
used to the previous problem, and two numerical methods including the Hantush–Jacob type-curve method and inflected point method are also used to solve this problem. The convergence comparisons of investigated optimizers



**Fig. 32** Curve fitting procedure of leaky aquifer

**Table 35** Pumping test data of a leaky aquifer

Time (min)	Drawdown (m)	Time (min)	Drawdown (m)	Time (min)	Drawdown (m)
1	0.05	75	0.62	360	0.772
4	0.054	90	0.64	390	0.785
7	0.12	120	0.685	420	0.79
10	0.175	150	0.725	450	0.792
15	0.26	180	0.735	480	0.794
20	0.33	210	0.755	510	0.795
25	0.383	240	0.76	540	0.796
30	0.425	270	0.76	570	0.797
45	0.52	300	0.763	600	0.798
60	0.575	330	0.77	660	0.80



**Fig. 33** Convergence comparisons of investigated optimizers for parameter identification of leaky aquifer

are depicted in Fig. 33 when solving this problem. It can be noted that AEO shows a better convergence behavior than the other meta-heuristic methods. Meanwhile, the performance comparison of different optimizers in Table 36

suggests that AEO performs the best of all the considered meta-heuristic methods in terms of ‘Mean,’ ‘SD,’ and ‘Best’ indices. The solutions, match points, hydrogeological parameters, and fitting errors offered by the reported optimizers are illustrated in Table 37; note that some considered methods and AEO are close in fitting errors but differ greatly in the values of parameters.

The curve fitting charts are plotted in Fig. 34 based on the hydrological parameters obtained by these considered methods. As seen in this figure, visually, the fitting effect of AEO is not inferior to those of the other methods. However, the fitting effects of both the Hantush–Jacob type-curve method and inflected point method are rather poor; beyond this, GSA, PSO and GA do not fit well. These overall comparisons may highly support that the proposed algorithm can effectively and efficiently find appropriate hydrological parameters in different hydrological models. These results of this section also suggest the applicability

**Table 36** Performance comparisons of algorithms for parameter identification of a leaky aquifer

Index	AEO	ABC	GSA	PSO	GA	DE
Mean	1.2148E–03	3.6328E–03	3.3434E–02	1.6187E–02	2.8540E–02	1.5482E–03
SD	4.3291E–08	1.2516E–03	6.8321E–04	4.2151E–04	1.2516E–03	2.4581E–05
Best	1.2148E–03	7.5325E–03	5.2189E–03	1.4291E–03	7.5325E–03	1.6215E–03

**Table 37** Result comparisons of different methods for parameter identification of a leaky aquifer

Method	Solution ( $x, y, z$ )	Match point ( $W, 1/u, s, t$ )	Hydrogeological parameters ( $T, \mu^*, B$ )	Fitting error ( $f_2$ )
AEO	(–0.7703, 0.4111, 0.4348)	(1, 10, 0.3881, 58.9194)	(340.04765, 1.4340E–04, 453.1)	1.2148E–03
ABC	(–0.7842, 0.3658, 0.5006)	(1, 10, 0.2541, 39.4647)	(519.3630, 1.4670E–04, 393.5)	3.6328E–03
GSA	(–0.74514, 0.4083, 0.3849)	(1, 10, 0.3906, 55.6089)	(337.87857, 1.3448E–04, 511.8)	3.3434E–02
PSO	(–0.4997, 0.6615, 0.1470)	(1, 10, 0.2411, 36.9255)	(605.3102, 1.3693E–04, 1340.1)	1.6187E–02
GA	(–0.5305, 0.6912, 0.1140)	(1, 10, 0.2036, 33.9214)	(648.1448, 1.5737E–04, 1728.1)	2.8540E–02
DE	(–0.7758, 0.4161, 0.4311)	(1, 10, 0.3836, 59.6794)	(344.0441, 1.9797E–04, 457.0)	1.5482E–03
Type-curve [117]	(–, –, 0.3500)	(2.45, 101, 0.78, 500)	(414.7, 1.47E–04, 562.9)	3.490E–02
Inflected point [117]	–	–	(397, 1.28E–04, 547)	5.9010E–02

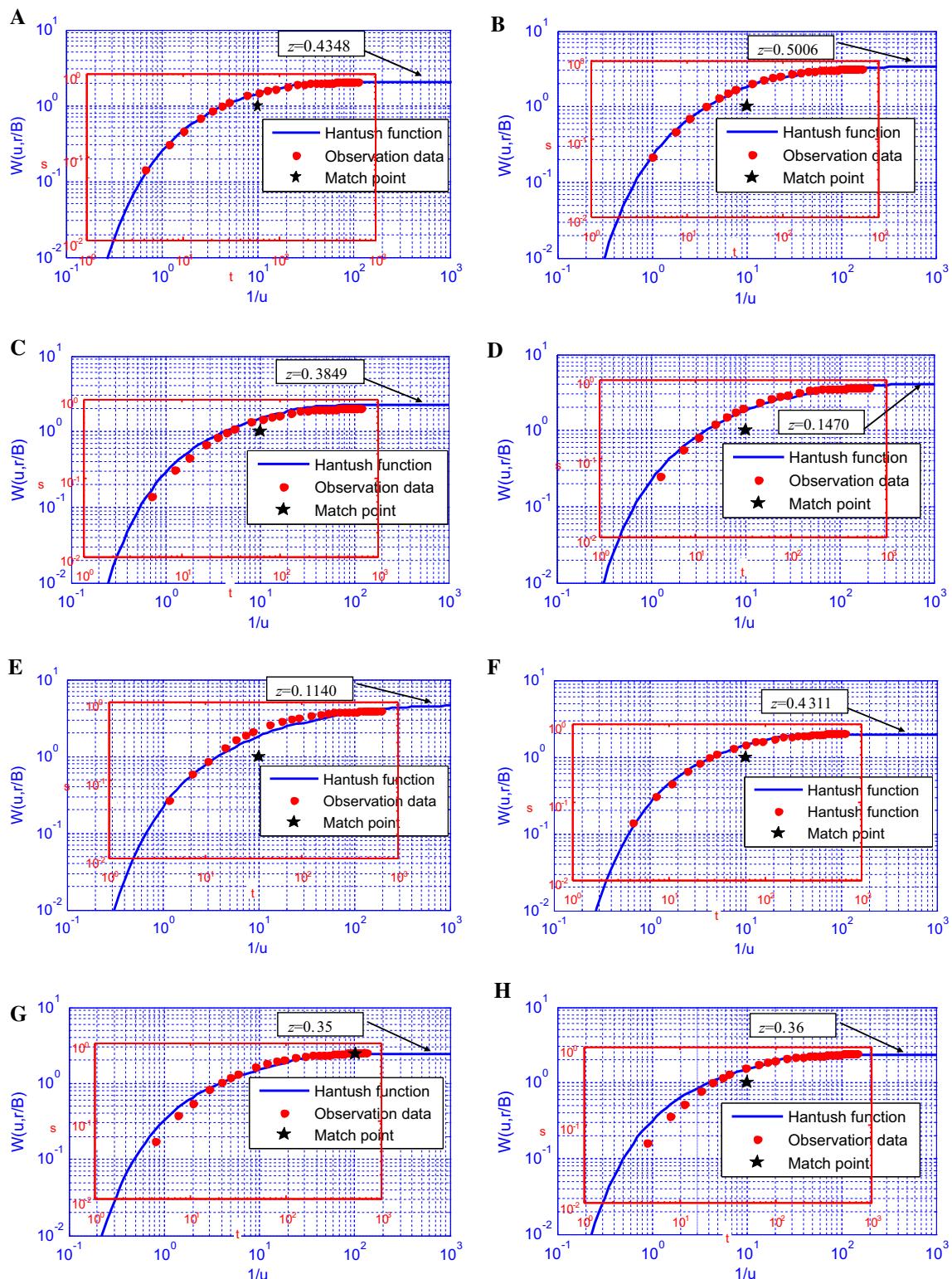
of our method in handing challenging real problems with unknown problem space.

## 6 Conclusions

This work proposed a novel swarm-based optimizer named artificial ecosystem-based optimization (AEO). Its concept and idea underlining this algorithm are driven from natural ecosystem on the earth. The proposed algorithm includes three behaviors to mimic production, consumption, and decomposition of living organisms. This approach, with few adjustable parameters, is easy to implement. An overall test is implemented on 31 benchmark functions to analysis exploration, exploitation, convergence behavior, avoidance of local optima, comprehensive significance, and quantitative rank of the presented optimizer. The comparisons of statistical results are very convincing that AEO is highly competitive with other meta-heuristics. To further test its ability to handle real-world problems, an extensive study is conducted on eight real-world engineering problems such as pressure vessel design and cantilever beam design. The statistical results with different indices suggest that, compared with other reported popular methods, AEO is more promising to tackle real-world

problems requiring a faster convergence rate and less computational expense with a specified precision for final solutions. Finally, the applications of AEO to the field of identification of hydrogeological parameters are investigated in this study. It is difficult to solve these employed problems because of their particularity and unknown search space. But AEO can manage to identify hydrogeological parameters of aquifers effectively, proving that the proposed algorithm is highly promising in different engineering fields.

Generally speaking, AEO provides enough competitive results compared to other considered meta-heuristic methods on the reported results in the literature and the experimental results in this study. However, the computational efficiency and solution quality depend largely on presentation and complexity of underlined problems. This is a general characteristic not just for AEO, but for most meta-heuristics. Therefore, our algorithm tends to handle the real-world problems focusing on convergence efficiency and computational expense with an acceptable precision for final solutions. For future work, the binary AEO may be presented to deal with complex discrete problems. AEO could also be tailored to address multiobjective optimization.



**Fig. 34** Fitting curves between well function and observation data obtained by different methods, **a** AEO, **b** ABC, **c** GSA, **d** PSO, **e** GA, **f** DE, **g** Hantush–Jacob type-curve method, **h** inflected point method

**Acknowledgements** This work was supported in part by Natural Science Foundation of Hebei Province of China (E2018402092, F2017402142) and Scientific Research Key Project of University of Hebei Province of China (ZD2017017).

## Compliance with ethical standards

**Conflict of interest** The authors declared that they have no conflict of interest to this work.

## Appendix: Constrained engineering problems

### Three-bar truss design

Consider variable  $\vec{x} = [x_1, x_2]$ .

$$\text{Minimize } f_1(\vec{x}) = (2\sqrt{2}x_2 + x_2) \times l$$

$$\text{subject to } g_1(\vec{x}) = \frac{\sqrt{2}x_1+x_2}{\sqrt{2}x_1^2+2x_2x_1}P - \sigma \leq 0, \quad g_2(\vec{x}) = \frac{x_2}{\sqrt{2}x_1^2+2x_2x_1}P - \sigma \leq 0 \text{ where } l = 10 \text{ cm, } P = 2 \text{ kN/cm}^2, \text{ and } \sigma = 2 \text{ kN/cm}^2.$$

Variable's range  $0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1$ .

### Cantilever beam design

Consider variable  $\vec{x} = [x_1, x_2]$ .

$$\text{Minimize } f_2(\vec{x}) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5)$$

$$\text{subject to } g_1(\vec{x}) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0.$$

Variable's range  $0.01 \leq x_i \leq 100, i = 1, \dots, 5$ .

### Tension/compression spring design

Consider variable  $\vec{x} = [x_1, x_2, x_3] = [d, D, N]$ .

$$\text{Minimize } f_3(\vec{x}) = (x_3 + 2)x_2x_1^2$$

$$\text{subject to } g_1(\vec{x}) = 1 - \frac{x_3x_2^3}{71,785x_1^4} \leq 0, \quad g_2(\vec{x}) = \frac{4x_2^2 - x_1x_2}{12,566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0, \quad g_3(\vec{x}) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0, \quad g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0.$$

Variable's range  $0.05 \leq x_1 \leq 2, 0.25 \leq x_2 \leq 1.3, 2 \leq x_3 \leq 15$ .

### Pressure vessel design

Consider variable  $\vec{x} = [x_1, x_2, x_3, x_4] = [T_s, T_h, R, L]$ .

$$\text{Minimize } f_4(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 +$$

$$3.1661x_1^2x_4 + 19.84x_1^2x_3$$

$$\text{subject to } g_1(\vec{x}) = -x_1 + 0.0193x_3 \leq 0, \quad g_2(\vec{x}) = -x_2 + 0.00954x_3 \leq 0, \quad g_3(\vec{x}) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1,296,000 \leq 0, \quad g_4(\vec{x}) = x_4 - 240 \leq 0.$$

Variable's range  $0 \leq x_1 \leq 99, 0 \leq x_2 \leq 99, 10 \leq x_3 \leq 200, 10 \leq x_4 \leq 200$ .

## Welded beam design

Consider variable  $\vec{x} = [x_1, x_2, x_3, x_4] = [h, l, t, b]$ .

$$\text{Minimize } f_5(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2)$$

$$\text{subject to } g_1(\vec{x}) = \tau(\vec{x}) + \tau_{\max} \leq 0, \quad g_2(\vec{x}) = \sigma(\vec{x}) + \sigma_{\max} \leq 0, \quad g_3(\vec{x}) = \delta(\vec{x}) + \delta_{\max} \leq 0, \quad g_4(\vec{x}) = x_1 - x_4 \leq 0, \quad g_5(\vec{x}) = P - P_c(\vec{x}) \leq 0, \quad g_6(\vec{x}) = 0.125 - x_1 \leq 0, \quad g_7(\vec{x}) = 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \leq 0 \text{ where}$$

$$\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau'' \frac{x_2}{2R} + (\tau'')^2}, \quad \tau' = \frac{P}{\sqrt{2}x_1x_2}, \quad \tau'' = \frac{MR}{J},$$

$$M = P(L + \frac{x_2}{2}), \quad R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1 + x_3}{2})^2},$$

$$J = 2 \left\{ \sqrt{2}x_1x_2 \left[ \frac{x_2^3}{4} + (\frac{x_1 + x_3}{2})^2 \right] \right\}, \quad \sigma(\vec{x}) = \frac{6PL}{x_4x_3^2},$$

$$\delta(\vec{x}) = \frac{4PL^3}{Ex_4x_3^2}, \quad P_c(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3x_4}{36}}}{L^2} \left( 1 - \frac{x_3}{2L} \sqrt{\frac{E}{4G}} \right).$$

where  $P = 6000$  lb,  $L = 14$  in,  $E = 30 \times 10^6$  psi,  $G = 12 \times 10^6$  psi,  $\tau_{\max} = 13,600$  psi,  $\sigma_{\max} = 30,000$  psi, and  $\delta_{\max} = 0.25$  in.

Variables' range  $0.1 \leq x_1 \leq 2, 0.1 \leq x_2 \leq 10, 0.1 \leq x_3 \leq 10$ , and  $0.1 \leq x_4 \leq 2$ .

### Speed reducer design

Consider variable  $\vec{x} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7] = [b, m, z, l_1, l_2, d_1, d_2]$ .

$$\text{Minimize } f_6(\vec{x}) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 0.7854x_1(x_4x_6^2 - x_5x_7^2)$$

$$\text{subject to } g_1(\vec{x}) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0, \quad g_2(\vec{x}) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0,$$

$$g_3(\vec{x}) = \frac{1.93x_4^2}{x_2x_6^4x_3} - 1 \leq 0, \quad g_4(\vec{x}) = \frac{1.93x_5^2}{x_2x_7^4x_3} - 1 \leq 0,$$

$$g_5(\vec{x}) = \frac{\left( \left( \frac{745x_4}{x_2x_3} \right)^2 + 16.9 \times 10^6 \right)^{0.5}}{110x_6^3} - 1 \leq 0,$$

$$g_6(\vec{x}) = \frac{\left( \left( \frac{745x_5}{x_2x_3} \right)^2 + 157.5 \times 10^6 \right)^{0.5}}{85x_7^3} - 1 \leq 0,$$

$$g_7(\vec{x}) = \frac{x_2x_3}{40} - 1 \leq 0, \quad g_8(\vec{x}) = \frac{5x_2}{x_1} - 1 \leq 0,$$

$$g_9(\vec{x}) = \frac{x_1}{12x_2} - 1 \leq 0, \quad g_{10}(\vec{x}) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0,$$

$$g_{11}(\vec{x}) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0.$$

Variables' range  $2.6 \leq x_1 \leq 3.6, 0.7 \leq x_2 \leq 0.8, 17 \leq x_3 \leq 28, 7.3 \leq x_4 \leq 8.3, 7.3 \leq x_5 \leq 8.3, 2.9 \leq x_6 \leq 3.9$ , and  $5.0 \leq x_7 \leq 5.5$ .

## Rolling element bearing design

Consider variable  $\vec{x} = [D_m, D_b, Z, f_i, f_o, K_{D\min}, K_{D\max}, \varepsilon, e, \zeta]$ .

$$\begin{aligned} & \text{Maximize } \begin{cases} f_7(\vec{x}) = f_c Z^{2/3} D_b^{1.8} & \text{if } D_b \leq 25.4 \text{ mm} \\ f_7(\vec{x}) = 3.647 f_c Z^{2/3} D_b^{1.4} & \text{if } D_b > 25.4 \text{ mm} \end{cases} \\ & \text{subject to } g_1(\vec{x}) = \frac{\phi_o}{2 \sin^{-1}(D_b/D_m)} - Z + 1 \geq 0, \quad g_2(\vec{x}) = 2D_b - K_{D\min}(D - d) \geq 0, \quad g_3(\vec{x}) = K_{D\max}(D - d) - 2D_b \geq 0, \quad g_4(\vec{x}) = D_m - (0.5 - e)(D + d) \geq 0, \quad g_5(\vec{x}) = (0.5 + e)(D + d) - D_m \geq 0, \quad g_6(\vec{x}) = D_m - 0.5(D + d) \geq 0, \\ & g_7(\vec{x}) = 0.5(D - D_m - D_b) - \varepsilon D_b \geq 0, \quad g_8(\vec{x}) = \zeta B_w - D_b \leq 0, \quad g_9(\vec{x}) = f_i \geq 0.515, \quad g_{10}(\vec{x}) = f_o \geq 0.515 \quad \text{where} \\ & f_c = 37.91 \left[ 1 + \left\{ 1.04 \left( \frac{1-\gamma}{1+\gamma} \right)^{1.72} \left( \frac{f_i(2f_o-1)}{f_o(2f_i-1)} \right)^{0.4} \right\}^{10/3} \right]^{-0.3} \\ & \left( \frac{\gamma^{0.3}(1-\gamma)^{1.39}}{f_o(1+\gamma)^3} \right) \left( \frac{2f_i}{2f_i-1} \right)^{0.41}, \gamma = \frac{D_b \cos \alpha}{D_m}, f_i = \frac{r_i}{D_b}, f_o = \frac{r_o}{D_b}, \phi_o = \\ & 2\pi - 2 \cos^{-1} \frac{\{(D-d)/2-3(T/4)\}^2 + \{D/2-(T/4)-D_b\}^2 - \{d/2+(T/4)\}^2}{2\{(D-d)/2-3(T/4)\}\{D/2-(T/4)-D_b\}}, \\ & T = D - d - 2D_b, \quad D = 160, \quad d = 90, \quad B_w = 30, \\ & r_i = r_o = 11.033. \end{aligned}$$

Variables' range  $0.5(D + d) \leq D_m \leq 0.6(D + d)$ ,  $0.15(D - d) \leq D_b \leq 0.45(D - d)$ ,  $4 \leq Z \leq 50$ ,  $0.515 \leq f_i \leq 0.6$ ,  $0.515 \leq f_o \leq 0.6$ ,  $0.4 \leq K_{D\min} \leq 0.5$ ,  $0.6 \leq K_{D\max} \leq 0.7$ ,  $0.3 \leq \varepsilon \leq 0.4$ ,  $0.02 \leq e \leq 0.1$ ,  $0.6 \leq \zeta \leq 0.85$ .

## Multiple disk clutch brake design

Consider variable  $\vec{x} = [r_i, r_o, t, F, Z]$ .

$$\begin{aligned} & \text{Minimize } f_8(\vec{x}) = \pi(r_o^2 - r_i^2)t(Z + 1)\rho \\ & \text{subject to } g_1(\vec{x}) = r_o - r_i - \Delta r \geq 0, g_2(\vec{x}) = l_{\max} - (Z + 1)(t + \delta) \geq 0, \quad g_3(\vec{x}) = p_{\max} - p_{rz} \geq 0, \quad g_4(\vec{x}) = p_{\max} \\ & v_{sr\max} - p_{rz}v_{sr} \geq 0, \quad g_5(\vec{x}) = v_{sr\max} - v_{sr} \geq 0, g_6(\vec{x}) = T_{\max} - T \geq 0, \quad g_7(\vec{x}) = M_h - sM_s \geq 0, g_8(\vec{x}) = T \geq 0 \quad \text{where } M_h = \\ & \frac{2}{3}\mu F Z \frac{r_o^3 - r_i^3}{r_o^2 - r_i^2}, \quad p_{rz} = \frac{F}{\pi(r_o^2 - r_i^2)}, \quad v_{sr} = \frac{2\pi n(r_o^3 - r_i^3)}{90(r_o^2 - r_i^2)}, \quad T = \frac{I_z \pi n}{30(M_h + M_f)}, \\ & \Delta r = 20 \text{ mm}, \quad l_{\max} = 30 \text{ mm}, \quad v_{sr\max} = 10 \text{ m/s}, \quad \delta = 0.5, \\ & s = 1.5, \quad M_s = 40 \text{ Nm}, \quad M_f = 3 \text{ Nm}, \quad n = 250 \text{ rpm}, \quad p_{\max} = 1 \text{ MPa}, \quad I_z = 55 \text{ kg mm}^2, \quad T_{\max} = 15 \text{ s}, \quad F_{\max} = 1000 \text{ N}, \\ & r_{i\min} = 60 \text{ mm}, \quad r_{i\max} = 80 \text{ mm}, \quad r_{o\min} = 90 \text{ mm}, \quad r_{o\max} = 110 \text{ mm}, \quad t_{\min} = 1 \text{ mm}, \quad t_{\max} = 3 \text{ mm}, \quad F_{\min} = 600, \quad F_{\max} = 1100, \quad Z_{\min} = 2, \quad \text{and } Z_{\max} = 9. \end{aligned}$$

## References

- Adeli H, Cheng NT (1993) Integrated genetic algorithm for optimization of space structures. *J Aerosp Eng ASCE* 6(4):315–328
- Wang L, Zhao W, Tian Y, Pan G (2018) A bare bones bacterial foraging optimization algorithm. *Cognit Syst Res* 52:301–311
- Talbi EG (2009) Metaheuristics: from design to implementation, vol 74. Wiley, New York
- Holland JH (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor
- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of sixth international symposium on micro machine and human science (SMMHS-1995), pp 39–43
- Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. *IEEE Trans Syst Man Cybern Part B* 26(1):29–41
- Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220(4598):671–680
- Storn R, Price K (1997) Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 11:341–359
- Milner S, Davis C, Zhang H, Llorca J (2012) Nature-inspired self-organization, control, and optimization in heterogeneous wireless networks. *IEEE Trans Mob Comput* 11(7):1207–1222
- Ayman AA (2011) Pid parameters optimization using genetic algorithm technique for electrohydraulic servo control system. *Intell Control Autom* 2:888–896
- Hamidreza RK, Karim F (2008) An improved feature selection method based on ant colony optimization (ACO) evaluated on face recognition system. *Appl Math Comput* 205:716–725
- Zhang H, Cao X, Ho JK, Chow TW (2016) Object-level video advertising: an optimization framework. *IEEE Trans Ind Inf* 13(2):520–531
- Lai C, Shao Q, Chen X, Wang Z (2016) Flood risk zoning using a rule mining based on ant colony algorithm. *J Hydrol* 542:268–280
- Tarek H, Mohamed S, Moustafa K (2011) Incorporating rework into construction schedule analysis. *Autom Constr* 20:1051–1059
- Nayak B, Misra B, Choudhury TR (2018) Meta-heuristic optimization algorithms for design of gain constrained state variable filter. *Int J Electron Commun (AEÜ)* 93:7–18
- Hare W, Nutini J, Tesfamariam S (2013) A survey of non-gradient optimization methods in structural engineering. *Adv Eng Softw* 59:19–28
- Mühlenbein H, Gorges-Schleuter M, Krämer O (1988) Evolution algorithms in combinatorial optimization. *Parallel Comput* 7(1):65–85
- Geem ZW, Kim J, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. *Trans Simul* 76(2):60–68
- Krause J, Cordeiro J, Parpinelli RS, Lopes HS (2013) A survey of swarm algorithms applied to discrete optimization problems. In: Yang X-S, Cui Z, Xiao R, Gandomi AH, Karamanoglu M (eds) Swarm intelligence and bio-inspired computation: theory and applications. Elsevier Science & Technology Books, Elsevier, London, pp 169–191
- Passino KM (2002) Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Syst* 22(3):52–67
- De Falco I, Della Cioppa A, Maistro D, Scafuri U, Tarantino E (2012) Biological invasion-inspired migration in distributed evolutionary algorithms. *Inf Sci* 207:50–65
- Beyer HG, Schwefel HP (2002) Evolution strategies: a comprehensive introduction. *Nat Comput* 1(1):3–52
- Vikhar P (2016) Evolutionary algorithm: a classical search and optimization technique. *Int J Pure Appl Res Eng Technol* 4(9):758–766
- Corno F, Reorda MS, Squillero G (1998) A new evolutionary algorithm inspired by the selfish gene theory. In: IEEE international conference on evolutionary computation proceedings.

- IEEE world congress on computational intelligence, vol 1976. IEEE, pp 575–580
25. Eusuff MM, Lansey KE (2003) Optimizing of water distribution network design using the shuffled frog leaping algorithm. *J Water Resour Plan Manag* 129(3):210–225
  26. Simon D (2009) Biogeography-based optimization. *IEEE Trans Evolut Comput* 12(6):702–713
  27. Uymaz SA, Tezel G, Yel E (2015) Artificial algae algorithm (AAA) for nonlinear global optimization. *Appl Soft Comput* 31:153–171
  28. Civicioglu P (2013) Backtracking search optimization algorithm for numerical optimization problems. *Appl Math Comput* 219(15):8121–8144
  29. Pan WT (2012) A new fruit fly optimization algorithm: taking the financial distress model as an example. *Knowl Based Syst* 26:69–74
  30. Akay B, Karaboga D (2012) A modified artificial bee colony algorithm for real-parameter optimization. *Inf Sci* 192:120–142
  31. Karaboga D, Akay B (2009) A comparative study of artificial bee colony algorithm. *Appl Math Comput* 214(1):108–132
  32. Oftadeh R, Mahjoob MJ, Sharatiapanahi M (2010) A novel meta-heuristic optimization algorithm inspired by group hunting of animals: hunting search. *Comput Math Appl* 60(7):2087–2098
  33. Kiran MS (2015) TSA: tree-seed algorithm for continuous optimization. *Expert Syst Appl* 42(19):6686–6698
  34. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
  35. Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. *Adv Eng Softw* 105:30–47
  36. Yang XS, Deb S (2009) Cuckoo search via Lévy flights. In: *Nature and biologically inspired computing, NaBIC 2009, world congress on IEEE*, pp 210–214
  37. Fathollahi-Fard AM, Hajighaei-Keshteli M, Tavakkoli-Moghaddam R (2018) The social engineering optimizer (SEO). *Eng Appl Artif Intell* 72:267–293
  38. Kaveh A, Farhoudi N (2013) A new optimization method: dolphin echolocation. *Adv Eng Softw* 59:53–70
  39. Yang XS (2010) Firefly algorithm, stochastic test functions and design optimisation. *Int J Bio Inspired Comput* 2(2):78–84
  40. Li MD, Zhao H, Weng XW, Han T (2016) A novel nature-inspired algorithm for optimization: virus colony search. *Adv Eng Softw* 92:65–88
  41. Zhao W, Wang L, Zhang Z (2019) Supply-demand-based optimization: a novel economics-inspired algorithm for global optimization. *IEEE Access* 7:73182–73206
  42. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Adv Eng Softw* 114:163–191
  43. Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. *Commun Nonlinear Sci Numer Simul* 17(12):4831–4845
  44. Cuevas E, Cienfuegos M, Zaldívar D, Pérez-Cisneros M (2013) A swarm optimization algorithm inspired in the behavior of the social-spider. *Expert Syst Appl* 40(16):6374–6384
  45. Askarzadeh A (2014) Bird mating optimizer: an optimization algorithm inspired by bird mating strategies. *Commun Nonlinear Sci Numer Simul* 19(4):1213–1228
  46. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. *Inf Sci* 179(13):2232–2248
  47. Formato RA (2007) Central force optimization: a new meta-heuristic with applications in applied electromagnetics. *Prog Electromagn Res* 77:425–491
  48. Bayraktar Z, Komurcu M, Bossard JA, Werner DH (2013) The wind driven optimization technique and its application in electromagnetics. *IEEE Trans Antennas Propag* 61(5):2745–2757
  49. Xie L, Zeng J (2010) The performance analysis of artificial physics optimization algorithm driven by different virtual forces. *ICIC Express Lett* 4(1):239–244
  50. Abedinpourshotorban H, Shamsuddin SM, Beheshti Z, Jawawi DNA (2016) Electromagnetic field optimization: a physics-inspired metaheuristic optimization algorithm. *Swarm Evolut Comput* 26:8–22
  51. Zhao W, Wang L, Zhang Z (2019) A novel atom search optimization for dispersion coefficient estimation in groundwater. *Future Gener Comput Syst* 91:601–610
  52. Zhao W, Wang L, Zhang Z (2019) Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. *Knowl Based Syst* 163:283–304
  53. Kaveh A, Bakhshpoori T (2016) Water evaporation optimization: a novel physically inspired optimization algorithm. *Comput Struct* 167:69–85
  54. Shah-Hosseini H (2011) Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation. *Int J Comput Sci Eng* 6(1–2):132–140
  55. Chuang CL, Jiang JA (2007) Integrated radiation optimization: inspired by the gravitational radiation in the curvature of space-time. In: *IEEE congress on evolutionary computation, CEC 2007. IEEE*, pp 3157–3164
  56. Shah-Hosseini H (2009) The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm. *Int J Bio Inspired Comput* 1(1–2):71–79
  57. Kaveh A, Dadras A (2017) A novel meta-heuristic optimization algorithm: thermal exchange optimization. *Adv Eng Softw* 110:69–84
  58. Kaveh A, Talatahari S (2010) A novel heuristic optimization method: charged system search. *Acta Mech* 213(3):267–289
  59. Zheng M, Liu G, Zhou C, Liang Y, Wang Y (2010) Gravitation field algorithm and its application in gene cluster. *Algorithms Mol Biol* 5(1):32
  60. Javidy B, Hatamlou A, Mirjalili S (2015) Ions motion algorithm for solving optimization problems. *Appl Soft Comput* 32:72–79
  61. Mirjalili SA, Hashim SZM (2012) BMOA: binary magnetic optimization algorithm. *Int J Mach Learn Comput* 2(3):204
  62. Zheng YJ (2015) Water wave optimization: a new nature-inspired metaheuristic. *Comput Oper Res* 55:1–11
  63. Flores JJ, López R, Barrera J (2011) Gravitational interactions optimization. In: *International conference on learning and intelligent optimization*. Springer, Berlin, pp 226–237
  64. Tamura K, Yasuda K (2011) Primary study of spiral dynamics inspired optimization. *IEEE Trans Electr Electron Eng* 6(S1):S98–S100
  65. Rao RV, Savsani VJ, Vakharia DP (2012) Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. *Inf Sci* 183(1):1–15
  66. Zarand G, Pazmandi F, Pál KF, Zimányi GT (2002) Using hysteresis for optimization. *Phys Rev Lett* 89(15):150201
  67. Shen J, Li Y (2009) Light ray optimization and its parameter analysis. In: *International joint conference on computational sciences and optimization, CSO 2009, vol 2. IEEE*, pp 918–922
  68. Kripka M, Kripka RML (2008) Big crunch optimization method. In: *International conference on engineering optimization, Brazil*, pp 1–5
  69. Patel VK, Savsani VJ (2015) Heat transfer search (HTS): a novel optimization algorithm. *Inf Sci* 324:217–246
  70. Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012) Water cycle algorithm: a novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput Struct* 110:151–166
  71. Moghaddam FF, Moghaddam RF, Cheriet M (2012) Curved space optimization: a random search based on general relativity theory. *arXiv preprint arXiv:1208.2214*

72. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Trans Evolut Comput* 1(1):67–82
73. O'Neill RV, Deangelis DL, Waide JB, Allen TF, Allen GE (1986) A hierarchical concept of ecosystems, vol 23. Princeton University Press, Princeton
74. Giannakos MN, Krogstie J, Aalberg T (2016) Video-based learning ecosystem to support active learning: application to an introductory computer science course. *Smart Learn Environ* 3(1):11
75. Yang X-S, Deb S (2013) Multiobjective cuckoo search for design optimization. *Comput Oper Res* 40:1616–1624
76. Viswanathan GM, Afanasyev V, Buldyrev SV, Murphy EJ, Prince PA, Stanley HE (1996) Lévy flight search patterns of wandering albatrosses. *Nature* 381:413–415
77. Brown C, Liebovitch LS, Glendon R (2007) Lévy flights in Dobe Ju/'hoansi foraging patterns. *Hum Ecol* 35:129–138
78. Ning AP, Zhang XY (2013) Convergence analysis of artificial bee colony algorithm. *Control Decis* 28(10):1554–1558
79. Solis FJ, Wets RJB (1981) Minimization by random search techniques. *Math Oper Res* 6(1):19–30
80. Luo J, Li X, Chen M (2010) The Markov model of shuffled frog leaping algorithm and its convergence analysis. *Dianzi Xuebao (Acta Electronica Sinica)* 38(12):2875–2880
81. Liang JJ, Qu BY, Suganthan PN (2013) Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical report, Nanyang Technological University, Singapore
82. Jain M, Singh V, Rani A (2019) A novel nature-inspired algorithm for optimization: squirrel search algorithm. *Swarm Evolut Comput* 44:148–175
83. Zhao W, Wang L (2016) An effective bacterial foraging optimizer for global optimization. *Inf Sci* 329:719–735
84. Nowcki H (1974) Optimization in pre-contract ship design. In: Fujita Y, Lind K, Williams TJ (eds) Computer applications in the automation of shipyard operation and ship design, vol 2. Elsevier, New York, pp 327–338
85. Ray T, Liew KM (2003) Society and civilization: an optimization algorithm based on the simulation of social behavior. *IEEE Trans Evolut Comput* 7(4):386–396
86. Liu H, Cai Z, Wang Y (2010) Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Appl Soft Comput* 10:629–640
87. Zhang M, Luo W, Wang X (2008) Differential evolution with dynamic stochastic selection for constrained optimization. *Inf Sci* 178:3043–3074
88. Wang Y, Cai Z, Zhou Y, Fan Z (2009) Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint handling technique. *Struct Multidisc Optim* 37:395–413
89. Gandomi AH, Yang XS, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Eng Comput* 29(1):17–35
90. Chickermane H, Gea HC (1996) Structural optimization using a new local approximation method. *Int J Numer Methods Eng* 39(5):829–846
91. Cheng MY, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. *Comput Struct* 139:98–112
92. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. *Knowl Based Syst* 89:228–249
93. Belegundu AD (1982) A study of mathematical programming methods for structural optimization. Department of Civil and Environmental Engineering, University of Iowa, Iowa City
94. Coello CAC (2000) Use of a self-adaptive penalty approach for engineering optimization problems. *Comput Ind* 41(2):113–127
95. Coello CAC, Montes EM (2002) Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Adv Eng Inf* 16:193–203
96. Coello CAC, Becerra RL (2004) Efficient evolutionary optimization through the use of a cultural algorithm. *Eng Optim* 36:219–236
97. He Q, Wang L (2006) An effective co-evolutionary particle swarm optimization for engineering optimization problems. *Eng Appl Artif Intell* 20:89–99
98. He Q, Wang L (2007) A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Appl Math Comput* 186:1407–1722
99. Dos Santos Coelho L (2010) Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. *Expert Syst Appl* 37(2):1676–1683
100. Parsopoulos KE, Vrahatis MN (2005) Unified particle swarm optimization for solving constrained engineering optimization problems. In: International conference on natural computation. Springer, Berlin, pp 582–591
101. Huang FZ, Wang L, He Q (2007) An effective co-evolutionary differential evolution for constrained optimization. *Appl Math Comput* 186(1):340–356
102. Mezura-Montes E, Coello CAC (2005) Useful infeasible solutions in engineering optimization with evolutionary algorithms. In: MICAI 2005. Lecture notes in artificial intelligence, vol 3789, pp 652–662
103. Kannan BK, Kramer SN (1994) An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *J Mech Des* 116:405–411
104. Akay B, Karaboga D (2012) Artificial bee colony algorithm for large-scale problems and engineering design optimization. *J Intel Manuf* 23(4):1001–1014
105. Askarzadeh A (2016) A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Comput Struct* 169:1–12
106. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
107. Ngo TT, Sadollah A, Kim JH (2016) A cooperative particle swarm optimizer with stochastic movements for computationally expensive numerical optimization problems. *J Comput Sci* 13:68–82
108. Wang L, Li LP (2010) An effective differential evolution with level comparison for constrained engineering design. *Struct Multidisc Optim* 41:947–963
109. Montes E, Coello CAC, Reyes JV (2006) Increasing successful offspring and diversity in differential evolution for engineering design. In: Proceedings of the seventh international conference on adaptive computing in design and manufacture, pp 131–139
110. Rao BR, Tiwari R (2007) Optimum design of rolling element bearings using genetic algorithms. *Mech Mach Theory* 42(2):233–250
111. Gupta S, Tiwari R, Shivashankar BN (2017) Multi-objective design optimization of rolling bearings using genetic algorithm. *Mech Mach Theory* 42:1418–1443
112. Rao RV, Savsani VJ, Vakharia DP (2011) Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Comput Aided Des* 43(3):303–315
113. Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2013) Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. *Appl Soft Comput* 13(5):2592–2612

114. Osyczka A (2002) Evolutionary algorithms for single and multicriteria design optimization: studies in fuzziness and soft computing. PhysicaVerlag, Heidelberg
115. Deb K, Srinivasan A (2005) Innovization: innovative design principles through optimization. Kanpur Genetic Algorithms Laboratory (KanGAL), Indian Institute of Technology Kanpur, KanGAL report number 2005007
116. McElwee CD (1980) Theis parameter evaluation from pumping tests by sensitivity analysis. *Ground Water* 18(1):56–60
117. Hui J, Bo C, Hongyu P (2009) Groundwater dynamics. Geological Publishing House, Beijing
118. Srivastava R, Guzman-Guzman A (1998) Practical approximations of the well function. *Groundwater* 36(5):844–848
119. Hantush MS, Jacob CE (1955) Non-steady radial flow in an infinite leaky aquifer. *Trans Am Geophys Union* 36(1):95–100
120. Samuel MP, Jha MK (2003) Estimation of aquifer parameters from pumping test data by genetic algorithm optimization technique. *J Irrig Drain Div* 129(5):348–359
121. Yeh HD, Lin YC, Huang YC (2007) Parameter identification for leaky aquifers using global optimization methods. *Hydrol Process* 21(7):862–872

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