scientific reports

Check for updates

OPEN

An aphid inspired metaheuristic optimization algorithm and its application to engineering

Renyun Liu[1], Ning Zhou[1], Yifei Yao[2] & Fanhua Yu[3]✉

The biologically inspired metaheuristic algorithm obtains the optimal solution by simulating the living habits or behavior characteristics of creatures in nature. It has been widely used in many fields. A new bio-inspired algorithm, Aphids Optimization Algorithm (AOA), is proposed in this paper. This algorithm simulates the foraging process of aphids with wings, including the generation of winged aphids, flight mood, and attack mood. Concurrently, the corresponding optimization models are presented according to the above phases. At the phase of the flight mood, according to the comprehensive influence of energy and the airflow, the individuals adaptively choose the flight mode to migrate; at the phase of attack mood, individuals use their sense of smell and vision to locate food sources for movement. Experiments on benchmark test functions and two classical engineering design problems, indicate that the desired AOA is more efficient than other metaheuristic algorithms.

Optimization has been widely used in engineering, aerospace, medical and many other fields[1]. Many optimization problems of practical as well as theoretical importance consist of searches for the "best" configuration of a set of variables to achieve optimal goals[2]. Currently, the prevailing methods for solving optimization problems are deterministic algorithms and metaheuristic algorithms. Although deterministic algorithms are efficient and useful in solving unimodal problems, most optimization problems are multimodal in the real world. For these problems, deterministic algorithms can easily fall into local optima[3].This leads to the need for more reliable optimization techniques[4]. Metaheuristic algorithms are proposed to meet such a need. A metaheuristic algorithm is a stochastic optimization algorithm. As such, it combines a stochastic method with a local search method. Compared with deterministic algorithms, metaheuristic algorithms have attracted considerable attention owing to their efficiency[5]. Many deterministic optimization methods tend to fall into local optima easily because of the lack of randomness. In comparison, a metaheuristic algorithm uses randomness in local searches. As a result, it can effectively avoid being trapped in a local optimal solution.

Metaheuristic algorithms can be divided into two main categories: single solution-based and population-based[6]. The difference between the two approaches is that the population-based approach performs a complete calculation of the algorithm with a set of solutions rather than a single solution. The population-based approach, which simulates the behavior of groups by observing them and applying it to the optimization fields, can be further divided into four categories[7,8]: evolutionary-based algorithms, swarm intelligence-based algorithms, human-based algorithms, and physics and chemistry-based algorithms. Although these algorithms are classified differently, they have one thing in common: the search phase is divided into the exploration phase and the exploitation phase[9] In the exploration phase, the algorithm explores multiple regions of the decision space benefitting from globality and randomness. The exploitation phase is usually carried out after the exploration phase is completed. At this phase, the algorithm pays more attention to the local area to improve the accuracy of the algorithm.

In recent years, metaheuristic algorithms have been widely used because of their reliability. However, according to the No Free Lunch (NFL) principle[10], these metaheuristic algorithms cannot obtain global optimal solutions for all optimization problems. Inspired by the NFL principle, this study proposes a new algorithm named "Aphid Optimization Algorithms (AOA)" to broaden the investigation of metaheuristic optimization algorithms.

The basic idea of an AOA is to simulate aphids' behavior when searching for food, including the stage of winged aphids' production, flight mood, and attack mood. To the best of our knowledge, this is the first optimization algorithm based on the foraging behavior of aphids. In the generation phase, the clustering method is applied to simulate the generation of winged aphids; in the flight mood, the crowding state of the aphid population is simulated by introducing the crowding distance, and aphid individuals select their flight mode

[1]Department of Mathematics, Changchun Normal University, Changchun 130032, Jilin, China. [2]Department of Computer Science, Changchun Normal University, Changchun 130032, Jilin, China. [3]Department of Computer Science, Beihua University, Jilin 132013, Jilin, China. ✉email: yufanhua@beihua.edu.cn
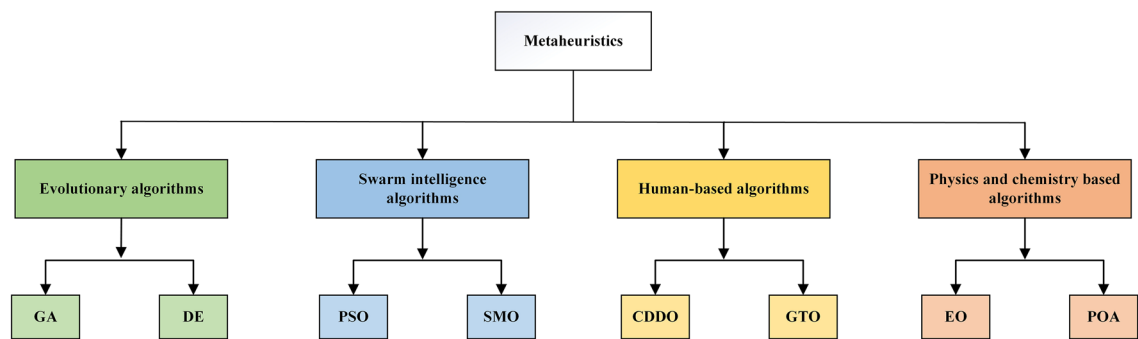
Figure 1. Classification of population-based metaheuristic algorithms**.** Four classes of population-based metaheuristics are introduced, and the corresponding more classical algorithms are cited.

adaptively according to the current crowding state, enhancing the algorithm's exploration ability and diversity; in the attack mood, the optimal solution is guided to ensure accuracy. The main contributions of this paper can be summarized as follows:

- A new bio-inspired metaheuristic algorithm, the Aphid Optimization Algorithm, is proposed.
- The global search and local search are considered in the AOA to ensure the exploration and development capability of the algorithm.
- The effectiveness of the AOA is verified through simulation experiments and the solution of constrained engineering problems. These experiments demonstrate that AOA not only has the feature of high convergence accuracy but also has a fast convergence speed.

The rest of this paper is organized as follows: " Literature review" section reviews the related literature on population-based metaheuristic algorithms. In "Aphid optimization algorithm", aphid behavior and aphid optimization algorithm are introduced in detail. "Experimental results and discussion" Section first presents and analyzes the experimental results of AOA for benchmark optimization problems. Then AOA is used to solve two classic engineering design problems. Finally, concluding remarks are presented in "Conclusion" section

Literature review

Significant attention in the optimization field has been given to population-based optimization algorithms due to their simplicity and efficiency. Population-based algorithms can be classified into four categories: evolutionary-based algorithms, human-based algorithms, physics, and chemistry-based algorithms, and swarm intelligence-based algorithms (see Fig. 1).

Evolutionary-based algorithms are generally based on Darwin's evolutionary theory. Several popular evolutionary-based algorithms, such as the Genetic Algorithm (GA)[11] and Differential Evolution (DE) algorithm[12] have been proved to be highly efficient. GA uses the methods of reproduction, mutation, and selection among individuals in the population to exchange information for the optimal solution. In contrast, for DE, the mutation vector is generated by mutating the difference vector, which is crossed with the parental individuals to generate a new individual. This new individual is directly compared with the parental individual.

Human-based algorithm simulate many aspects of human behavior. Child Drawing Development Optimization (CDDO)[13] is among the powerful human-based algorithms proposed recently. The motivation of CDDO is children's learning behavior and cognitive development in early childhood. Group Teaching Optimization (GTO)[14], Poor and Rich Optimization Algorithm (PRO)[15], Supply-Demand-Based Optimization (SDO)[16], Search and Rescue Optimization Algorithm(RSO)[17], and Student Psychology Based Optimization Algorithm (SPBO)[18] are other popular examples of human-based algorithms.

Physics and Chemistry-based algorithms are inspired by chemical reaction phenomena or laws of physical phenomena. For example, Equilibrium Optimizer (EO)[19] is a new optimization algorithm inspired by the control volume mass balance model, which is used to estimate the dynamic and equilibrium state. Billiards-inspired Optimization Algorithm (BOA)[20], Big-Bang Big-Crunch(BB-BC)[21], Central Force Optimization (CFO)[22], Electron Radar Search (ERS) algorithm[23], Gravitational Search (GS) algorithm[24], and Planet Optimization Algorithm (POA)[25] are also representatives of this kind of algorithms.

Swarm intelligence-based algorithms originate from simulations of biological group behavior or foraging behavior in nature. In such an algorithm, the search and optimization process is simulated as evolving or foraging process to solve optimization problems. Particle Swarm Optimization (PSO)[26], one of the most popular swarm intelligence-based algorithms, is inspired by birds foraging. The algorithm gives the position and velocity of each particle, and each particle updates its position by updating its velocity. Through iterative search, the population can constantly find better positions to get optimal solutions for optimization. Butterfly Optimization Algorithm (BOA)[27] solves optimization problems based on the foraging process of butterflies. Butterflies use their sensors to locate the food source. In BOA, it is assumed that each butterfly produces a scent of a certain intensity, which is transmitted and sensed by other butterflies in the area. Harris Hawks Optimization (HHO) algorithm[28] finds the optimal solution by simulating the predation behavior of Harris Hawks. According to the current state of the prey, Harris Hawks choose different attack methods to hunt. Slime Mould Algorithm (SMA)[29] simulates the

diffusion and foraging behavior of slime mold. It uses adaptive weights to simulate the process of positive and negative feedback generated by the propagation wave based on biological oscillators, forming an optimal path to connect food. Additionally, some scholars have proposed other effective excellent algorithms, such as, Fitness Dependent Optimizer (FDO)[30], Ebola Optimization Search Algorithm(EOSA)[31], Aquila Optimizer(AO)[32], Farmland fertility[33], Barnacles Mating Optimizer (BMO)[34], and Falcon Optimization Algorithm(FOA)[35].

Despite their abilities to solve optimization problems, existing optimization methods face challenges. For instance, the accuracy of GA[11] needs to be improved; DE[12] tends to fall into a local optimum; and PSO[26] and BOA[27] may converge prematurely, resulting in the decline of population diversity. In this article, to resolve some of the shortcomings of the above optimization algorithms, we propose the Aphid Optimization Algorithm (AOA). The algorithm simulates aphids' foraging behavior while taking into account natural environmental factors, such as density, temperature, light, and wind. By designing a mechanism for adaptively selecting flight mode, the diversity of the population is increased and the exploration ability of the algorithm is improved. At the same time, using the optimal solution to guide individuals' movement could ensure the algorithm's accuracy.

Aphid optimization algorithm

In this section, the general biology of the aphid and the optimization model of the proposed algorithm will be discussed in detail.

Biological habits.

Aphids are herbivorous insects, which are widely distributed around the world, but mainly concentrated in temperate regions. About 4700 species of aphids have been discovered[36]. Like other species of migratory insects, many species of aphids possess the nature of migration. Migration allows aphids to successfully avoid adverse environments to find appropriate host plants for feeding, and the winged aphid is the prerequisite for aphid migration. The generation of winged aphids is influenced by many factors, including crowded environment, temperature, light, and so on[37]. As a result, pheromones will be produced among populations to stimulate the production of winged aphids to migrate.

Aphids in the flight stage can be divided into active flight and passive flight according to the nature of flight. Active flight is the spiral flight under the control of the aphid itself, while passive flight is the linear flight driven by the wind[38]. Such uninterrupted flight will cause the winged aphid's energy reserves to be constantly depleted. When the stored energy is low, the aphid cannot fly on its own but relies on the wind to migrate. When there is more energy of its own, the aphid power system and resistance are stronger, and individuals perform the active flight. At this time, when the population is more crowded, the poorer survival conditions cause individuals to migrate to sparse areas. With the increase in flight time, the aphids are inclined to the long-wave light of the hosts and enter the attack stage.

In the attack stage, the interaction of smell and vision lures individuals to land and find hosts to feed on. Winged aphids search for their hosts, mainly by identifying host-specific volatiles through olfactory sensors. The individual's sense of smell navigates its landing based on the odor of the host plant. At the same time, the yellow wave light in the host during flight is a strong stimulus to the visual organs of winged aphids[39]. Therefore, in locating the host location, the visual signal reflected from the host has a solid attraction to aphids. In summary, the foraging process of aphids can be broadly summarized in the following three stages: the production of winged aphids, migration, and the search for hosts to feed.

Mathematical model.

Kaveh proposed the Cyclical Parthenogenesis Algorithm (CPA)[40], which was inspired by the reproduction behavior of aphids. In the algorithm, some rules are enforced according to the life cycle of aphids, and it iteratively updates the solutions with these rules followed. The aphid optimization algorithm proposed in this paper is inspired by the whole process of aphid foraging, including the generation phase of the winged aphids, the phase when winged aphids migrate in order to find the host, and the phase when winged aphids locate the host location to feed. We call these three phases generation stage, flight mood, and attack mood. According to the process of foraging, we will establish the mathematical model for each of the three phases. The whole process of flight and attack moods is shown in Fig. 2. The pseudo code of the entire process of AOA is shown in Algorithm 1.
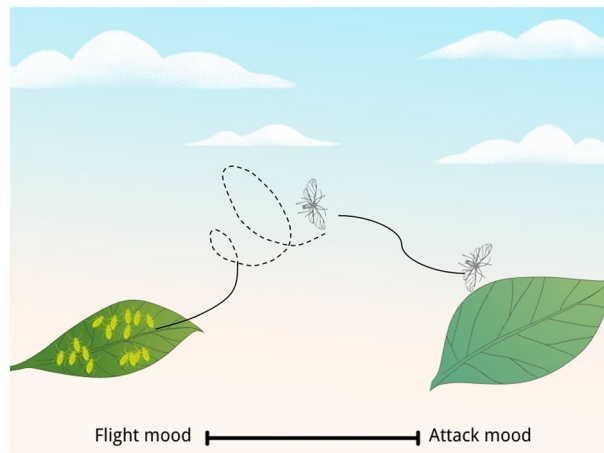
Figure 2. Process of flight mood and attack mood. The simulation of aphid migration and attack.

Algorithm 1 Aphid optimization algorithm
Input
NC: number of individuals in aphid swarm
N: number of clusters
D: dimension of the search space
T: the maximum number of iterations
t: the number of iterations at current
Output
H_{smell}: the global optima
1: Generate the initial population of *NC* aphids
2: Partition the population *NC* into *N* clusters
3: Find the best individual in each cluster $X = \{x_1, x_2, \cdots, x_N\}$
4: Calculate the fitness of each individual in *X*
4: Rank the fitness values and find the current best individual
5: If $A < rand$ then
6: Update global optima using Algorithm 2
7: Else
8: Update global optima using Algorithm 3
9: Return H_{smell}
10: End procedure

Generation stage. Before taking off, individuals need to evolve into winged aphids in order to fly. However, not all aphids can evolve into winged aphids because of the influence of temperature, light, and crowding between populations. At the same time, aphids are not evenly distributed in their habitats but clustered in multiple locations. In order to simulate this stage, we adopt the method of K-means clustering[41].

First, we randomly initialize to generate *NC* individuals, and perform K-means clustering on these individuals to generate *N* classes. In each class, according to the fitness value, the optimal individual is selected as the winged aphid for the next evolutionary operation.

Flight mood. During the flight mood, the individuals begin the exploration phase. With the increase in flight time, the energy stored by the individuals gradually decreases. This process can be described mathematically as:

$$E = 1 - \frac{t}{T} \tag{1}$$

where *E* denotes the energy of the aphid. The influence of wind as the driving force for aphids to fly cannot be underestimated. The effect of wind in this paper is represented as *w*, which can be simulated as a random number drawn from the random variable uniformly distributed in (0, 1) because the wind is randomly varying.

When the energy stored in an individual is relatively small, that is, when $E/w < 1$, aphids perform the passive flight. In this situation, the aphids fly in a straight line according to the direction of the wind. This process is simulated as:
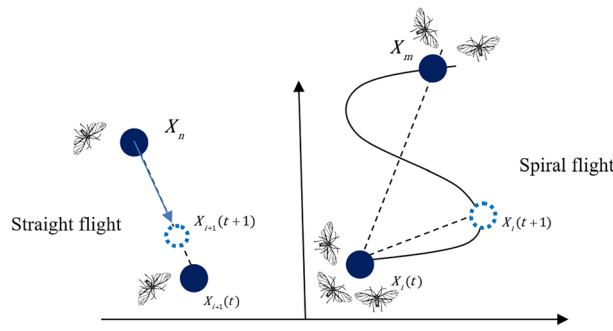
4

Figure 3. Movement of aphids in flight mood. Simulation of straight flight and spiral flight of aphid flight phase.

$$X_i(t+1) = w * (X_n(t) - X_i(t)) \tag{2}$$

where X_n is a randomly selected individual in the population and X_i denotes the position of the ith individual in the current population. When the individual's stored energy is large, that is, when $E/w \geq 1$, the aphids fly actively. At this time, the aphids perform a spiral flight under their control. In this paper, the crowding distance $CD(X_i)$ is used to judge the crowding degree of the individual, which can be denoted as follows:

$$CD(X_i) = \sum_{j=1}^{N} d_{i,j} \tag{3}$$

where $CD(X_i)$ represents the Euclidean distance from the i th individual to the j th individual in the population. The smaller the crowding distance is, the more crowded the current individual is. When the crowding distance is less than the average crowding distance $CD(X_{avg})$ between the population, it indicates that the individual is in a more crowded state; otherwise, it indicates that the individual is in a relatively loose state.

At this time, considering that the crowded environment is not conducive to the evolution of aphids, individuals will adaptively select appropriate individuals to guide them to spiral flight according to the crowding distance. Therefore, when the individual is in a crowded state, the individual with the largest crowding distance is selected to guide the current solution to the loose area. Otherwise, the crowd chooses a random individual to guide the flight. This process is simulated as:

$$X_i(t+1) = X_i(t) + w * (e^R \cos(2\pi R) * (X_m(t) - X_i(t))) + w * (e^R \sin(2\pi R) * (X_m(t) - X_i(t))) \ if \ CD(X_i) < CD(X_{avg}) \tag{4}$$

$$X_i(t+1) = X_i(t) + w * (e^R \cos(2\pi R) * (X_n(t) - X_i(t))) + w * (e^R \sin(2\pi R) * (X_n(t) - X_i(t))) \ else \tag{5}$$

Here X_m is the individual with the largest crowding distance in the current population and R is the spiral radius which is a uniformly distributed random variable in $(-1, 1)$. Figure 3 depicts the movement of aphids in flight mood. The pseudo code of flight mood of AOA is shown in Algorithm 2.

Algorithm 2 Flight mood

Input: E, R
Output:
H_{smell}: the global optima of in the current iteration
1： procedure Flight mood
2： For i=1 to N do
3： IF $E / w < 1$ then
4： calculate X_i according to equation.(2)
5： Else
6： if $CD(X_i) < CD(X_{avg})$
7： calculate X_i according to equation.(4)
8： else
9： calculate X_i according to equation.(5)
10： end if
11： end IF
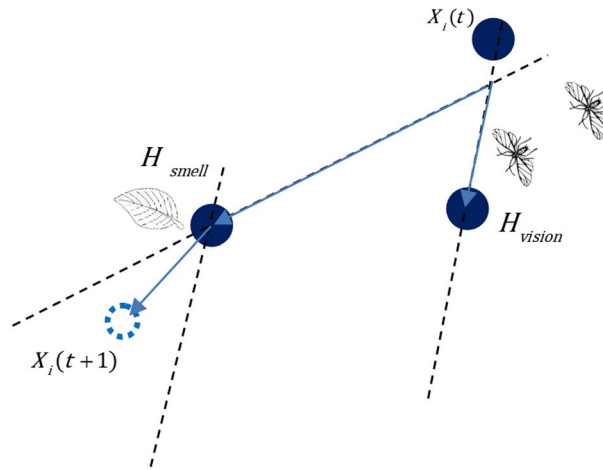12： Retur H_{smell}n
13： end procedure

Figure 4. Movement of aphids in attack mood. Simulation of aphid's movement in attack phase.

Attack mood. In the attack mood, individuals begin the exploitation phase. After entering the attack mood, the aphid will find its host, and it mainly navigates landing position by olfactory sense according to the smell of host plants, so the volatiles of host plants play an important role in the tendency of the aphid to fly to its host. Simultaneously, the visual sense is also a significant player in the search for the host. The long wave light of the host plant will affect the landing of aphids because aphids have a strong tendency to the long wave light. The location of host plants by smell H_{smell} is regarded as the global optimal solution, and the location of host plants by vision H_{vision} is regarded as the personal optimal location. Figure 4 shows the movement of aphids in an attack mood. This process is simulated as:

$$X_i(t+1) = H_{smell}(t) + r * ((H_{smell}(t) - X_i(t)) + (H_{vision}(t) - X_i(t))) \tag{6}$$

where r is a random number between $(0, 0.1)$. The pseudo code of Flight mood of AOA is shown in Algorithm 3.

Algorithm 3 Attack mood

Input: E, r
Output
H_{smell}: the global optima of in the current iteration
1： procedure： Attack mood
2： For i=1 to N do
3： calculate X_i according to equation.(6)
4： update H_{smell}
5： end For
6： H_{smell} is output as the global optima of the fitness function
7： Return H_{smell}
8： end procedure

Transition. Considering that aphids are stimulated by yellow waves during flight, the stimulation degree by yellow waves is simulated as follows:

$$I = \frac{1}{1 + e^{-\frac{1}{S+1}}} \tag{7}$$

where S is the absolute value of Euclidean distance between the average position of the population and the host plant. The closer the distance, the greater the degree of stimulation and the value of I.

Aphids have a tendency to long-wave light from plants in the process of flight, so they enter the attack mood. We use the following formula to express the tendency of individuals to long-wave light:

$$A = I * \left(1 - \frac{t}{T}\right) \tag{8}$$

Let $rand$ be a number drawn from the uniformly distributed random variable in $(0, 1)$. Then we can determine if the flight mood or the attack mood is performed as follows: when $A < rand$, the flight mood is performed; otherwise, the attack mood is performed.

Algorithm	Parameters	Value
PSO	Learning factors C_1	2
	Learning factors C_2	2
	Inertia weight	0.798
DE	Scaling factor	0.8
	Crossover constant	0.2
GA	Crossover probability	0.8
	Mutation probability	0.1
BOA	p	0.8
	power exponent	0.1
SMA	z	0.03
HHO	β	1.5
MFO	helix parameters b	1
COA	N_{pack}	10
	N_{coyoye}	5
AOA	NC	1000
	N	50

Table 1. Parameter setting of various optimization algorithms. The specific parameter settings of the algorithms compared in this paper.

Experimental results and discussion

In order to verify the efficiency of the proposed algorithm, we compare the performance of the proposed algorithm with that of other algorithms. From the current popular metaheuristic algorithms, we choose six of them as comparison objects: PSO, DE, GA, BOA, SMA, HHO, MFO[42], COA[43], and CAMES[44]. The parameter setting of the above algorithms is shown in Table 1. In this section, we compare the proposed algorithm with other algorithms on 23 well-known benchmark test functions from the benchmark set proposed in[45] and 10 benchmark test functions from CEC 2019. The constrained industrial optimization applications are also employed to demonstrate the efficiency of the proposed algorithm. All experiments are run with MATLAB 2018a. The details of the running system are shown in Table 2.

To clearly compare the algorithm performance, in this paper, the maximum number of iterations is set to be 1000, and each function is repeated 30 times. The population size of the above algorithms is designed to be 50. The average values of the convergence results of 30 runs are calculated as an evaluation metric of the accuracy of the algorithm. The smaller the average value, the higher the convergence accuracy of the algorithm. Meanwhile, the convergence speed of the algorithm is compared by plotting the convergence curves.

Benchmark function validation. We test the performance of the proposed AOA on the proposed benchmark sets. These 23 well-known functions are characterized as multimodal and unimodal as shown in Table 3. Additional new test functions of CEC 2019 are shown in Table 4. Here Type denotes the type of the function such as M: Multimodal and U: Unimodal, Dim represents the dimension of function, Range denotes the definition domain of the function, and Optima denotes the optimal value of the function. The simulation results of AOA on benchmark functions used in this study are shown in Table 5.

Analysis of simulation results. To compare AOA with other algorithms, the average and standards are listed in Table 6, where the mean in bold indicates optimal. We can see from the table that in most test functions, AOA performs better than other algorithms on both indicators: the average and standard deviation. From Table 6, it can be concluded that AOA obtains the best results on 20 out of 33 benchmark functions, ranking first in the number of best results among all algorithms. The second in the number of best results is SMA, which has 17 out of 33 benchmark functions. According to the results in Table 6, AOA algorithm is significantly superior in solving the unimodal test function and multimodal test functions.

In addition, we use Fig. 5 to show the distribution of population on a multimodal function named Ackley's Function after 30 iterations of flight mood and 20 iterations of the attack mood. In the flight mood, individuals adaptively choose flight modes according to the crowding state among populations, thus maintaining the diversity of populations. In the attack mood, individuals are attracted by the host plants and gather near the optimal solution, which shows the strong exploration ability of the algorithm.

To further demonstrate the significant difference between the proposed algorithm and other algorithms, we perform the Friedman test[46] with $\alpha = 0.05$ significance level, which is shown in Table 7. We list the average ranking of the results of the algorithms on all benchmark test functions with $p < 0.05$ in Table 7. It can be concluded from the table that AOA outperforms all the other algorithms. To further analyze the significance of paired differences, we chose the Nemenyi test as a post hoc analysis method and drew a critical difference plot to visualize the results. From Fig. 6, it can be concluded that AOA is comparable with SMA, HHO, CMAES, MFO and COA, while it is significantly better than DE, BOA, PSO, and GA.

System information	Content
Operating system	Windows 10
Processor	Intel(R) Core(TM) i5-7200U
CPU	@ 2.50 GHz(4 CPUs), ~ 2.7 GHz
RAM	8192 MB

Table 2. System and resource characteristics. The running environment in this study.

No	Type	Formula	Dim	Range	Optima
F1	U	$f(x) = \sum_{i=1}^{n} x_i^2$	30	$[-100,100]$	0
F2	U	$f(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|$	30	$[-10,10]$	0
F3	U	$f(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$	30	$[-100,100]$	0
F4	U	$f(x) = \max(|x_i|, 1 \le i \le n)$	30	$[-100,100]$	0
F5	U	$f(x) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$	30	$[-30,30]$	0
F6	U	$f(x) = \sum_{i=1}^{n} (x_i + 0.5)^2$	30	$[-100,100]$	0
F7	U	$f(x) = \sum_{i=1}^{n} i x_i^4 + \text{rand}(0,1)$	30	$[-1.28,1.28]$	0
F8	M	$f(x) = -\sum_{i=1}^{n} \left[x_i \sin(\sqrt{|x_i|})\right]$	30	$[-500,500]$	−12,569.5
F9	M	$f(x) = \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$	30	$[-5.12,5.12]$	0
F10	M	$f(x) = -20\exp\left(-0.2 \times \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + \exp(1)$	30	$[-32,32]$	0
F11	M	$f(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600,600]$	0
F12	M	$f(x) = \frac{\pi}{n}\left\{10 sin^2(\pi y_i) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10 sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$	30	$[-50,50]$	0
F13	M	$f(x) = (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2$	2	$[-4.5,4.5]$	0
F14	M	$f(x) = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j+(x_i - a_{ij})^6}\right)^{-1}$	2	$[-65.536,65.536]$	1
F15	M	$f(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	$[-5,5]$	0.00030
F16	M	$f(x) = 4x_1^2 - 2.1x_2^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	$[-5,5]$	−1.0316
F17	M	$f(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	$[-5,10]\times[0,15]$	0.398
F18	M	$f(x) = \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)\right] \times \left[30 + (2x_1 - 3x_2)^2\right.$ $\left.\times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)\right]$	2	$[-2,2]$	3
F19	M	$f(x) = -\sum_{i=1}^{4} c_i exp\left(-\sum_{j=1}^{4} a_{ij}\left(x_j - p_{ij}\right)\right)^2$	4	$[0,1]$	−3.86
F20	M	$f(x) = -\sum_{i=1}^{4} c_i exp\left(-\sum_{j=1}^{6} a_{ij}\left(x_j - p_{ij}\right)\right)^2$	6	$[0,1]$	−3.32
F21	M	$f(x) = -\sum_{i=1}^{5}\left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$	4	$[0,10]$	−10.1532
F22	M	$f(x) = -\sum_{i=1}^{7}\left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$	4	$[0,10]$	−10.4028
F23	M	$f(x) = -\sum_{i=1}^{10}\left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$	4	$[0,10]$	−10.5363

Table 3. Standard Benchmark functions. Specific 23 standard test functions, including the dimension, upper and lower bounds, and optimal value of each function.

The convergence speed of an algorithm is also considered to be an important indicator for measuring algorithm performance. Figure 7 plots the convergence curves for 200 iterations of each algorithm. From Fig. 7, it can be seen that AOA outperforms most algorithms in the majority of test functions. As for F8, though the convergence speed of AOA is slower than that of other algorithms at the beginning of the iteration, its ability to find the global optimal solution at the later stage is almost the same as SMA and HHO. F16 is a multimodal function, and there are many local minima distributed in the domain of definition, which makes it more difficult for the algorithm to find the global minimum. Although SMA and HHO converge faster than AOA at the beginning, the accuracy is the same at later stage.

Engineering problems. In this section, we solve two engineering design problems to test the capability of the proposed AOA algorithm and compare the result of AOA with that of other metaheuristic algorithms. These

No	Function	Dim	Range	Optima
cec01	Storn's chebyshev polynomial fitting problem	9	[−8192, 8192]	1
cec02	Inverse hilbert matrix problem	16	[−16384, 16384]	1
cec03	Lennard-jones minimum energy cluster	18	[−4, 4]	1
cec04	Rastrigin's function	10	[−100, 100]	1
cec05	Griewangk's function	10	[−100, 100]	1
cec06	Weierstrass Function	10	[−100, 100]	1
cec07	Modified schwefel's function	10	[−100, 100]	1
cec08	Expanded schaffer's f6 function	10	[−100, 100]	1
cec09	Happy cat function	10	[−100, 100]	1
cec10	Ackley function	10	[−100, 100]	1

Table 4. CEC 2019 benchmark functions. Specific CEC2019 test functions, including the dimension, upper and lower bounds, and optimal value of each function.

Function	Aphid optimization algorithm			
	Mean	SD	Best	Worst
F1	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F3	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F4	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F5	1.81E−29	2.63E−02	2.77E−32	3.64E−25
F6	1.22E−26	5.92E−30	7.35E−28	8.42E−24
F7	1.57E−02	2.41E−02	4.93E−04	4.42E−02
F8	−1.26E+04	4.12E−02	−1.26E+04	−1.26E+04
F9	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F10	8.88E−16	0.00E+00	8.88E−16	8.88E−16
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F12	2.17E−30	4.59E−31	6.25E−31	3.63E−28
F13	1.60E−30	2.09E−30	1.03E−30	5.21E−29
F14	9.98E−01	0.00E+00	9.98E−01	9.98E−01
F15	1.74E−03	1.12E−04	8.63E−04	9.77E−02
F16	−1.03E+00	0.00E+00	−1.03E+00	−1.03E+00
F17	3.98E−01	0.00E+00	3.98E−01	3.98E−01
F18	3.00E+00	7.26E−08	3.00E+00	3.00E+00
F19	−3.86E+00	3.58E−11	−3.86E+00	−3.86E+00
F20	−2.71E+00	1.75E−01	−3.32E+00	−2.56E+00
F21	−1.01E+01	1.87E−15	−1.01E+01	−1.01E+01
F22	−1.04E+01	5.92E−16	−1.04E+01	−1.04E+01
F23	−1.05E+01	1.54E−02	−1.05E+01	−1.05E+01
cec01	6.68E+04	1.65E+04	4.78E+04	7.55E+04
cec02	1.74E+01	4.49E−01	1.75E+01	1.74E+01
cec03	1.27E+01	6.87E−08	1.27E+01	1.30E+01
cec04	9.45E+01	3.30E+01	3.80E+01	1.04E+02
cec05	2.25E+00	2.23E−01	2.11E+00	2.84E+00
cec06	9.08E+00	7.81E−01	8.57E+00	1.12E+01
cec07	2.75E+02	7.96E+01	2.57E+02	3.48E+02
cec08	6.02E+00	5.94E−01	5.96E+00	7.46E+00
cec09	3.15E+00	4.67E−01	2.71E+00	4.32E+00
cec10	2.01E+01	1.15E−01	2.01E+01	2.03E+01

Table 5. Simulation results of AOA. Results of the proposed algorithm on each test function, including best solution, worst solution, average solution, and standard deviation.

problems have several inequality constraints, which show the capability of AOA in solving constrained optimization problems. In this study, the constrained optimization problem is transformed into a series of unconstrained optimization problems to be solved by converting the constraints into penalty functions to be added to the objective function.

Welded beam design. The purpose of this problem is to minimize the cost of welded beam design under the condition of strength as shown in Fig. 8. In this problem, the thickness of weld (h), length (l), height (t), and thickness of the bar (b) are selected as the design variables, and the cost of the welded beam is taken as the objective function. The mathematical formulation of this problem is as follows:

$$\text{Consider } \boldsymbol{x} = [h\,l\,t\,b] = [x_1 x_2 x_3 x_4]$$

$$\text{Minimize } f(\boldsymbol{x}) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14 + x_2)$$

Subject to:

$$g_1(\boldsymbol{x}) = \tau(\boldsymbol{x}) - 13600 \leq 0 \tag{9}$$

$$g_2(\boldsymbol{x}) = \sigma(\boldsymbol{x}) - 30000 \leq 0 \tag{10}$$

$$g_3(\boldsymbol{x}) = x_1 - x_4 \leq 0 \tag{11}$$

$$g_4(\boldsymbol{x}) = 0.10471(x_1^2) + 0.04811 x_3 x_4 (14 + x_2) - 5 \leq 0 \tag{12}$$

$$g_5(\boldsymbol{x}) = \delta(x) - 0.25 \leq 0 \tag{13}$$

$$g_6(\boldsymbol{x}) = 6000 - p_c(x) \leq 0 \tag{14}$$

where

$$\tau(\boldsymbol{x}) = \sqrt{(\tau') + (2\tau'\tau'') \frac{x_2}{2R} + (\tau'')^2} \tag{15}$$

$$\tau' = \frac{6000}{\sqrt{2} x_1 x_2} \tag{16}$$

$$\tau'' = \frac{MR}{J} \tag{17}$$

$$M = 6000 \left(14 + \frac{x_2}{2}\right) \tag{18}$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \tag{19}$$

$$J = 2\left\{ x_1 x_2 \sqrt{2} \left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2 \right] \right\} \tag{20}$$

$$\sigma(\boldsymbol{x}) = \frac{504000}{x_4 x_3^2} \tag{21}$$

$$\delta(\boldsymbol{x}) = \frac{65856000}{(30 \times 10^6) x_4 x_3^3} \tag{22}$$

$$p_c(\boldsymbol{x}) = \frac{4.013 (30 \times 10^6) \sqrt{\frac{x_3^2 x_4^6}{36}}}{196} \left(1 - \frac{x_3 \sqrt{\frac{30 \times 10^6}{4(12 \times 10^6)}}}{28} \right) \tag{23}$$

Variable ranges are given as $0.1 \leq x_1, x_4 \leq 2, 0.1 \leq x_2, x_3 \leq 10$.

		AOA	DE	PSO	BOA	GA	SMA	HHO	MFO	COA	CMAES
F1	Mean	**0.00E+00**	5.60E−03	1.64E−02	5.66E−10	1.55E+01	6.55E−311	1.25E−194	4.12E−31	1.49E+01	6.40E−62
	SD	0.00E+00	2.10E−03	8.61E−03	1.58E−10	7.09E+00	0.00E+00	0.00E+00	1.45E−30	9.74E+00	5.29E−61
F2	Mean	**0.00E+00**	8.15E−03	1.79E+00	5.42E−18	1.27E+00	9.68E−140	2.24E−100	4.32E−19	2.74E−01	1.98E−30
	SD	1.81E−149	1.35E−03	1.09E+00	1.56E−15	2.45E−01	5.30E−139	1.23E−99	4.26E−09	1.15E−01	1.23E−30
F3	Mean	**0.00E+00**	4.59E+04	4.07E−01	5.14E−10	1.38E−03	8.85E−229	3.00E−160	1.56E−01	2.97E+02	1.73E−41
	SD	0.00E+00	9.24E+03	2.48E−01	1.15E−10	2.60E−03	0.00E+00	1.64E−159	6.84E−01	1.34E+02	4.09E−41
F4	Mean	**0.00E+00**	1.94E+01	6.64E−01	4.46E−07	6.85E+00	5.25E−176	4.57E−101	1.47E−06	1.38E+01	5.03E−28
	SD	0.00E+00	1.99E+00	2.88E−01	7.97E−08	1.04E+00	0.00E+00	1.12E−100	6.15E−06	4.00E+00	2.73E−28
F5	Mean	**1.81E−29**	4.34E+02	6.36E+01	2.99E+01	1.47E+04	2.32E−01	5.55E−06	6.66E+02	6.96E+01	7.06E−02
	SD	2.63E−02	8.81E+01	5.08E+01	2.19E−02	5.16E+03	1.84E−01	1.91E−07	1.22E+03	5.46E+02	1.62E−02
F6	Mean	1.22E−26	5.39E−03	1.42E−02	4.78E+00	1.48E+01	6.90E−04	8.99E−06	9.21E−24	1.25E+01	**0.00E+00**
	SD	5.92E−30	1.24E−03	8.01E−03	4.11E−01	5.04E+00	4.76E−04	2.33E−05	9.20E−26	8.01E+00	0.00E+00
F7	Mean	1.57E−02	6.07E−01	9.76E−02	8.08E−04	3.82E−02	4.44E−02	**4.33E−05**	3.71E−02	3.88E−02	2.20E−03
	SD	2.41E−02	1.12E−01	1.70E−02	3.91E−04	1.11E−02	4.06E−02	3.55E−05	3.41E−02	1.70E−02	1.00E−03
F8	Mean	**−1.26E+04**	−1.28E+31	−3.18E+03	−3.23E+03	−9.50E+03	**−1.26E+04**	**−1.26E+04**	−3.28E+03	−4.06E+03	NaN
	SD	4.12E−04	1.08E+29	5.14E+02	1.55E+02	1.10E+02	4.43E−02	1.57E−01	3.81E+02	5.75E+01	NaN
F9	Mean	**0.00E+00**	8.29E+01	4.10E+01	**0.00E+00**	2.76E+01	**0.00E+00**	**0.00E+00**	1.95E+01	7.05E+00	1.45E+01
	SD	0.00E+00	6.91E+00	1.11E+01	0.00E+00	2.94E+00	0.00E+00	0.00E+00	1.23E+01	1.60E+01	1.03E+01
F10	Mean	**8.88E−16**	2.08E−02	3.77E+00	2.22E−11	5.51E+00	**8.88E−16**	**8.88E−16**	4.62E−15	3.98E+00	1.95E−15
	SD	0.00E+00	4.39E−03	9.40E−01	7.62E−13	4.57E−01	0.00E+00	0.00E+00	7.94E−16	1.02E+00	1.72E−15
F11	Mean	**0.00E+00**	7.09E−02	1.51E−02	2.39E−11	4.23E+00	**0.00E+00**	**0.00E+00**	1.23E−01	1.10E+00	7.40E−04
	SD	0.00E+00	3.51E−02	1.46E−02	7.99E−15	6.26E−01	0.00E+00	0.00E+00	5.93E−02	1.38E−01	2.30E−03
F12	Mean	2.17E−30	4.11E−03	2.09E−01	5.97E−01	4.45E−01	7.32E−04	2.13E−07	2.03E−01	1.99E+00	**4.71E−32**
	SD	4.59E−31	1.91E−03	1.06E−01	1.14E−01	6.55E−01	2.74E−03	1.62E−06	5.84E−01	1.49E+00	1.15E−47
F13	Mean	1.60E−30	6.82E−03	3.36E−01	2.92E+00	3.26E+01	7.24E−04	1.45E−06	3.30E−03	4.66E+00	**1.35E−32**
	SD	2.09E−30	3.43E−03	6.06E−01	1.50E−01	1.60E+01	6.22E−04	7.10E−12	5.20E−03	2.32E+00	2.89E−48
F14	Mean	**9.98E−01**	**9.98E−01**	1.49E+00	1.12E+00	**9.98E−01**	**9.98E−01**	**9.98E−01**	1.25E+00	**9.98E−01**	7.51E+00
	SD	0.00E+00	0.00E+00	7.02E−01	3.15E−01	1.67E−08	2.70E−12	0.00E+00	7.10E−01	1.75E−06	5.32E+00
F15	Mean	1.74E−03	2.03E−03	4.36E−04	5.01E−04	5.55E−03	7.44E−03	**3.47E−04**	1.10E−03	4.86E−04	4.16E−03
	SD	1.12E−04	7.02E−04	4.07E−04	3.23E−04	3.17E−02	1.10E−02	1.11E−09	4.55E−04	1.46E−04	1.37E−03
F16	Mean	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	−1.29E+00	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**	**−1.03E+00**
	SD	0.00E+00	0.00E+00	0.00E+00	7.21E−01	5.12E−02	6.29E−12	0.00E+00	2.28E−16	7.20E−17	0.00E+00
F17	Mean	**3.98E−01**	**3.98E−01**	**3.98E−01**	4.05E−01	1.08E+00	**3.98E−01**	**3.98E−01**	3.98E−01	3.98E−01	**3.98E−01**
	SD	0.00E+00	0.00E+00	0.00E+00	1.42E−02	6.04E−01	5.73E−08	0.00E+00	0.00E+00	3.10E−14	0.00E+00
F18	Mean	**3.00E+00**	**3.00E+00**	**3.00E+00**	3.12E+00	5.71E+00	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**
	SD	7.26E−16	5.34E−16	4.19E−16	2.02E−01	1.90E−03	9.01E−12	1.24E−06	9.34E−16	3.13E−15	0.00E+00
F19	Mean	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	−3.79E+00	−3.85E+00	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**	**−3.86E+00**
	SD	3.58E−11	9.36E−16	1.10E−03	9.30E−02	6.67E−04	7.37E−08	0.00E+00	2.26E−15	3.28E−14	9.36E−16
F20	Mean	−2.71E+00	**−3.32E+00**	−3.29E+00	−2.55E+00	−2.84E+00	−3.22E+00	−3.79E+00	−3.22E+00	**−3.32E+00**	−3.27E+00
	SD	1.75E−01	1.97E−12	5.74E−02	3.71E−01	1.04E−05	4.36E−02	2.05E−03	4.73E−02	1.35E−04	6.14E−02
F21	Mean	**−1.01E+01**	−9.59E+00	−5.91E+00	−4.80E+00	−6.12E+00	−1.02E+01	−5.06E+00	−6.89E+00	−1.02E+01	−7.17E+00
	SD	1.87E−15	1.25E−02	3.73E+00	9.52E−02	1.70E−02	3.43E−05	2.43E−06	3.73E+00	1.37E−04	3.86E+00
F22	Mean	**−1.04E+01**	−1.02E+01	−6.68E+00	−4.70E+00	−6.43E+00	**−1.04E+01**	−5.09E+00	−7.99E+00	**−1.04E+01**	−9.64E+00
	SD	5.92E−16	2.31E−02	3.94E+00	1.46E−01	2.65E+00	5.32E−05	8.60E−03	3.41E+00	3.90E−03	2.42E+00
F23	Mean	**−1.05E+01**	−1.04E+01	−9.23E+00	−4.66E+00	−4.92E+00	**−1.05E+01**	−5.13E+00	−7.76E+00	**−1.05E+01**	−9.93E+00
	SD	1.54E−02	2.11E−02	2.82E+00	1.95E−01	2.39E+00	6.40E−05	4.28E−03	3.53E+00	1.35E−04	1.90E+00
cec01	Mean	6.68E+04	1.75E+12	3.59E+12	5.74E+04	3.38E+10	**4.19E+04**	6.20E+04	1.23E+10	5.49E+09	1.15E+09
	SD	1.65E+04	2.11E+10	1.75E+12	9.76E+03	2.39E+10	2.91E+04	4.28E+04	2.10E+10	4.91E+09	7.96E+07
cec02	Mean	1.74E+01	**1.73E+01**	1.41E+04	1.78E+01	3.16E+01	1.74E+01	**1.73E+01**	1.75E+01	1.74E+01	1.47E+02
	SD	4.49E−01	3.74E−15	3.51E+03	1.79E−01	1.21E+01	3.63E−02	0.00E+00	3.74E−03	1.21E−02	3.35E+01
cec03	Mean	**1.27E+01**	**1.27E+01**	**1.27E+01**	**1.27E+01**	**1.27E+01**	**1.27E+01**	**1.27E+01**	**1.27E+01**	**1.27E+01**	**1.27E+01**
	SD	6.87E−08	1.97E−06	1.87E−05	4.42E−04	2.65E−07	4.16E−03	0.00E+00	4.89E−06	9.93E−08	1.20E−03
cec04	Mean	9.45E+01	1.64E+03	3.75E+03	1.61E+04	4.91E+04	**2.91E+01**	4.84E+01	8.16E+01	1.25E+02	2.75E+01
	SD	3.30E+01	3.88E+02	2.09E+02	7.38E+03	2.29E+04	1.13E+01	0.00E+00	1.14E+02	3.55E+01	4.98E+00
cec05	Mean	2.25E+00	3.18E+00	3.28E+00	4.80E+00	3.60E+00	2.31E+00	2.46E+00	2.30E+00	1.49E+00	**1.00E+00**
	SD	2.23E−01	3.45E−01	4.99E−01	5.62E−01	1.80E+00	1.27E−01	0.00E+00	2.41E−01	1.11E−01	2.30E−03
Continued											

		AOA	DE	PSO	BOA	GA	SMA	HHO	MFO	COA	CMAES
cec06	Mean	9.08E+00	1.35E+01	1.13E+01	1.17E+01	1.19E+01	**5.95E+00**	8.36E+00	4.91E+00	7.49E+00	1.14E+01
	SD	7.81E−01	9.93E+00	9.47E+00	4.23E−01	6.14E+00	8.50E−01	7.39E−01	2.09E+00	9.36E−01	4.26E−01
cec07	Mean	2.75E+02	2.98E+03	1.30E+03	8.38E+02	3.04E+03	2.95E+02	3.56E+02	4.51E+02	**2.23E+02**	2.85E+02
	SD	7.96E+01	1.24E+02	4.70E+02	1.78E+02	1.11E+02	1.85E+02	3.30E+02	2.37E+02	9.45E+01	5.01E+02
cec08	Mean	**6.02E+00**	6.26E+00	6.22E+00	6.96E+00	9.92E+00	6.06E+00	6.20E+00	6.02E+00	6.06E+00	6.93E+00
	SD	5.94E−01	3.76E−01	8.63E−01	2.62E−01	8.31E+00	1.01E−01	5.76E−01	6.25E−01	4.88E−01	1.66E−01
cec09	Mean	3.15E+00	6.12E+00	5.26E+00	1.87E+03	3.13E+03	**2.37E+00**	2.68E+00	2.55E+00	2.87E+00	2.45E+00
	SD	4.67E−01	4.12E+00	5.74E−01	5.19E+02	2.82E+02	3.22E−01	6.29E−01	1.24E−01	1.48E−01	3.66E−02
cec10	Mean	**2.01E+01**	2.04E+01	2.09E+01	2.04E+01	1.83E+02	**2.01E+01**	**2.01E+01**	**2.01E+01**	**2.01E+01**	2.04E+01
	SD	1.15E−01	1.68E−01	1.37E−01	7.33E−02	5.30E+01	1.61E−02	9.87E−02	8.42E−02	5.62E−02	1.24E−01

Table 6. Comparison of AOA with different algorithms. The results of AOA with other different algorithms on each test function and the optimal solution is marked in bold.(significant values are in bold).
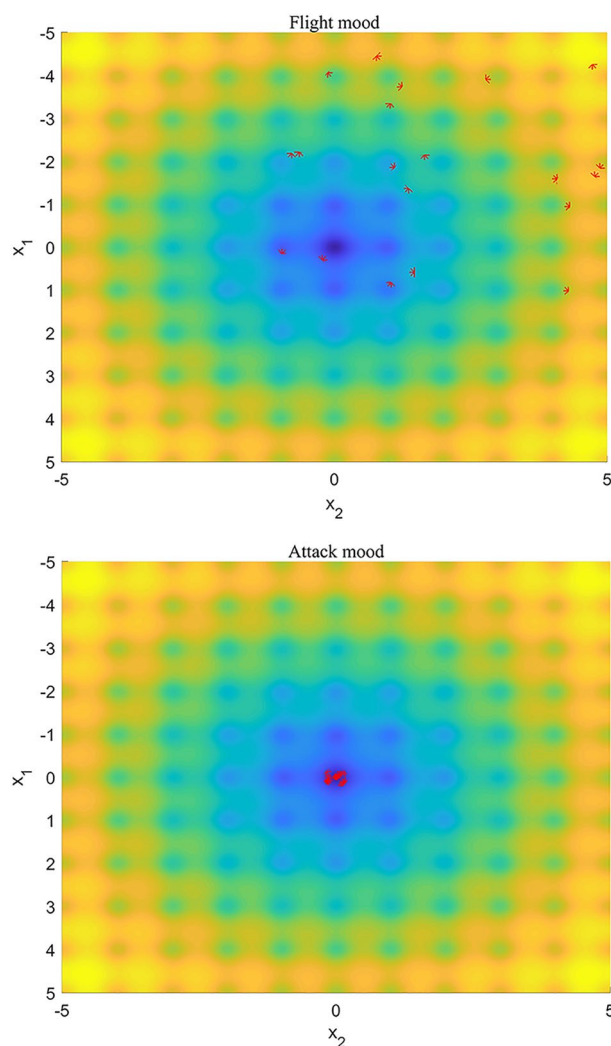
Figure 5. Aphids' locations after 50 iterations in flight mood and their locations after 30 iterations in attack mood. The distribution of 20 aphids on a multimodal function after 30 iterations of flight mood and 20 iterations of the attack mood.

The result of AOA is compared with DE, PSO, and GA as shown in Table 8. The bold values indicate the best one among all methods. The table demonstrates that AOA performs better than other algorithms in this engineering problem.

	Avg	Rank
AOA	3.2969	1
DE	6.4844	7
PSO	7.0313	9
BOA	6.875	8
GA	8.5625	10
SMA	3.6094	2
HHO	3.8281	3
MFO	5.2813	5
CAO	5.4688	6
CMAES	4.5625	2
p-value	9.49061E-19	

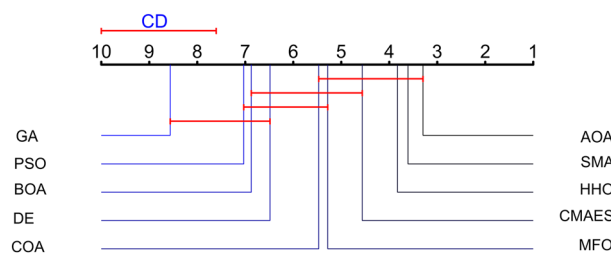**Table 7.** Mean rank and *p*-value of Friedman test. The mean ranking of all compared algorithms on test functions.

Figure 6. Plots of critical differences. Critical Differences plotting by post-hoc test.

Reliability analysis of bolt. The bolted connection of the rigid coupling is optimally designed under the condition of predetermined reliability R (set $R \geq 0.99$), with the lightest bolt mass as the optimization target as shown in Fig. 9. The number of the limit bolt is $4 \leq n \leq 8$, and the diameter of the bolt is $5 \leq d_0 \leq 18$ mm. The mathematical expression is as follows:

$$\text{Consider } \boldsymbol{x} = [nd_0]$$

$$\text{Minimize } f(\boldsymbol{x}) = 2.6 \times 10^{-4} nd_0^3$$

Subject to

$$g_1(\boldsymbol{x}) = \frac{200 - \frac{24 \times 10^4}{\pi nd_0^2}}{2.326\sqrt{14^2 + \left(\frac{12 \times 10^3}{\pi nd_0^2}\right)^2}} - 1 \geq 0 \tag{24}$$

$$g_2(\boldsymbol{x}) = \frac{150 - \frac{12 \times 10^4}{\pi nd_0^2}}{2.326\sqrt{10.5^2 + \left(\frac{6 \times 10^3}{\pi nd_0^2}\right)^2}} - 1 \geq 0 \tag{25}$$

$$g_3(\boldsymbol{x}) = \frac{280 - \frac{7.5 \times 10^4}{\pi nd_0^2}}{2.326\sqrt{25.2^2 + \left(\frac{37.5 \times 10^3}{\pi nd_0^2}\right)^2}} - 1 \geq 0 \tag{26}$$

$$g_4(\boldsymbol{x}) = \frac{n}{4} - 1 \geq 0 \tag{27}$$

$$g_5(\boldsymbol{x}) = 1 - \frac{n}{8} \geq 0 \tag{28}$$
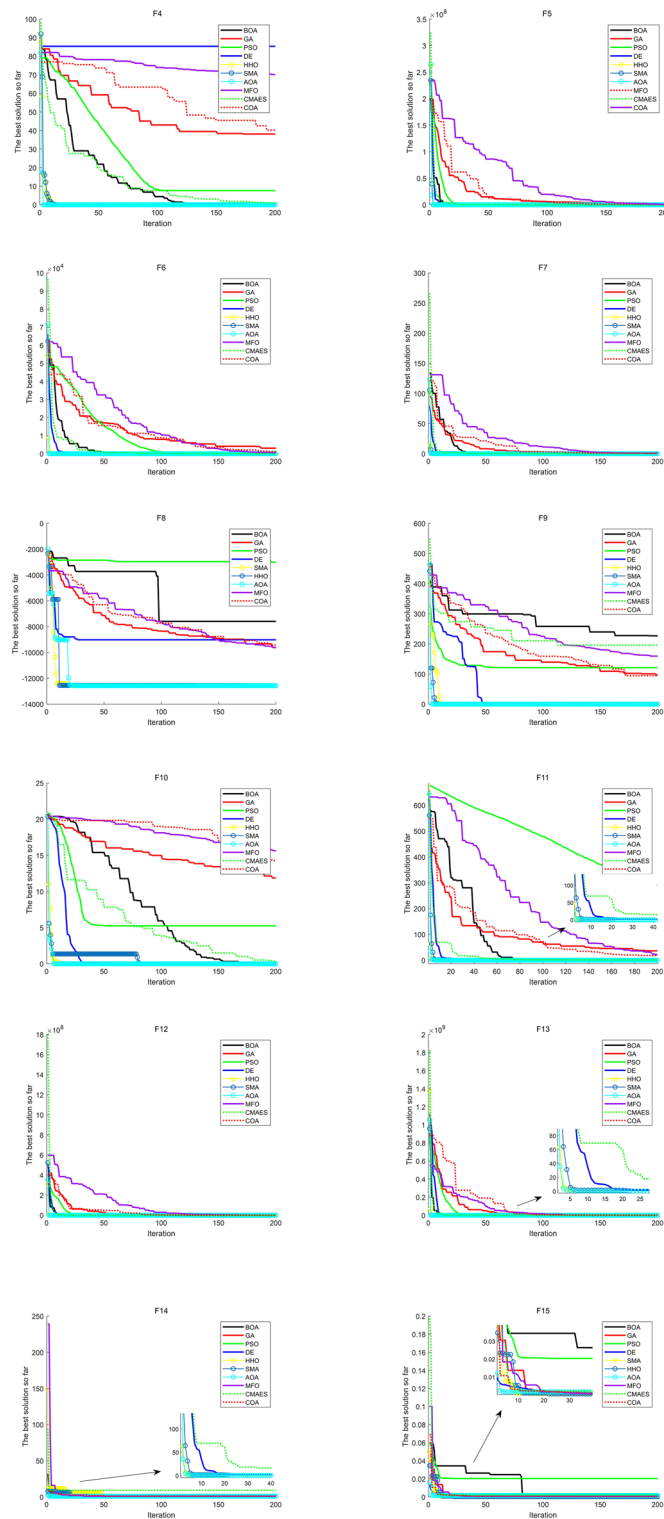
Figure 7. Convergence curves of benchmark functions. The figure shows the convergence speed of each algorithm by performing 200 iterations.

$$g_6(\boldsymbol{x}) = \frac{d_0}{5} - 1 \geq 0 \tag{29}$$

$$g_7(\boldsymbol{x}) = 1 - \frac{d_0}{18} \geq 0 \tag{30}$$
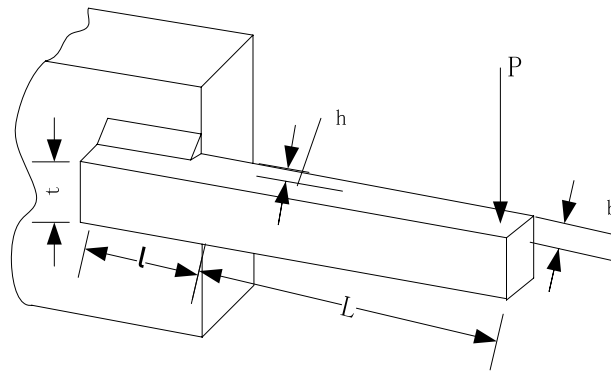
Figure 8. Welded beam problem. Model of welded beam.

Algorithm	Optimum variables				Optimum value
	h	**l**	**t**	**b**	
AOA	0.2056	3.9764	9.8395	0.2128	1.9965
PSO	0.2443	6.2059	8.3079	0.2453	2.3898
GA	0.2845	2.9364	7.9174	0.3462	2.4959
DE	0.2336	5.6579	9.6553	0.2766	2.8669

Table 8. Comparison of results for welded beam problem. Results of AOA and other different algorithms on welded beam problem.
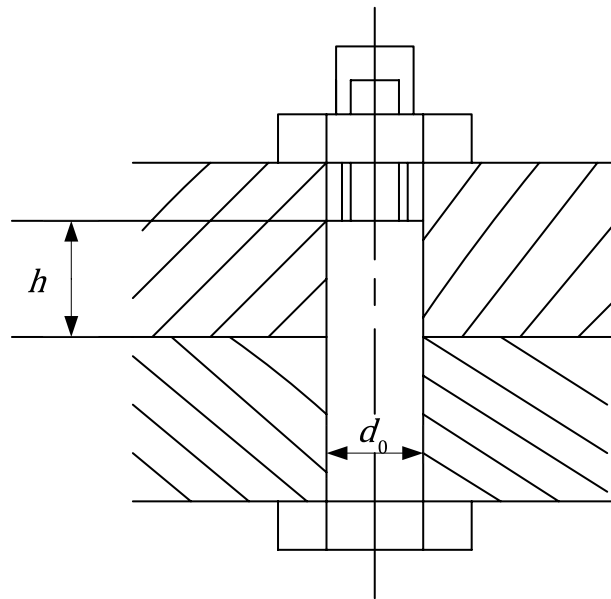
Figure 9. Bolt reliability analysis problem. Model of bolt.

The result of AOA is compared with DE, PSO, and GA as shown in Table 9. The bold values indicate the best one among all methods. As can be seen from the table, AOA outperforms all the other algorithms.

Conclusion

We proposed a new bio-inspiration based optimization algorithm called Aphid Optimization Algorithm (AOA) to solve optimization problems. The algorithm simulates the foraging process of aphids with wings, including the generation of winged aphids, flight mood, and attack mood. Concurrently, the corresponding optimization models have been presented according to the above phases. By comparing the performance of AOA with existing

| Algorithm | Optimum variables | | Optimum value |
	n	d_0	
AOA	7.8470	7.7437	0.94738
PSO	7.4601	7.9551	0.97646
GA	7.9536	10.0049	2.07096
DE	5.4063	9.4754	1.19582

Table 9. Comparison of results for reliability analysis problem. Results of AOA and other different algorithms on reliability analysis problem.

algorithms on 33 benchmark functions and two constrained engineering design problems, AOA algorithm has been shown its overall superiority. The presentation of AOA provides a new approach for solving optimization problems in the field of swarm intelligence optimization algorithms.

In future work, incorporating variation mechanisms and combinations with other algorithms will be considered to improve the performance of the algorithm. A future research direction also includes multi-objective AOA to solve multi-objective problems.

Data availability
All data generated or analyzed during this study are included in this article.

References
1. Agushaka, J. O., Ezugwu, A. E. & Abualigah, L. Dwarf mongoose optimization algorithm. *Comput. Method Appl. M.* **391**, 114570 (2022).
2. Blum, C. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv.* **35**, 268–308 (2003).
3. Abualigah, L., Elaziz, M. A., Sumari, P., Geem, Z. W. & Gandomi, A. H. Reptile search algorithm (rsa): A nature-inspired meta-heuristic optimizer. *Expert Syst. Appl.* **191**, 116158 (2022).
4. Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M. & Gandomi, A. H. The arithmetic optimization algorithm. *Comput. Method Appl. M.* **376**, 113609 (2021).
5. Zitouni, F., Harous, S., Belkeram, A. & Hammou, L. E. The archerfish hunting optimizer: A novel metaheuristic algorithm for global optimization. *Arab. J. Sci. Eng.* **47**, 2513–2553 (2021).
6. Wang, M., Li, B., Zhang, G. M. & Yao, X. Population evolvability: Dynamic fitness landscape analysis for population-based metaheuristic algorithms. *IEEE Trans. Evol. Comput.* **22**, 550–563 (2017).
7. Fan, X., Sayers, W., Zhang, S., Han, Z. & Chizari, H. Review and classification of bio-inspired algorithms and their applications. *J. Bionic. Eng.* **17**, 611–631 (2020).
8. Faris, H., Mafarja, M. M., Heidari, A. A., Aljarah, I. & Fujita, H. An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. *Knowl. Based Syst.* **154**, 43–67 (2018).
9. Salcedo-Sanz, S. Modern meta-heuristics based on nonlinear physics processes: A review of models and design procedures. *Phys. Rep.* **655**, 1–70 (2016).
10. Wolpert, D. H. & Macready, W. G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1**, 67–82 (1997).
11. Goldberg, D. E. & Bridges, C. L. An analysis of a reordering operator on a ga-hard problem. *Biol Cybern.* **62**, 397–405 (1990).
12. Das, S. & Suganthan, P. N. Differential evolution: A survey of the state-of-the-art. *IEEE Trans. Evol. Comput.* **15**, 4–31 (2011).
13. Abdulhameed, S. & Rashid, A. Child drawing development optimization algorithm based on child's cognitive development. *Arab. J. Sci. Eng.* **47**, 1337–1351 (2021).
14. Zhang, Y. & Jin, Z. Group teaching optimization algorithm: A novel metaheuristic method for solving global optimization problems. *Expert Syst. Appl.* **148**, 113246 (2020).
15. Moosavi, S. & Bardsiri, V. K. Poor and rich optimization algorithm: A new human-based and multi populations algorithm. *Eng. Appl. Artif. Intel.* **86**, 165–181 (2019).
16. Zhao, W., Wang, L. & Zhang, Z. Supply-demand-based optimization: A novel economics-inspired algorithm for global optimization. *IEEE Access.* **7**, 73182–73206 (2019).
17. Shabani, A., Asgarian, B., Salido, M. & Asil Gharebaghi, S. Search and rescue optimization algorithm: A new optimization method for solving constrained engineering optimization problems. *Expert Syst. Appl.* **161**, 113698 (2020).
18. Das, B., Mukherjee, V. & Das, D. Student psychology based optimization algorithm: A new population based optimization algorithm for solving optimization problems. *Adv. Eng. Softw.* **146**, 102804 (2020).
19. Faramarzi, A., Heidarinejad, M., Stephens, B. & Mirjalili, S. Equilibrium optimizer: a novel optimization algorithm. *Knowl. Based Syst.* **191**, 105190 (2020).
20. Kaveh, A., Khanzadi, M. & Rastegar Moghaddam, M. Billiards-inspired optimization algorithm: A new meta-heuristic method. *Structures.* **27**, 1722–1739 (2020).
21. Erol, O. K. & Eksin, I. A new optimization method: Big bang–big crunch. *Adv. Eng. Softw.* **37**, 106–111 (2006).
22. Formato, R. A. Central force optimization: A new metaheuristic with applications in applied electromagnetics. *Electromagn. Waves (Camb)* **77**, 425–491 (2007).
23. Rahmanzadeh, S. & Pishvaee, M. S. Electron radar search algorithm: A novel developed meta-heuristic algorithm. *Soft Comput.* **24**, 8443–8465 (2019).
24. Saryazdi, N. P. Gsa: A gravitational search algorithm. *Inf. Sci* **179**, 2232–2248 (2009).
25. Sang-To, T., Hoang-Le, M., Wahab, M. A. & Cuong-Le, T. An efficient planet optimization algorithm for solving engineering problems. *Sci. Rep.* **12**, 8362 (2022).
26. Kennedy, J & Eberhart, R. In Proceedings of ICNN'95 - International Conference on Neural Networks. 1942–1948 vol. 1944.
27. Arora, S. & Singh, S. Butterfly optimization algorithm: A novel approach for global optimization. *Soft Comput.* **23**, 715–734 (2018).
28. Aaha, B. *et al.* Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **97**, 849–872 (2019).

29. Li, S., Chen, H., Wang, M., Heidari, A. A. & Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* **111**, 300–323 (2020).
30. Abdullah, J. M. & Rashid, T. A. Fitness dependent optimizer: Inspired by the bee swarming reproductive process. *IEEE Access* **7**, 43473–43486 (2019).
31. Ovelade, O. N. & Ezugwu, A. E. Ebola Optimization Search Algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. in *International Conference on Electrical, Computer, and Energy Technologies, ICECET 2021* (Institute of Electrical and Electronics Engineers Inc., 2021). doi:https://doi.org/10.1109/ICECET52533.2021.9698813.
32. Abualigah, L. *et al.* Aquila optimizer: A novel meta-heuristic optimization algorithm. *Comput. Ind. Eng.* **157**, 107250 (2021).
33. Shayanfar, H. & Gharehchopogh, F. S. Farmland fertility: A new metaheuristic algorithm for solving continuous optimization problems. *Appl. Soft Comput.* **71**, 728–746 (2018).
34. Sulaiman, M. H., Mustaffa, Z., Saari, M. M. & Daniyal, H. Barnacles mating optimizer: A new bio-inspired algorithm for solving engineering optimization problems. *Eng. Appl. Artif. Intel.* **87**, 103330 (2020).
35. de Vasconcelos Segundo, E. H., Mariani, V. C. & dos Santos Coelho, L. Design of heat exchangers using falcon optimization algorithm. *Appl. Thermal Eng.* **156**, 119–144 (2019).
36. von Dohlen, C. D., Rowe, C. A. & Heie, O. E. A test of morphological hypotheses for tribal and subtribal relationships of aphidinae (insecta: hemiptera: aphididae) using dna sequences. *Mol. Phylogenet. Evol.* **38**, 316–329 (2006).
37. Johnson, C. G. Aphid migration in relation to weather. *Biol. Rev.* **29**, 87–118 (1954).
38. Kieckhefer, R. W., Lytle, W. F. & Spuhler, W. Spring movement of cereal aphids into south dakota. *Environ. Entomol.* **3**, 347–350 (1974).
39. Kring, J. B. Flight behavior of aphids. *Annu. Rev. Entomol.* **17**, 461–492 (1972).
40. Kaveh, A. & Zolghadr, A. Cyclical parthenogenesis algorithm for guided modal strain energy based structural damage detection. *Appl. Soft Comput.* **57**, 250–264 (2017).
41. Hartigan, J. A. & Wong, M. A. Algorithm as 136: A k-means clustering algorithm. *J. R. Stat. Soc. Ser. C Appl. Stat.* **28**, 100–108 (1979).
42. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl-Based Syst.* **89**, 228–249 (2015).
43. Pierezan, J., & Coelho, L. Coyote optimization algorithm: a new metaheuristic for global optimization problems. *2018 IEEE CEC.* 1–8 (2018).
44. Hansen, N. The cma evolution strategy: a tutorial. (2005).
45. Yao, X., Liu, Y. & Lin, G. M. Evolutionary programming made faster. *IEEE Trans. Evol. Comput.* **3**, 82–102 (1999).
46. Carrasco, J., García, S., Rueda, M. M., Das, S. & Herrera, F. Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review. *Swarm Evol. Comput.* **54**, 100665 (2020).

Acknowledgements

Author contributions

R.Y.L: Visualization, Investigation, Writing—Original draft, preparation Funding acquisition. N.Z: Conceptualization, Software, Writing—Original Draft. Y.F.Y: Visualization, Investigation. F.H.Y: Methodology, Supervision, Writing- Reviewing and Editing.

Competing interests

The authors declare no competing interests

Additional information

Correspondence and requests for materials should be addressed to F.Y.

Reprints and permissions information is available at www.nature.com/reprints.

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