Received 1 June 2022, accepted 22 June 2022, date of publication 27 June 2022, date of current version 30 June 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3186308

RESEARCH ARTICLE

Improved Adaptive Komodo Mlipir Algorithm

QIXUAN LIU[1] AND XIAOYI ZHANG[1,2]
[1]School of Biological and Agricultural Engineering, Jilin University, Changchun 130022, China
[2]Key Laboratory of Bionic Engineering, Ministry of Education, Jilin University, Changchun 130022, China
Corresponding author: Xiaoyi Zhang (xiaoyizhang@jlu.edu.cn)

ABSTRACT In order to improve the global search performance of the Komodo Mlipir Algorithm, this paper proposed two adaptive Komodo Mlipir Algorithms with variable fixed parameters (IKMA-1; IKMA-2). Among them, IKMA-1 adaptively controls the parthenogenesis radius of female Komodo dragons to achieve more efficient conversion of global search and local search. Second, IKMA-2 introduces adaptive weighting factors to the "mlipir" movement formula of Komodo dragons to improve the local search performance. Both IKMA-1 and IKMA-2 were tested on 23 benchmark functions in CEC2013 and compared with the other seven optimization algorithms. The Wilcoxon rank-sum test and Friedman rank test were used to compare the performance of different algorithms. Furthermore, IKMA-1 and IKMA-2 are applied to two constrained engineering optimization problems to verify the engineering applicability of the improved algorithm. The results show that both IKMA-1 and IKMA-2 have better convergence accuracy than the initial KMA. In terms of the benchmark function simulation results, IKMA-1 improves the performance by 17.58% compared to KMA; IKMA-2 improves by 10.99%. Both IKMA-1 and IKMA-2 achieve better results than other algorithms for engineering optimization problems, and IKMA-2 outperforms IKMA-1.

INDEX TERMS Komodo Mlipir Algorithm, variable fixed parameters, adaptive optimization, engineering design optimization problems.

I. INTRODUCTION

Establishing models to deal with practical problems is an essential means of current academic research, and how to solve models faster and better depends on the actual solution performance of various algorithms. An optimization algorithm is an application technology based on mathematics and used to solve various practical optimization problems. At present, optimization algorithms can be mainly divided into data processing algorithms, neural network algorithms, and swarm intelligence algorithms [1]. Applied optimization problems widely exist in many fields, such as signal processing, production scheduling, medical applications, image processing, and path planning.

However, since these optimization problems often involve discrete, discontinuous, and uncertain factors, it is not realistic to rely on a single algorithm to solve all optimization problems in life [2]. Therefore, the innovation of new algorithms and the improvement of existing algorithms are essential for solving practical optimization problems.

Looking back on the development of algorithms, due to the shortcomings of some classical optimization methods, such as the time-out of Newton's method in the face of complex mathematical processes, researchers are paying increasing attention to optimization algorithms inspired by nature. These algorithms include the genetic algorithm (GA) [3], differential evolution algorithm (DE) [4], immune algorithm (IA) [5], ant colony algorithm (ACO) [6], particle swarm algorithm (PSO) [7] and simulation annealing algorithm (SA) [8]. However, since the accuracy of these algorithms in solving practical problems cannot meet the actual needs of the current society, an increasing number of scholars have focused on the improvement and application of population optimization algorithms inspired by natural biological populations. Examples include the firefly algorithm (FA) [9], Harris-Hawk algorithm (HHO) [10], marine predator algorithm (MPA) [11], and slime mold algorithm (SMA) [12], and these swarm algorithms are used in practical problems. The solution accuracy and speed are satisfactory.

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

In addition, with the rapid development of hardware technology, the data processing capability is becoming increasingly more robust, and even very complex optimization algorithms can still be solved quickly. Therefore, how to develop or improve a new algorithm to obtain the global optimal solution more quickly and accurately has become one of the goals pursued by researchers [13]. The primary purpose of this paper is to adaptively improve the Komodo Mlipir Algorithm to better balance the development and exploration performance of KMA and enhance its ability and accuracy to solve practical optimization problems.

The main contributions of this study are as follows:

(1) The Komodo Mlipir Algorithm is adaptively improved, the two fixed parameters in the initial algorithm are replaced by adaptive parameters, and the adaptive weight factor is introduced for improvement. Then, two improved methods are proposed for the Komodo Mlipir Algorithm (IKMA-1; IKMA-2).

(2) The development and exploration performances of KMA, IKMA-1, IKMA-2 and other comparative algorithms were tested using the CEC2013 standard example, and the test results were examined for differences using the Wilcoxon signed-rank test.

(3) The proponents of the Komodo Mlipir Algorithm have not provided the verification of constrained engineering optimization problems, so this paper applies KMA, IKMA-1 and IKMA-2 to two engineering optimization problems and conducts supplementary research.

Compared with the KMA algorithm, IKMA-1 realizes the adaptive control of the female Komodo dragon lizard parthenogenetic radius α by introducing the search range variable, which strengthens the conversion efficiency of the algorithm's global search and local optimization search. Secondly, IKMA-2 changes the fixed parameters in the KMA algorithm to adaptive parameters by introducing an adaptive weight factor and further calculates the weight corresponding to the fitness of each Komodo dragon individual. Thus, the search space of the improved algorithm is expanded, and there is a greater probability of obtaining a better algorithm solution.

The rest of this paper is organized as follows: Section II presents the main categories and development overviews of intelligent optimization algorithms. Section III presents the mathematical description of the Komodo Mlipir algorithm and its adaptive variants, as well as the pseudocode of the algorithm. Then, in Section IV, the simulation calculation of KMA, IKMA-1, and IKMA-2 is carried out, and the difference between the calculation results and other algorithms is compared. Finally, Section V discusses the main conclusions and limitations of this study.

II. RELATED WORKS

Intelligent optimization algorithms are mainly divided into evolutionary algorithms, swarm intelligence optimization algorithms, and physical law algorithms [14]. Table 1 lists the related research on evolutionary algorithms and swarm intelligence optimization algorithms, including original algorithms and improved algorithm variants in recent years. The Komodo Mlipir Algorithm studied in this paper belongs to the swarm intelligence optimization algorithm [15]. However, it also has the characteristics of crossover mutation of the genetic algorithm, which has the characteristics of fast convergence speed and high precision.

A. EVOLUTIONARY ALGORITHMS

Traditional evolutionary algorithms mainly include genetic algorithms and differential evolution algorithms. However, due to the continuous complexity of optimization problems, the solution accuracy of traditional evolutionary algorithms cannot be satisfied, so variants based on evolutionary algorithms are produced. For example, the implementation of crossover, mutation, and selection operators due to genetic algorithm has many parameter limitations, such as crossover rate and mutation rate, and the choice of these parameters seriously affects the quality of the solution. Therefore, using an adaptive genetic operator to select an appropriate number of individuals with high fitness as parents and mutate the remaining individuals can reasonably achieve the purpose of balancing exploration and utilization [16]. In addition, redesigning three operators (sorting group selection, direction-based crossover, and typical mutation) to encode the genetic algorithm with real numbers [17] can also achieve calculation results similar to the adaptive improved genetic algorithm.

Compared with the genetic algorithm, the differential evolution algorithm has a stagnation phenomenon, making the algorithm stop prematurely and fall into the optimal local solution prematurely. In the existing research, the strategy of adaptive improvement is effective [18]. By updating parameters adaptively through individual similarity and evolutionary state and designing evolutionary backtracking strategies to control population diversity, evolutionary algorithms can be well prevented from falling into premature results [19].

B. SWARM INTELLIGENCE ALGORITHM

Current swarm intelligence algorithms are usually inspired by human intelligence, natural phenomena, biological group sociality, and other phenomena. They have attracted in-depth research by many scholars because of their diversity, robustness, and self-organization [20]. Academia generally defines swarm intelligence as "algorithms inspired by nature, such as human intelligence, biological swarm sociality, or the laws of natural phenomena, involving the emerging collective behavior of multiple interacting agents that follow some simple rules [21]."

For most swarm intelligence algorithms, it is found that there are some necessary connections between the algorithms, that is, the interaction between groups or the similarity of evolutionary operators, such as the traditional ant colony algorithm and particle swarm algorithm. However, with the increasing complexity of optimization problems, the simplification of coding and the precision and speed of the calculation results have gradually become the focus of researchers [22], so the demand for innovative and improved algorithms has arisen. For example, the search accuracy achieved by the Harris-Hawk optimization (HHO) algorithm is satisfactory in the application of scheduling problems [23]. However, there is still room for improvement in the simplification of coding. In addition, the arctanh function of the slime mold algorithm [12] will generate a series of warnings/errors in the coding process of the actual problem, and the use of the control formula in the sine cosine algorithm can make up for this defect well. It is outstanding in solving the EED problem [24].

C. LIMITATIONS AND IMPROVEMENTS OF THE KOMODO MLIPIR ALGORITHM

The original algorithm mainly explored in this paper is the Komodo Mlipir Algorithm, which was proposed by [15] in 2021. The swarm intelligence optimization algorithm is proposed by simulating the behavior between Komodo dragon lizard populations on Komodo Island. By simulating the "mlipir" gait behavior of large male Komodo dragon lizards and small Komodo dragon lizards, the algorithm itself has a higher probability of sampling in the global scope, and the population number is updated in real-time to achieve global adaptive control for the optimal solution. It can be found from the actual calculation example that the Komodo Mlipir Algorithm has the advantages of easy implementation, and high solution quality.

However, in the study of the example results of Komodo dragons, it can be found that the Komodo Mlipir Algorithm sometimes overestimates the output results when calculating the multimodal functions, which makes the algorithm stop early and makes the solution results deviate from the global optimum. Moreover, when exploring the movement behavior of small male Komodo dragons, it can be found that the original Komodo Mlipir Algorithm lacks adaptive control of parameters during parthenogenesis and is challenging to adapt to population changes. Therefore, improving the adaptive ability of the Komodo Mlipir Algorithm can better balance the ratio of development and exploration.

In addition, [15] also mentioned in the article that the "mlipir" activity probability in KMA is a fixed parameter value with a random distribution of [0, 1], which lacks adaptive balance ability, and the "mlipir" activity parameter affects the Komodo dragon. The interaction between individuals and the appropriate "mlipir" parameter can better control the ratio of the algorithm between the local search and the global search and then obtain the simulation results faster.

In the related literature on adaptive improvement, [30] adopted a dual adaptive weight strategy to optimize the whale optimization algorithm. They optimized the fixed linear inertia weight into a nonlinear local optimality function to balance the algorithm's global and local optimization. Reference [31] proposed an enhanced adaptive differential evolution algorithm by matching the appropriate fitness for each individual and sorting the fitness according to the crossover rate to maintain the population's diversity and strengthen the algorithm's global optimization ability. Therefore, inspired by the relevant literature, this study proposes an improved adaptive Komodo Mlipir Algorithm IKMA (IKMA-1 and IKMA-2) with variable fixed parameters. IKMA-1 realizes the adaptive control of the female Komodo dragon's parthenogenetic radius by introducing the search range variable. IKMA-2 introduces the adaptive weight factor to the Komodo dragon's "mlipir" movement formula. The improvement enhances the original algorithm's adaptability, realizes the adaptive search of the search space, strengthens the global optimization ability of the original algorithm, and reduces the possibility of falling into the local optimal solution.

III. METHODOLOGY

This chapter mainly introduces the Komodo Mlipir Algorithm (KMA) and the improved methods used in this paper.

A. KOMODO MLIPIR ALGORITHM

1) KOMODO DRAGON BEHAVIOR

Komodo dragons mainly refer to the dragon living on Komodo Island. The original KMA algorithm is an advanced algorithm proposed by simulating the unique foraging and breeding behavior of Komodo dragons.

For Komodo dragon populations, female Komodo dragons can produce offspring by mating with large adult males or by parthenogenesis. At the same time, small male Komodo dragons lack the ability to hunt prey and will eat by seeking out leftovers left by larger males. However, this behavior risks being eaten by large males, so they should keep their distance and look for opportunities to quickly approach the leftovers and eat when large males leave. We refer to this behavior as "mlipir" behavior, which is "walking on the edge to avoid danger" or "moving carefully along the side of the road, unnoticed by anyone, and successfully reaching a purpose safely."

2) DEMARCATION OF POPULATIONS

The population of Komodo individuals can be divided into three groups: high-quality large males, medium-quality females, and low-quality small male individuals. The corresponding division is shown in Equations (2.1) and (2.2); a population of n Komodo dragons can be divided into q large male individuals, one female individual, and s small Komodo dragon individuals.

$$q = \lfloor (p-1)\, n \rfloor \tag{2.1}$$

$$s = n - q \tag{2.2}$$

where the parameter p is a random value in the interval (0, 1), the value is usually taken as 0.5 to prevent the s and q values from converging to zero.

TABLE 1. A summary of some intelligent optimization algorithms and their variants, including their source of inspiration, key features of the algorithms, and the year they were proposed.

	Intelligent optimization algorithm	Inspiration or improvement ideas	Key Features	Year
Evolutionary Algorithms	Genetic Algorithm (GA)[3]	A process of biological evolution that simulates the natural selection and genetic mechanism of Darwin's theory of biological evolution	The problem-solving process is transformed through mathematical methods and computer simulation operations into a process similar to the crossover and mutation of chromosome genes in biological evolution.	1967
	Differential Evolution (DE)[4]	Simulate biological evolution through repeated iterations so that those individuals who adapt to the environment are preserved	The mutation vector is generated from the difference vector of the parent. It crosses with the individual parent vector to generate a new individual vector, which is directly selected with the parent individual.	1997
	Real-coded genetic algorithm (RCGA-rdn)[17]	An Improved Genetic Algorithm Based on Real Number Coding	Three specially designed operators are integrated: Ranked Group Selection (RGS), Direction-Based Crossover (DBX), and Normal Variation (NM). Moreover, a new step, called replacement operation, periodically initializes the population locally to increase the diversity of the population.	2019
	Multi-population adaptive genetic algorithm (MPAGA)[25]	Adaptively adjust the genetic algorithm's parameter values to maintain the algorithm population's diversity.	An alternative method was used for population assessment, and a migration operator was introduced. The exchange of information between populations takes place through the immigration process. Finally, the optimal solution is generated through multi-group co-evolution.	2020
	Adaptive Differential Evolution (ADE)[26]	Adaptively adjust the parameters of the difference algorithm.	Real-time storage of past control parameter values and indexing of individuals according to random numbers.	2020
	A backtracking differential evolution (bDE-MsAC)[19]	Multiple mutations and adaptive parameter control	Five modified mutation strategies were employed to construct a global exploration domain (GED) and a local development domain (LED). Then, a multi-mutation strategy autonomous cooperation mechanism was introduced to realize the co-evolution of GED and LED. Finally, the parameter adaptation scheme based on individual similarity and evolution state can update parameters adaptively, bringing vitality to the evolution process.	2021
Swarm Intelligence Optimization Algorithm	Ant Colony Algorithm (ACO)[6]	The behavior of ants to find paths in the process of finding food.	Ant colonies release pheromones during foraging behavior, and the concentration is inversely proportional to the path length. Therefore, the optimal foraging path will have the highest pheromone concentration after numerous iterations.	1991
	Firefly Algorithm (FA)[9]	flashing patterns and behavior of tropical fireflies	A given firefly will be attracted to other fireflies based on brightness, which can be proportional to the objective function value. Attractiveness and brightness both decrease as the distance between fireflies increases. The less bright one will move towards the brighter one for any two given fireflies.	2007
	Harris hawks optimization (HHO)[10]	The process of harris hawk preying on rabbits	According to the escape energy of the prey, the transition from the exploration stage to the development stage is realized. Moreover, choose four strategies according to the actual situation to simulate the attack state.	2019
	Slime mold algorithm (SMA)[12]	Modeling the foraging behavior of slime mold populations	The higher the food concentration, the greater the biological fluctuation of the bio-oscillator of the slime mold itself, the faster the cytoplasmic flow of the slime mold itself, and the thicker the venous route between the cells. As a result, slime molds have a higher probability of searching for areas with high food concentrations.	2020
	Marine Predators Algorithm (MPA)[11]	Interactions between marine predators and prey	Based on the movement type and velocity of prey, there is an optimal movement policy for a predator (Lévy or Brownian) in a way that maximizes its encounter rate with prey.	2020
	Adaptive ant colony algorithm[27]	The introduction of adaptive function	The information is first updated using an adaptive heuristic function for the shortest actual distance traveled by the ants. Then, the reward and punishment rules are introduced to optimize the local pheromone update strategy.	2020
	A staged adaptive firefly algorithm (SA-FA)[28]	Adaptively adjust the parameters of the FA.	The attractiveness model of the FA algorithm is improved, and three adaptive functions are used to adjust the relevant parameters.	2020
	IHHO[29]	Search Mechanism of Reinforcement Algorithm Using Elite Opposition Learning	Use elite adversarial learning to enhance the search mechanism, relying on mutation, mutation neighborhood search, and rollback strategies to improve solution search capabilities.	2020
	ISMA[24]	Introduction of Sine-Cosine Algorithm	The optimal solution is obtained by updating the position of the solution based on two equations borrowed from the Sine-Cosine Algorithm (SCA)	2021
	Komodo Mlipir Algorithm (KMA)[15]	Simulating reproduction and foraging behavior of Komodo dragon populations	Distinctive "mlipir" gait behavior of small Komodo dragons; bisexual reproductive characteristics of female Komodo dragons.	2021

3) LARGE MALE KOMODO DRAGON BEHAVIOR

According to a simple rule introduced in this study, larger males interact through attraction or distraction. Large males of low quality should be attracted to high-quality males, and the following two equations can be used to define the moving behavior of a large male Komodo dragon k_i and produce a new position k_i'.

$$w_{ij} = \begin{cases} r_1\left(k_j - k_i\right), & if\ f\left(k_j\right) < f\left(k_i\right)\ or\ r_1 < 0.5 \\ r_1\left(k_j - k_i\right), & otherwise \end{cases} \tag{2.3}$$

$$k_i' = k_i + \sum_{j=1}^{q} w_{ij}, \quad where\ j \neq i \tag{2.4}$$

where $f(k_i)$ and $f(k_j)$ are the fitness of the ith and jth large male Komodo dragons, respectively, with k_i representing the ith large male Komodo dragon; r_1 and r_2 are both random values between [0, 1]; and q represents the number of large male Komodo dragons. As a high-quality large male Komodo dragon will be attracted or distracted by a low-quality large male Komodo dragon, then the low-quality male Komodo dragon will be arranged to perform local optimality seeking. In contrast, the high-quality male Komodo dragon will also perform local optimality seeking or global optimality seeking behavior with a probability of 0.5, which indirectly ensures that the probability of local optimization is greater than that of global optimization.

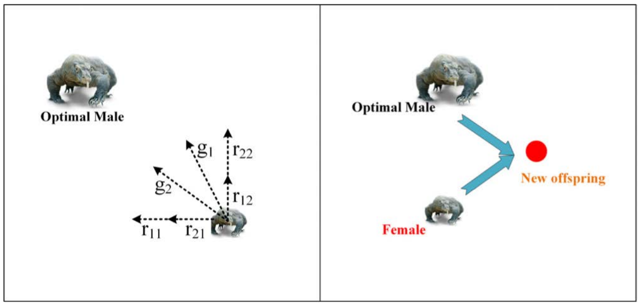

FIGURE 1. Females mating with optimal males. (With a normally distributed probability of 0.5, the female mates with the winner big male, and then, the female is updated by the best offspring.)

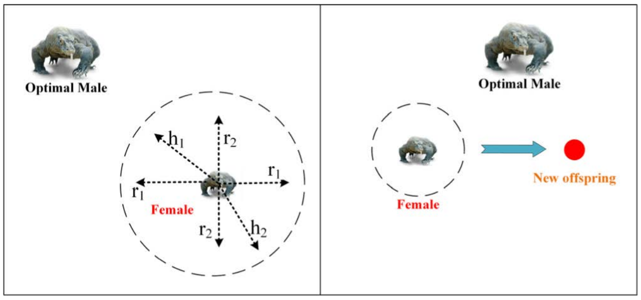

FIGURE 2. New offspring after parthenogenesis. (With a normally distributed probability of 0.5, the female undergoes parthenogenesis to generate an offspring, and then, the female is updated if the offspring is better than itself.)

4) REPRODUCTIVE BEHAVIOR OF FEMALE KOMODO GIANT DRAGONS

As shown in Fig. 1, in the original KMA design, female Komodo dragons were identified as medium-quality Komodo dragons, which also had the ability to search for superiority. If females chose to engage in local optimization activities, they would produce offspring by mating with the highest-quality male Komodo dragons, which would engage in global optimization activities through the foraging behavior of offspring Komodo dragons. In addition, parthenogenesis is also adopted by some female individuals, thus ensuring the diversity of solutions. The mating behavior of a female Komodo dragon is represented by the following formula:

$$\begin{cases} k'_{il} = r_l \cdot k_{il} + (1 - r_l)\, k_{jl} \\ k'_{jl} = r_l \cdot k_{jl} + (1 - r_l)\, k_{il} \end{cases} \tag{2.5}$$

where k_{il} and k_{jl} represent the ith male Komodo dragon and the jth female Komodo dragon individual in the lth dimension, respectively; meanwhile, k'_{il} and k'_{jl} represent the two offspring individuals in the kth dimension, respectively, where r_l is a random number in the lth dimension that lies within the interval [0, 1] showing a normal distribution.

Meanwhile, the process of parthenogenesis is represented by the following formula, which is achieved by appending a small value to each female dimension, where the small value is randomly generated using a symmetric normal distribution, and Figure 2 shows the process of parthenogenesis.

$$(k_{i1}, k_{i2}, \cdots, k_{im}) \rightarrow (k'_{i1}, k'_{i2}, \cdots, k'_{im}) \tag{2.6}$$

$$k'_{ij} = k_{ij} + (2r - 1)\alpha \left| ub_j - lb_j \right| \tag{2.7}$$

where ub_j and lb_j are the lower and upper bounds of the jth dimension, respectively, and $k_{i1}, k_{i2}, \cdots, k_{im}$ are the m-dimensional elements of the k-individual located within $[ub_j, lb_j]$; $r$is a random value of normal distribution; and α represents the radius of parthenogenesis, which is set to 0.1, that is, new solutions generated by the offspring can be carried out with a search radius of 10% in the search space.

5) MOVEMENT BEHAVIOR OF SMALL KOMODO DRAGON

As small Komodo dragons are attracted to the leftovers of large male Komodo dragons, two types of behavior exist for small Komodo dragons to survive: (1) seek out the leftovers left by large Komodo dragons whenever possible and (2) adopt a "mlipir" behavioral strategy to feed. Thus, there are two scenarios: first, a small Komodo dragon conducts "mlipir" activities around a large Komodo dragon to explore a wide area, and second, a small Komodo dragon moves straight toward the leftovers to achieve optimal local exploration. The probability of a small Komodo dragon adopting the "mlipir" activity is a random value of [0, 1]. The formula for the movement of the ith Komodo dragon following the jth Komodo dragon is shown below:

$$w_{ij} = \begin{cases} \sum_{l=1}^{m} r_1 \left(k_{jl} - k_{il}\right), & if\ r_2 < d \\ 0, & otherwise \end{cases} \tag{2.8}$$

$$k'_i = k_i + \sum_{j=1}^{q} w_{ij}, \quad where\ j \neq i \tag{2.9}$$

where r_1 is a random number normally distributed in the interval [0, 1] and represents the speed of movement of the small Komodo dragon; r_2 defines the dimension to be followed by the small Komodo dragon; m represents the dimension; k_{il} and k_{jl} represent the ith small male Komodo dragon in the lth dimension and the jth large male Komodo dragon, respectively; q represents the number of large male Komodo dragons; and l is a randomly selected dimension.

If two small Komodo dragons and two large male Komodo dragons are assumed to exist, then nine foraging behaviors of small Komodo dragons exist. If the movement behavior of small male Komodo dragons is counted separately, 18 forms of movement exist, and 12 are carried out in a wide search space, which fully ensures that the probability of global search is higher than the probability of local search.

6) POPULATION UPDATE

Since the population size n determines whether there are enough individuals for global optimization exploration, the original KMA proposes an adaptive method to control the population size. The formulas are as follows:

$$n' = \begin{cases} n - a, & if\ \delta f_1 > 0\ and\ \delta f_2 > 0 \\ n + a, & if\ \delta f_1 = 0\ and\ \delta f_2 = 0 \end{cases} \tag{2.10}$$

$$\delta f_1 = \frac{|f_1 - f_2|}{f_1}; \quad \delta f_2 = \frac{|f_2 - f_3|}{f_2} \tag{2.11}$$

Algorithm 1 Pseudo-Code for IKMA-1

Input: Dimension; PopSize; MaxNumEva
Output: Global optimum solution $\mathbf{k}_{best}$
Initialization: Set n, p, and d as the population of Komodo individuals, the portion of big males, and the "mlipir" rate, respectively;
Initialization of n individuals with m dimensions
while Stopping Criterion = false **do**
  for Calculate the fitness of all Komodo individuals **do**
  Rank all Komodo individuals;
    Based on their ranks and the portion p, split the population into three groups: q highest-quality big males, 1 middle-quality female, and s low-quality small males using Eq. (2.1) and Eq. (2.2);
  end for
  for Each big male **do**
    Move it using Eq. (2.4), and keep the q highest-quality big males to survive in the next generation;
    Update the female by either mating the winner big male using Eq. (2.5) or doing parthenogenesis using Eq. (2.6) and Eq. (2.7); control the parthenogenesis radius by Eq.(3.1);
  end for
  for Each small male **do**
    Move it using Eq. (2.8) and Eq. (2.9), and keep all their new positions to survive in the next generation;
    Update the population size n using Eq. (2.10);
  end for
end while
return Select the highest-quality Komodo from the three groups as the best-so-far solution $\mathbf{k}_{best}$

Algorithm 2 Pseudo-Code for IKMA-2

Input: Dimension; PopSize; MaxNumEva
Output: Global optimum solution $\mathbf{k}_{best}$
Initialization: Set n, p, and d as the population of Komodo individuals, the portion of big males, and the mlipir rate, respectively;
Initialization of n individuals with m dimensions
while Stopping Criterion = false **do**
  for Calculate the fitness of all Komodo individuals **do**
  Rank all Komodo individuals;
    Based on their ranks and the portion p, split the population into three groups: q highest-quality big males, 1 middle-quality female, and s low-quality small males using Eq. (2.1) and Eq. (2.2);
  end for
  for Each big male **do**
    Move it using Eq. (2.4), and keep the q highest-quality big males to survive in the next generation;
    Update the female by either mating the winner big male using Eq. (2.5) or doing parthenogenesis using Eq. (2.6) and Eq. (2.7);
  end for
  for Each small male **do**
    Move it using Eq. (3.4) and Eq. (2.9), and keep all their new positions to survive in the next generation;
    Update the population size n using Eq. (2.10);
  end for
end while
return Select the highest-quality Komodo from the three groups as the best-so-far solution $\mathbf{k}_{best}$

The size of the population is adjusted by the fitness between generations. If the fitness of two consecutive generations is optimized, the population size is reduced by removing the relevant individuals; if the fitness is not optimized, new individuals are generated to increase the population size to increase the probability of global optimization. Here, a is the number of population adjustments and δf_1 and δf_2 represent the differences between the ith- and $(i-1)$th-generation individuals.

B. IMPROVED ADAPTIVE KOMODO MLIPIR ALGORITHMS

1) IKMA-1 WITH AN ADAPTIVE THE PARTHENOGENESIS RADIUS

In the process of optimizing the Komodo Mlipir Algorithm, we refer to the search range update formula of the eagle perching optimization algorithm [32] as follows:

$$z = z * (eta + 1) \tag{3.1}$$

where z represents the population search range variable, which ensures that the hawk roosting optimization algorithm can achieve conversion between global search and local optimization search. In this paper, z is used to replace the radius α of female parthenogenesis in the original KMA to adaptively control the search radius of small- and medium-sized Komodo dragons in parthenogenesis. The initial setting is $z = 0.3$, where eta represents the shrinkage search variable, and the original $(eta + 1) \in [0, 1]$. In this paper, the eta parameter has been adjusted, and the specific eta calculation is as follows:

$$eta = \begin{cases} 0 & if\ \delta f_1 > 0\ and\ \delta f_2 > 0 \\ \frac{s}{n}, & if\ \delta f_1 = 0\ and\ \delta f_2 = 0 \end{cases} \tag{3.2}$$

where s is the number of small male Komodo dragons; n is the number of Komodo dragon populations; and δf_1 and δf_2 represent the differences between the ith- and $(i-1)$th-generation individuals.

The pseudo-code of the Adaptive Komodo Algorithm (IKMA-1) for variable parthenogenesis with fixed parameters is shown in Algorithm 1.

2) IKMA-2 WITH ADAPTIVE WEIGHTING FACTORS

In optimizing the Komodo Mlipir Algorithm, this paper introduces an adaptive weight factor [33], changes the fixed parameters in the original algorithm to adaptive parameters, and calculates the weight corresponding to the fitness of each Komodo dragon individual. The expression for the adaptive weight factor is:

$$\propto = \left| \frac{\mathrm{F_{worst}} - \mathrm{F_{il}}}{\mathrm{F_{worst}} - \mathrm{F_{best}}} \right| \tag{3.3}$$

where F_{worst} and F_{best} represent the fitness values of the worst and best individuals in this iteration respectively; and F_{il} represents the fitness value of the ith individual in the lth dimension.

Equation (2.8) is changed to the following formula:

$$w_{ij} = \begin{cases} \sum_{l=1}^{m} r_1(\propto_1 k_{jl} - \propto_2 k_{il}), & \textit{if } r_2 < d \\ 0, & \textit{otherwise} \end{cases} \quad (3.4)$$

The weight of an individual is determined by the size of the individual's own fitness; the higher the individual's fitness is, the greater the weight, and the stronger the reliability in Komodo dragon populations. r_1 still represents the rate of movement of an individual to increase the randomness of the local search of the algorithm.

The pseudo-code for the improved adaptive Komodo Mlipir Algorithm (IKMA-2) is shown in Algorithm 2.

IV. SIMULATION RESULTS

A. CEC2013 BENCHMARK FUNCTIONS

In this section, IKMA is tested and compared with other algorithms by using 23 benchmark functions (Tables 2 and 3) from the CEC2013 [34]. The fitness function values' average and standard deviation (STD) are used as evaluation metrics to compare the algorithms' merits. The description of the parameters for the CEC2013 benchmark algorithm is presented in Table 2 and Table 3. F1-F7 are single-modal benchmark functions that can effectively test the algorithm's convergence rate and local searchability and only one global optimal solution. F8-F13 are multimodal benchmark functions with multiple locally optimal solutions and one optimal global solution, which can effectively test the global search ability of the algorithm. F14-F23 are composite benchmark functions that can effectively try the algorithm exploitation ability and the balance search between the algorithm's performance [35].

B. EXPERIMENTAL RESULTS OF BENCHMARK FUNCTIONS

In this paper, the algorithms used for comparison include HHO, AOA, TSA, GWO, SMA, ESMA, KMA, IKMA-1, and IKMA-2. The test environment was MATLAB 2020b, and the hardware environment was a laptop computer with 2.5GHz, Intel(R) Core i5-7200U processor and 4GB RAM. Twenty independent operations were performed for each algorithm to overcome randomness, and the measured results and convergence curves are shown in Table 4 and Figure 3.

The fitness test results of IKMA-1, IKMA-2, and other algorithms are shown in Table 4, in which the bold experimental data are the best results of all comparative data. Firstly, the test results of the unimodal benchmark functions F1-F7 are discussed. For IKMA-1 and IKMA-2, the F1-F4 and F6 benchmark functions can achieve the optimal value, showing excellent global convergence performance. While for the benchmark function F5, IKMA-1 is slightly worse than HHO, SMA, and ESMA, the test results are still better than the original algorithm KMA; for F7, IKMA-1 is slightly inferior to AOA but better than IKMA-2, and both have similar performance.

Secondly, for the multi-peaked benchmark functions F8-F13. The F8-F11 functions, IKMA-1, and IKMA-2 achieve optimal global convergence and have excellent standard deviation performance. In contrast, for the F12 and F13 benchmark functions, IKMA-1 is second to HHO, and IKMA-2 is second to IKMA-1, with similar performance.

Finally, the composite benchmark functions F14-F23. F14-F16, IKMA-1 and IKMA-2 both achieve the best global convergence results among all the comparison algorithms, tied for first place, and the performance both are significantly better than KMA. In addition, for F17-F19, IKMA-1 and KMA are equivalent; the comparison results of F20-F23 benchmark functions can be seen that the optimization performance of IKMA-1 and IKMA-2 is similar to that of KMA, SMA, and ESMA. But IKMA- 1 on the F20-F22 function, with more stable variance results. By changing the fixed parameters to improve the Komodo Mlipir Algorithm, the balance exploration ability and global optimization ability of the original KMA can be significantly improved.

C. WILCOXON RANK-SUM TEST AND FRIEDMAN RANKING TEST

To compare the performance differences between the algorithms more intuitively, as suggested by [37], we need to perform a nonparametric Wilcoxon rank-sum test [36] between the results to determine whether the calculation results of IKMA and other compared algorithms are statistically significant. This study used the Wilcoxon rank-sum test at the $p = 0.05$ significance level to verify the difference in the test results. Finally, the Friedman ranking test [38] is performed on all algorithms. This test aims to more intuitively show the performance differences of the algorithm calculation results. The results of the Wilcoxon rank-sum test are shown in Table 5.

As shown in Table 5, IKMA-1 differs significantly from most other algorithms in most cases. Still, in the cases of F1, F3, F9, F10, F11, and F18, IKMA-1 and SMA measure the same results, i.e., they have similar optimization-seeking performance. For the test results of the two algorithms, IKMA-1 and KMA, eleven benchmark functions with the same results, all achieving the ideal global optimal result. While for the F5, F7, F8, F12, F13, F19, and F21 benchmark functions, the test results do not accept the original hypothesis at $p = 0.05$, i.e., the optimization performance of IKMA-1 and KMA is significantly different. Unfortunately, for the F16, F17, F22, and F23 benchmark functions, the original hypothesis is not rejected at $p = 0.05$, and therefore IKMA-2 performs similarly to the original KMA.

In addition, for the test results of the two improved algorithms, IKMA-1 and IKMA-2, only for F17, F22, and F23 benchmark functions, the original hypothesis is not rejected at $p = 0.05$, i.e., there is no significant difference in the performance of the two improved algorithms, and the searchability is similar.

Discussing the Wilcoxon rank-sum test results alone lacks intuition, so the Friedman ranking test was introduced for visual ranking. Furthermore, separate scales are performed according to different test function types, and finally, the

TABLE 2. Single-modal benchmark functions.

Name	Functions	dim	range
1	$f_1(x)=\sum_{i=1}^{n} x_i^2$	30	[-100,100]
2	$f_2(x)=\sum_{i=1}^{n} \lvert x_i\rvert+\prod_{i=1}^{n} \lvert x_i\rvert$	30	[-10,10]
3	$f_3(x)=\sum_{i=1}^{n}\left(\sum_{j-1}^{i} x_j\right)^2$	30	[-100,100]
4	$f_4(x)=max_i\{\lvert x_i\rvert, 1\le i\le n\}$	30	[-100,100]
5	$f_5(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(-1)^2]$	30	[-30,30]
6	$f_6(x)=\sum_{i=1}^{n}([x_i+0.5])^2$	30	[-100,100]
7	$f_7(x)=\sum_{i=1}^{n} ix_i^4+random[0,1]$	30	[-128,128]

average scale of all functions is given. As shown in Table 6, for single-peak and multi-peak test functions F1-F13, IKMA-1 has an average ranking of 3.1538, ahead of all the other algorithms IKMA-2 ranks second after IKMA-1. It can be seen that for single-peak as well as multi-peak test functions, IKMA-1 has a better global optimization ability.

D. WALL-CLOCK TIME ANALYSIS

In this part of the experiments, we compared KMA, IKMA-1, and IKMA-2 with the six participants to perform the experiments on the 13 benchmarks mentioned above. The time-consuming calculation [39] was performed by all participants independently running each function ten times and recording the results in Table 7. From the data in the table, it can be seen that the calculation of KMA, IKMA-1, and IKMA-2 in this study requires a relatively long time because the algorithm itself requires more computing power for crossover, mutation, and selection operations. However, IKMA can still outperform other algorithms in some cases, such as F1, F6, F11, F17, F19, and F23. It is also easy to calculate the time complexity of the KMA iterative process as $O(nm+nc+logn)$, where n, m, c, and $logn$ are the number of individuals in the population, the dimension, the calculation of the objective function and the ordering of fitness values, respectively. Overall, due to the limitations of KMA, IKMA takes longer but still has a huge effectiveness advantage, so the time results are expected.

E. EXPERIMENTS ON ENGINEERING DESIGN OPTIMIZATION PROBLEMS

The ultimate goal of algorithm design is to solve practical problems. Although the superiority of the adaptive optimization Komodo Mlipir Algorithm has been illustrated through the above benchmark functions, we still need to consider whether the algorithm has limitations in solving practical problems. For more practical problems, there will be many equality and inequality constraints. These constraints will cause the algorithm solution to be divided into feasible and infeasible solutions. To facilitate writing code that deals with infeasible solutions, this paper adopts the death penalty method. However, this will lose some of the valuable solution space obtained by the algorithm. In the following part of the article, we conduct experiments on two engineering design optimization problems, and a comparison with the current advanced algorithms is provided. Two engineering design optimization problems are the welded beam structure problem and the pressure vessel problem.

1) THE WELDED BEAM STRUCTURE PROBLEM

The Welded Beam Structural Problem (WBD) is a function minimization optimization problem in which the optimization algorithm is designed to reduce the manufacturing cost of the design. The optimization problem can be described as finding four design variables: Length (l), height (t), thickness (b), and weld thickness of beam bars(h); and they need to satisfy constraints such as shear stress (τ), bending stress (θ), beam bending load (Pc), end deviation (δ) and boundary conditions. The cost of manufacturing the welded beam is minimized, so the welded beam structure problem is a typical nonlinear programming problem. The mathematical description of the WBD problem is as follows:

Consider: $X=[x_1\ x_2\ x_3\ x_4]=[h\ l\ t\ b]$

Objective function:

$$F(X)=1.10471x_1^2x_2+0.04811x_3x_4(14.0+x_2)$$

Subject to:

$$g_1(X)=\tau(X)-\tau_{\max}\le 0$$
$$g_2(X)=\sigma(X)-\sigma_{\max}\le 0$$
$$g_3(X)=\delta(X)-\delta_{\max}\le 0$$

TABLE 3. Multimodal benchmark function.

Name	Functions	dim	range
8	$minf_8(x)=\sum_{i=1}^{n}-x_i sin\left(\sqrt{\lvert x_i\rvert}\right)$	n	[-500,500]
9	$minf_9(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$	n	[-5.12,5.12]
10	$minf_{10}(x)=-20\exp\left(-0.2\left(\frac{1}{n}\sum_{i=1}^{n}x_i^2\right)^{0.5}\right)-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))+20+e$	n	[-32,32]
11	$minf_{11}(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$	n	[-600,600]
12	$minf_{12}(x)=\frac{\pi}{n}\left\{10sin(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2[1+10sin^2(\pi y_{i+1})]+(y_n-1)^2\right\}$ $+\sum_{i=1}^{n}u(x_i,10,100,4), y_i=1+\frac{x_i+1}{4}$ $u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m x_i>a\\ 0-a<x_i<a\\ k(-x_i-a)^m x_i<a\end{cases}$	n	[-50,50]
13	$minf_{13}(x)=0.1\left\{sin^2(3\pi x_1)+\sum_{i=1}^{n}(x_i-1)^2[1+sin^2(3\pi x_i+1)]+(x_n-1)^2[1+sin^2(2\pi x_n)]\right\}$ $+\sum_{i=1}^{n}u(x_i,5,100,4)$	n	[-50,50]
14	$F_{14}(x)=(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6})^{-1}$	2	[-65,65]
15	$F_{15}(x)=\sum_{i=1}^{11}[a_i-\frac{x_1(b_i^2+b_1x_2)}{b_i^2+b_1x_3+x_4}]^2$	4	[-5,5]
16	$F_{16}(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	[-5,5]
17	$F_{17}(x)=(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6)^2+10(1-\frac{1}{8\pi})\cos x_1+10$	2	[-5,5]
18	$F_{18}(x)=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]$ $\times[30+(2x_1-3x_2)^2\times(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$	2	[-2,2]
19	$F_{19}(x)=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2)$	3	[1,3]
20	$F_{20}(x)=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2)$	6	[0,1]
21	$F_{21}(x)=-\sum_{i=1}^{5}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]
22	$F_{22}(x)=-\sum_{i=1}^{7}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]
23	$F_{23}(x)=-\sum_{i=1}^{10}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]

$$g_4(X)=x_1-x_4\le 0$$
$$g_5(X)=P-Pc(X)\le 0$$
$$g_6(X)=0.125-x_1\le 0$$
$$g_7(X)=1.10471x_1^2+0.04811x_3x_4(14.0+x_2)-5.0\le 0$$

where:

$$\tau(\vec{x})=\sqrt{(\tau')^2+2\tau'\tau''\frac{x_2}{2R}+(\tau'')^2}$$
$$\tau'=\frac{P}{\sqrt{2}\,x_1x_2},\quad \tau''=\frac{MR}{J},\quad \mathrm{M}=\mathrm{P}(\mathrm{L}+\frac{x_2}{2})$$
$$\mathrm{R}=\sqrt{\frac{x_2^2}{4}+(\frac{x_1+x_3}{2})^2}$$
$$J=2\{\sqrt{2}\,x_1x_2[\frac{x_2^2}{4}+(\frac{x_1+x_3}{2})^2]\}$$
$$\sigma(\vec{x})=\frac{6PL}{x_4x_3^2},\quad \delta(\vec{x})=\frac{6PL^3}{Ex_3^2x_4}$$
$$P_C(\vec{x})=\frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}(1-\frac{x_3}{2L}\sqrt{\frac{E}{4G}})$$

TABLE 4. Comparison results with other algorithms on the CEC 2013 benchmark function.

		HHO	AOA	TSA	GWO	SMA	ESMA	KMA	IKMA-1	IKMA-2
F1	AVG	5.159E-180	9.698E-66	2.083E-46	2.160E-59	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	STD	0.000E+00	2.909E-65	4.790E-46	1.567E-59	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F2	AVG	5.516E-98	0.000E+00	5.445E-29	1.109E-34	2.853E-156	2.336E-184	**0.000E+00**	**0.000E+00**	**0.000E+00**
	STD	1.413E-97	0.000E+00	6.508E-29	7.843E-35	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F3	AVG	1.394E-18	1.964E-03	1.268E-12	1.453E-14	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	STD	4.182E-18	3.048E-03	3.206E-12	2.632E-14	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F4	AVG	3.779E-89	6.550E-03	4.262E-03	1.243E-14	2.301E-197	6.640E-238	**0.000E+00**	**0.000E+00**	**0.000E+00**
	STD	1.133E-88	1.371E-02	6.169E-03	1.455E-14	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F5	AVG	**2.250E-03**	2.833E+01	2.871E+01	2.682E+01	7.813E+00	3.757E+00	4.831E+01	2.703E+01	1.486E+01
	STD	2.113E-03	1.541E-01	2.552E-01	7.119E-01	6.370E-01	8.436E+00	1.820E-01	4.404E-01	4.771E-01
F6	AVG	2.579E-05	2.732E+00	3.853E+00	6.382E-01	6.126E-03	2.018E-03	**0.000E+00**	**0.000E+00**	**0.000E+00**
	STD	3.560E-05	2.382E-01	7.892E-01	2.285E-01	4.150E-04	8.364E-04	0.000E+00	0.000E+00	0.000E+00
F7	AVG	8.724E-05	**2.485E-05**	6.182E-03	6.541E-04	1.747E-04	1.499E-04	1.715E-04	4.228E-05	1.179E-04
	STD	8.199E-05	2.359E-05	1.761E-03	3.989E-04	7.118E-04	1.294E-04	1.188E-04	2.946E-05	1.088E-04
F8	AVG	-1.257E+04	-5.815E+03	-6.217E+03	-6.004E+03	-1.257E+04	-1.257E+04	-9.016E+03	**-1.257E+04**	**-1.257E+04**
	STD	3.384E-01	5.962E+02	6.601E+02	8.053E+02	3.355E-01	1.575E-01	2.453E+03	1.839E+03	1.503E+03
F9	AVG	**0.000E+00**	**0.000E+00**	1.763E+02	2.274E-14	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	STD	0.000E+00	0.000E+00	4.037E+01	3.771E-14	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F10	AVG	8.882E-16	8.882E-16	1.912E+00	1.723E-14	8.882E-16	8.882E-16	8.882E-16	**8.882E-16**	8.882E-16
	STD	0.000E+00	0.000E+00	1.580E+00	3.256E-15	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F11	AVG	**0.000E+00**	9.773E-02	1.029E-02	1.604E-03	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	STD	0.000E+00	6.215E-02	1.050E-02	4.812E-03	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F12	AVG	**8.332E-07**	4.042E-01	8.506E+00	3.191E-02	2.989E-03	3.308E-03	2.799E-03	1.517E-03	3.393E-03
	STD	8.212E-07	4.679E-02	2.336E+00	1.585E-02	3.916E-03	4.855E-03	1.815E-03	8.682E-04	2.760E-03
F13	AVG	**2.568E-05**	2.794E+00	2.778E+00	4.775E-01	6.515E-03	3.603E-03	5.270E-02	5.807E-02	5.919E-02
	STD	3.050E-05	9.327E-02	4.251E-01	2.455E-01	6.834E-03	5.389E-03	4.689E-02	4.162E-02	8.151E-02
F14	AVG	1.097E+00	1.141E+01	7.153E+00	2.959E+00	9.980E-01	9.980E-01	9.980E-01	**9.980E-01**	**9.980E-01**
	STD	2.982E-01	3.482E+00	4.554E+00	3.370E+00	4.106E-13	4.106E-13	4.382E-16	0.000E+00	0.000E+00
F15	AVG	3.296E-04	6.183E-03	1.249E-02	4.319E-03	5.577E-04	5.610E-04	**3.075E-04**	**3.075E-04**	**3.075E-04**
	STD	1.690E-05	1.135E-02	9.883E-03	8.022E-03	2.502E-04	2.183E-04	5.878E-15	0.000E+00	0.000E+00
F16	AVG	-1.032E+00	-1.032E+00	-1.030E+00	-1.032E+00	-1.032E+00	-1.032E+00	-1.032E+00	**-1.032E+00**	**-1.032E+00**
	STD	1.046E-11	7.141E-08	7.890E-03	5.390E-09	2.342E-09	1.767E-09	9.779E-06	5.717E-07	8.856E-06
F17	AVG	3.979E-01	4.048E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	3.979E-01	**3.979E-01**	3.979E-01
	STD	8.932E-07	5.016E-03	3.887E-05	1.090E-04	1.239E-07	1.154E-07	3.514E-05	1.559E-05	3.795E-05
F18	AVG	3.000E+00	2.597E+01	2.100E+01	8.400E+00	3.000E+00	3.000E+00	**3.000E+00**	**3.000E+00**	**3.000E+00**
	STD	4.902E-08	2.034E+01	3.220E+01	2.020E+01	1.407E-10	3.194E-11	4.103E-10	0.000E+00	1.323E-14
F19	AVG	-3.862E+00	-3.853E+00	-3.862E+00	-3.861E+00	-3.863E+00	**-3.863E+00**	-3.861E+00	-3.861E+00	-3.862E+00
	STD	1.463E-03	2.510E-03	1.822E-03	2.922E-03	3.212E-07	3.053E-06	8.201E-04	4.430E-04	7.669E-04
F20	AVG	-3.164E+00	-3.090E+00	-3.265E+00	-3.252E+00	-3.254E+00	-3.252E+00	-3.320E+00	-3.320E+00	**-3.309E+00**
	STD	7.962E-02	8.092E-02	5.954E-02	7.727E-02	6.004E-02	5.955E-02	3.165E-04	2.591E-04	3.525E-02
F21	AVG	-5.727E+00	-3.695E+00	-6.091E+00	-1.015E+01	-1.015E+01	-1.015E+01	-1.015E+01	**-1.015E+01**	-1.015E+01
	STD	1.714E+00	9.349E-01	3.353E+00	2.393E-04	2.516E-04	2.196E-04	6.452E-15	0.000E+00	1.312E-08
F22	AVG	-5.438E+00	-4.176E+00	-6.866E+00	-1.040E+01	-1.040E+01	-1.040E+01	**-1.040E+01**	**-1.040E+01**	**-1.040E+01**
	STD	1.314E+00	1.360E+00	3.306E+00	4.982E-03	2.733E-04	2.454E-04	1.014E-05	5.120E-06	6.133E-06
F23	AVG	**-5.492E+00**	-3.717E+00	-5.287E+00	-1.054E+01	-1.054E+01	-1.054E+01	-1.054E+01	-1.054E+01	-1.054E+01
	STD	1.347E+00	1.275E+00	3.662E+00	2.146E-04	3.233E-04	2.128E-04	2.538E-06	2.113E-06	1.256E-06

Note: Bold data is the best result among all test results of the tested function;

TABLE 5. Comparison results on Wilcoxon rank sum test with algorithms.

	SMA	HHO	AOA	TSA	GWO	ESMA	KMA	IKMA-2
F1	NaN	6.386E-05	6.340E-05	6.386E-05	6.386E-05	NaN	NaN	NaN
F2	7.512E-04	6.386E-05	NaN	6.386E-05	6.386E-05	2.200E-03	NaN	NaN
F3	NaN	6.386E-05	6.386E-05	6.386E-05	6.386E-05	NaN	NaN	NaN
F4	2.313E-04	6.386E-05	6.386E-05	6.386E-05	6.386E-05	2.200E-03	NaN	NaN
F5	1.827E-04	1.827E-04	1.827E-04	1.717E-04	4.274E-01	1.827E-04	2.200E-03	2.498E-04
F6	6.386E-05	6.386E-05	6.386E-05	6.386E-05	6.386E-05	6.386E-05	NaN	NaN
F7	3.120E-02	8.900E-02	2.413E-01	1.827E-04	1.817E-04	9.097E-01	2.700E-03	4.848E-03
F8	6.386E-05	6.386E-05	1.827E-04	1.827E-04	1.827E-04	6.386E-05	2.350E-02	NaN
F9	NaN	NaN	NaN	6.386E-05	7.760E-02	NaN	NaN	NaN
F10	NaN	NaN	NaN	6.340E-05	4.040E-05	1.594E-05	NaN	NaN
F11	NaN	NaN	6.386E-05	2.200E-03	3.681E-01	NaN	NaN	NaN
F12	5.205E-01	1.827E-04	1.827E-04	1.827E-04	1.827E-04	3.120E-02	4.760E-02	1.780E-02
F13	1.700E-03	1.827E-04	1.827E-04	1.827E-04	1.700E-03	2.461E-04	2.979E-02	2.301E-02
F14	NaN	1.681E-01	1.282E-04	6.024E-05	1.430E-02	1.594E-05	NaN	NaN
F15	6.386E-05	6.386E-05	6.386E-05	6.340E-05	6.340E-05	6.340E-05	NaN	NaN
F16	3.081E-06	3.081E-06	3.576E-05	3.465E-04	3.502E-05	6.386E-05	4.260E-01	2.433E-02
F17	2.902E-05	3.347E-05	3.588E-05	7.603E-01	3.600E-03	1.720E-02	1.590E-01	1.489E-01
F18	NaN	4.100E-03	1.953E-05	1.953E-05	1.953E-05	NaN	NaN	NaN
F19	3.563E-05	6.000E-03	3.588E-05	3.465E-04	1.741E-01	6.386E-05	2.350E-02	1.242E-04
F20	3.465E-04	3.588E-05	3.588E-05	2.555E-01	8.029E-01	4.650E-01	1.041E-01	4.800E-02
F21	2.162E-04	1.953E-05	1.953E-05	1.945E-05	1.953E-05	1.594E-05	2.031E-02	1.837E-05
F22	3.347E-05	3.588E-05	3.588E-05	3.588E-05	3.539E-05	6.386E-05	6.239E-02	1.985E-01
F23	3.456E-04	3.588E-05	3.588E-05	3.588E-05	3.563E-05	6.386E-05	2.413E-01	1.697E-01

Note: Wilcoxon rank-sum test at the p=0.05 significant level; NaN represents the same test value.

TABLE 6. Comparison results on Friedman's ranking test with traditional algorithms.

Test problems	Metric	Optimization algorithm								
		HHO	AOA	TSA	GWO	SMA	ESMA	KMA	IKMA-1	IKMA-2
Scalable test problems F1-F13	Mean rank	3.8077	6.6538	8.4615	7.3077	4.1154	3.7308	4.2308	3.1538	3.5385
	rank	**4**	**7**	**9**	**8**	**6**	**3**	**5**	**1**	**2**
Non-scalable test problems F14-F23	Mean rank	5.7500	8.4500	6.8500	5.400	4.300	4.0500	3.6500	3.400	3.300
	rank	**7**	**9**	**8**	**6**	**5**	**4**	**3**	**2**	**1**
Total test problems F1-F23	Total mean rank	4.6957	7.4348	7.8043	6.5217	3.9565	3.8478	3.9565	3.2609	3.5217
	Total rank	**6**	**8**	**9**	**7**	**5**	**3**	**4**	**1**	**2**

TABLE 7. Wall-clock time costs of IKMA and other candidates on benchmark functions.

	SMA	HHO	AOA	TSA	GWO	ESMA	KMA	IKMA-1	IKMA-2
F1	3.1383	2.079693	1.5091	0.5131	1.533	4.2327	1.0835	0.92515	0.8138
F4	1.6524	0.72987	1.2708	0.4691	1.5729	3.362	1.0087	0.86823	1.1685
F6	2.0583	0.80207	0.97711	0.44686	1.284	3.1393	0.012631	0.014425	0.051098
F7	2.1854	1.3861	1.6596	0.75503	1.8079	3.833	8.0801	7.9075	9.9084
F8	1.634	0.83396	2.327	0.54792	1.5301	3.4075	6.199	7.7857	6.0875
F10	1.4345	0.61387	1.1997	0.34375	1.3874	3.1425	5.9987	6.9877	5.9987
F11	1.8966	1.4573	2.6187	1.6916	1.7124	3.6439	0.28419	0.56075	0.70454
F13	2.4259	2.8327	2.2765	1.2102	2.1701	4.4163	10.8126	30.3282	15.5274
F15	0.79184	0.45006	0.87206	0.19095	0.91191	2.0995	2.7895	10.8939	11.3085
F17	0.86093	0.78255	0.98151	0.67405	1.66	2.7041	0.12225	0.75548	0.11693
F19	0.7835	0.66953	1.0674	0.63853	1.6883	2.1952	0.1225	0.10488	0.17124
F21	0.82576	0.88504	1.2784	0.23763	1.1017	2.3618	2.9173	7.8958	2.3127
F23	0.93168	0.98053	1.3161	0.32363	1.3111	3.4281	0.35197	0.87278	1.0482

Note: The unit of measure in the table in seconds

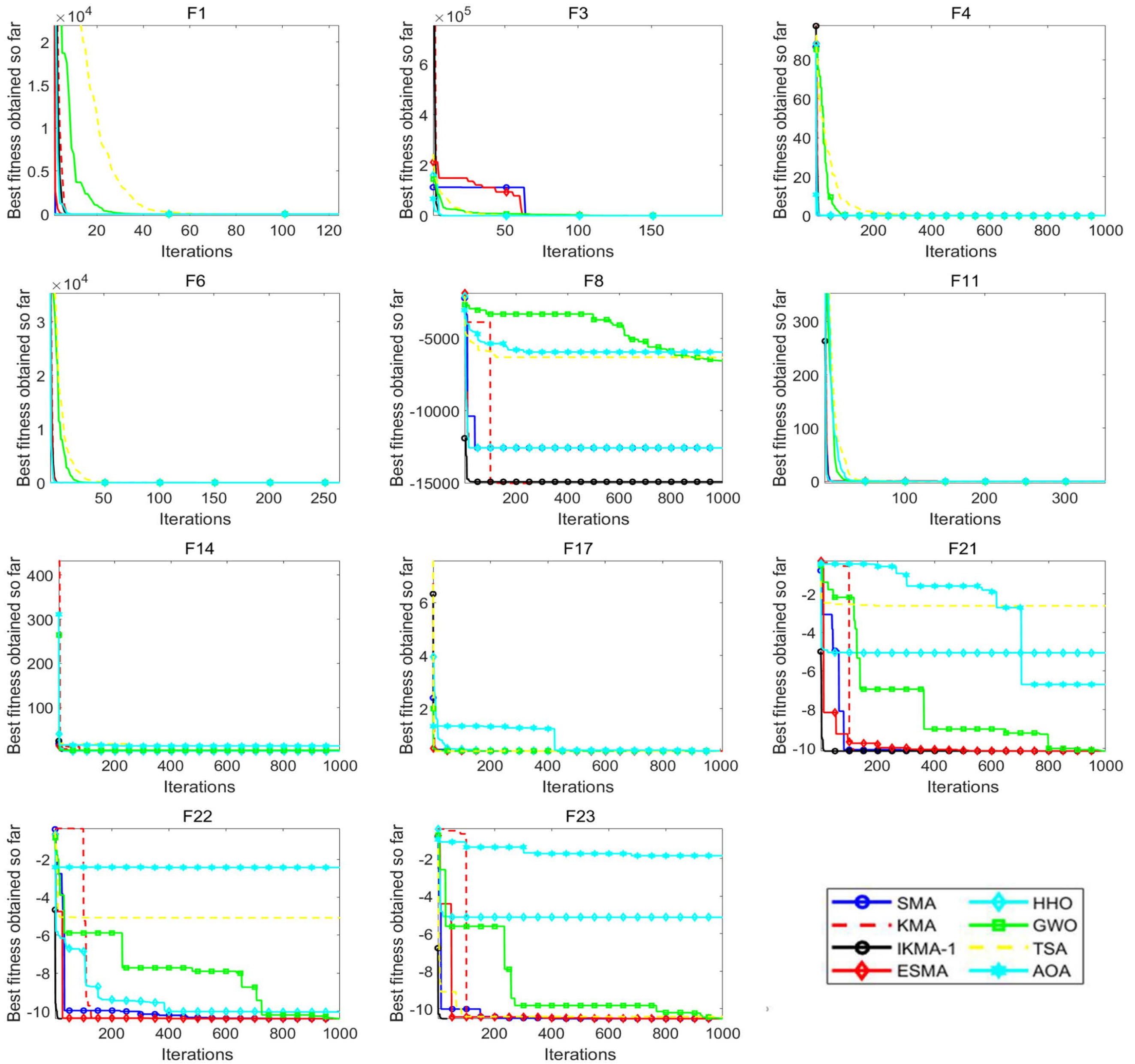

FIGURE 3. Convergence curves of the eight optimization algorithms which are used in our experiments. F1-F7 are single-modal benchmark functions; F8-F13 are multimodal benchmark functions; F14-F23 composite benchmark functions.

Variable ranges:

$$P = 6000lb; \quad L = 14in.; \quad \sigma_{max} = 30000psi;$$
$$E = 3 \times 10^6 psi;$$
$$G = 12 \times 10^6 psi; \quad \tau_{max} = 13600psi; \quad \delta_{max} = 0.25in.$$

In this section, IKMA is compared with SMA [12], MFO [40], GSA, WOA [42], BOA [43], GWO [44], BA [45] and Simplex [46]. From the data in Table 8, It can be seen that both IKMA-1 and IKMA-2 have superior convergence values to KMA, and IKMA-2 gives slightly better results than IKMA-1.

2) THE PRESSURE VESSEL DESIGN PROBLEM

The pressure vessel design problem (PVD) objective is to minimize the total cost f(x) while meeting production needs. The four design variables are shell thickness T_s (x_3), head thickness T_h (x_4), internal radius $R(x_1)$ and vessel length L (x_2 excluding head), where T_s and T_h are integer multiples of 0.625 and R and L are continuous variables. The specific mathematical model is as follows:

Consider:

$$X = [x_1 \;\; x_2 \;\; x_3 \;\; x_4] = [T_s \;\; T_h \;\; R \;\; L]$$

TABLE 8. Results of welded beam structure problem compared with other competitors.

Algorithms	Optimum values for variables				Optimum cost
	h	l	t	b	
IKMA-1	0.17421	2.75969	9.03154	0.20596	1.59237
IKMA-2	0.18203	2.68828	9.03532	0.20580	1.59132
KMA	0.17291	2.86342	9.05251	0.20565	1.60493
SMA	0.20540	3.25890	9.03840	0.20580	1.69604
MFO	0.20570	3.47030	9.03640	0.2057	1.72452
GSA	0.1821	3.85700	10.00000	0.2024	1.87995
WOA	0.20541	3.48430	9.03740	0.2063	1.73050
BOA	0.18797	2.70008	8.57988	0.23435	1.72084
GWO	0.2057	3.47840	9.03680	0.2058	1.72624
BA	0.20426	2.50395	9.22877	0.20479	1.61608
Simplex	0.2792	5.62560	7.75120	0.2796	2.53073

TABLE 9. Results of pressure vessel design problem compared with other competitors.

Algorithms	Optimum values for variables				Optimum cost
	T_s	T_h	R	L	
IKMA-1	0.78322	0.38851	40.3398	200.000	5936.48687
IKMA-2	0.79272	0.39171	41.07105	190.8629	5934.31828
KMA	0.78842	0.38473	40.32582	199.92878	5959.51166
SMA	0.79310	0.39320	40.67110	196.21780	5994.18570
HHO	0.81760	0.40730	42.09170	176.71960	6000.46260
PSO	2.61370	1.84250	61.22650	28.70790	24060.05370
MFO	0.81250	0.43750	42.09840	176.63660	6059.71430
CSS	0.81250	0.43750	42.10360	176.57270	6059.08880
GWO	0.81250	0.43450	42.08920	176.75870	6051.56390
WOA	0.83897	0.42272	43.24555	162.92238	6051.71279
Branch-bound	1.12500	0.62500	47.70000	117.70100	8129.10360

Objective function:

$$(X)_{\min} = 0.6224x_1x_3x_4 + 1.7781x_3x_1^2 + 3.1661x_4x_1^2 + 19.84x_3x_1^2$$

Subject to:

$$g_1(X) = -x_1 + 0.0193x_3 \leq 0$$
$$g_2(X) = -x_3 + 0.00954x_3 \leq 0$$
$$g_3(X) = -\pi x_4 x_3^2 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0$$
$$g_4(X) = x_4 - 240 \leq 0$$

Variable ranges:

$$0 \leq x_1 \leq 99, \quad 0 \leq x_2 \leq 99, \quad 10 \leq x_3 \leq 200, \quad 10 \leq x_4 \leq 200$$

In this section, IKMA is compared with SMA [12], HHO [10], PSO [41], [47], MFO [40], CSS [48], GWO, HPSO and Branch-bound [50]. As can be seen from Table 9, the convergence value of IKMA-2 is the best, followed by IKMA-1, showing the excellent searchable of KMA.

V. CONCLUSION

In this paper, two adaptive Komodo Mlipir Algorithms (IKMA-1 and IKMA-2) with variable fixed parameters are proposed. IKMA-1 uses the population search range variable to control the parthenogenesis of female Komodo dragons and the reproductive radius and then adaptively control the search space of small Komodo dragons. IKMA-2 introduces an adaptive weight factor to improve Formula (3.8), which strengthens the local search ability of the algorithm. To verify the effectiveness of the IKMA, 23 benchmark functions and two engineering design optimization problems are tested, and a comparison with other optimization algorithms is conducted. The results fully demonstrate the superior performance of IKMA. The following conclusions can be drawn from the experimental results:

(1) The IKMA is superior to the original KMA in solving continuity problems, especially for the unimodal benchmark function. Compared with KMA and other optimization algorithms, IKMA has a stronger global optimization ability and stable convergence speed.

(2) By adaptively controlling the fixed-value parthenogenesis radius, the flexible transformation of the original KMA in global search and local search can be enhanced, and the comprehensive performance of the algorithm can be effectively improved.

(3) For the constrained engineering design optimization problem, IKMA-2 has a better global optimal solution than IKMA-1, indicating that IKMA-2 has better exploration capability than IKMA-1.

Although the research results in this paper confirm the feasibility of adaptive optimization with variable fixed parameters, there are still some limitations in the research process. First, the test results of the benchmark function are not further calculated from multiple dimensions, and the influence of dimension changes on the algorithm simulation results is lacking. Second, although the method of using the death penalty in the solution of the constrained engineering optimization problem reduces the search space of the algorithm, it discards the possibility of finding a better solution. Finally, it is not convincing to use engineering optimization problems to test the applicability of the improved algorithm; thus, using actual optimization problems (scheduling problems, path planning problems, medical applications, etc.) to further test the ability of IKMA and KMA to solve discrete problems will be the next research step.

In addition, future researchers can adaptively optimize these two fixed parameters from another perspective. For example, when a learning mechanism is introduced, the idea of adaptive control algorithm parameters will have better results. In addition, the author of the Komodo Mlipir Algorithm also mentioned in the article that the fixed parameters of large male individuals can be improved.

These are effective research ideas, but they need further verification by researchers in the future.

ACKNOWLEDGMENT

The authors would like to thank the editors and anonymous reviewers whose insightful comments helped to improve the quality of this article and are very helpful in improving the authors' own academic level.

REFERENCES

[1] L. Angori, W. Didimo, F. Montecchiani, D. Pagliuca, and A. Tappini, "Hybrid graph visualizations with ChordLink: Algorithms, experiments, and applications," *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 2, pp. 1288–1300, Feb. 2022, doi: 10.1109/TVCG.2020.3016055.

[2] D. H. Wolper and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997, doi: 10.1109/4235.585893.

[3] M. Gen and L. Lin, *Genetic Algorithms*. Hoboken, NJ, USA: Wiley, 2008.

[4] H. A. Abbass and R. Sarker, "The Pareto differential evolution algorithm," *Int. J. Artif. Intell. Tools*, vol. 11, no. 4, pp. 531–552, Dec. 2002, doi: 10.1142/S0218213002001039.

[5] D. Wang, R. Y. K. Fung, and W. H. Ip, "An immune-genetic algorithm for introduction planning of new products," *Comput. Ind. Eng.*, vol. 56, no. 3, pp. 902–917, Apr. 2009, doi: 10.1016/j.cie.2008.09.036.

[6] Y. H. Song, C. S. Chou, and T. J. Stonham, "Combined heat and power economic dispatch by improved ant colony search algorithm," *Electr. Power Syst. Res.*, vol. 52, no. 2, pp. 115–121, Nov. 1999, doi: 10.1016/S0378-7796(99)00011-5.

[7] R. K. Ursem and P. Vadstrup, "Parameter identification of induction motors using stochastic optimization algorithms," *Appl. Soft Comput.*, vol. 4, no. 1, pp. 49–64, Feb. 2004, doi: 10.1016/j.asoc.2003.08.002.

[8] J. Pospichal and V. Kvasnicka, "Multistage decision-making using simulated annealing applied to a fuzzy automaton," *Appl. Soft Comput.*, vol. 2, no. 2, pp. 140–151, Dec. 2002, doi: 10.1016/S1568-4946(02)00058-3.

[9] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimization," *Int. J. Bio-Inspired Comput.*, vol. 2, no. 2, pp. 78–84, 2010, doi: 10.1504/IJBIC.2010.032124.

[10] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019, doi: 10.1016/j.future.2019.02.028.

[11] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: A nature-inspired Metaheuristic," *Expert Syst. Appl.*, vol. 152, Aug. 2020, Art. no. 113377, doi: 10.1016/j.eswa.2020.113377.

[12] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Gener. Comput. Syst.*, vol. 111, pp. 300–323, Oct. 2020, doi: 10.1016/j.future.2020.03.055.

[13] M. Noroozi, H. Mohammadi, E. Efatinasab, A. Lashgari, M. Eslami, and B. Khan, "Golden search optimization algorithm," *IEEE Access*, vol. 10, pp. 37515–37532, 2022, doi: 10.1109/ACCESS.2022.3162853.

[14] L. Goel, "An extensive review of computational intelligence-based optimization algorithms: Trends and applications," *Soft Comput.*, vol. 24, no. 21, pp. 16519–16549, Nov. 2020, doi: 10.1007/s00500-020-04958-w.

[15] S. Suyanto, A. A. Ariyanto, and A. F. Ariyanto, "Komodo mlipir algorithm," *Appl. Soft Comput.*, vol. 114, Jan. 2022, Art. no. 108043, doi: 10.1016/j.asoc.2021.108043.

[16] E. Shojaedini, M. Majd, and R. Safabakhsh, "Novel adaptive genetic algorithm sample consensus," *Appl. Soft Comput.*, vol. 77, pp. 635–642, Apr. 2019, doi: 10.1016/j.asoc.2019.01.052.

[17] Y. Song, F. Wang, and X. Chen, "An improved genetic algorithm for numerical function optimization," *Appl. Intell.*, vol. 49, no. 5, pp. 1880–1902, May 2019, doi: 10.1007/s10489-018-1370-4.

[18] Z. Zhang, Y. Cai, and D. Zhang, "Solving ordinary differential equations with adaptive differential evolution," *IEEE Access*, vol. 8, pp. 128908–128922, 2020, doi: 10.1109/ACCESS.2020.3008823.

[19] Y. Li, S. Wang, H. Liu, B. Yang, H. Yang, M. Zeng, and Z. Wu, "A backtracking differential evolution with multi-mutation strategies autonomy and collaboration," *Int. J. Speech Technol.*, vol. 52, no. 3, pp. 3418–3444, Feb. 2022, doi: 10.1007/s10489-021-02577-y.

[20] H. D. Phan, K. Ellis, J. C. Barca, and A. Dorin, "A survey of dynamic parameter setting methods for nature-inspired swarm intelligence algorithms," *Neural Comput. Appl.*, vol. 32, no. 2, pp. 567–588, Jan. 2020, doi: 10.1007/s00521-019-04229-2.

[21] M.-C. Su, J.-H. Chen, A. M. Utami, S.-C. Lin, and H.-H. Wei, "Dove swarm optimization algorithm," *IEEE Access*, vol. 10, pp. 46690–46696, 2022, doi: 10.1109/ACCESS.2022.3170112.

[22] J. Zhao, Z.-M. Gao, and H.-F. Chen, "The simplified Aquila optimization algorithm," *IEEE Access*, vol. 10, pp. 22487–22515, 2022, doi: 10.1109/ACCESS.2022.3153727.

[23] C. Li, J. Li, H. Chen, and A. A. Heidari, "Memetic Harris hawks optimization: Developments and perspectives on project scheduling and QoS-aware web service composition," *Expert Syst. Appl.*, vol. 171, Jun. 2021, Art. no. 114529, doi: 10.1016/j.eswa.2020.114529.

[24] M. H. Hassan, S. Kamel, L. Abualigah, and A. Eid, "Development and application of slime mould algorithm for optimal economic emission dispatch," *Expert Syst. Appl.*, vol. 182, Nov. 2021, Art. no. 115205, doi: 10.1016/j.eswa.2021.115205.

[25] A. K. Shukla, "Multi-population adaptive genetic algorithm for selection of microarray biomarkers," *Neural Comput. Appl.*, vol. 32, no. 15, pp. 11897–11918, Aug. 2020, doi: 10.1007/s00521-019-04671-2.

[26] Z. Zhang, Y. Cai, and D. Zhang, "Solving ordinary differential equations with adaptive differential evolution," *IEEE Access*, vol. 8, pp. 128908–128922, 2020, doi: 10.1109/ACCESS.2020.3008823.

[27] Y. Zheng, Q. Luo, H. Wang, C. Wang, and X. Chen, "Path planning of mobile robot based on adaptive ant colony algorithm," *J. Intell. Fuzzy Syst.*, vol. 39, no. 4, pp. 5329–5338, Oct. 2020, doi: 10.3233/JIFS-189018.

[28] L. Cheng, L. Zhong, X. Zhang, and J. Xing, "A staged adaptive firefly algorithm for UAV charging planning in wireless sensor networks," *Comput. Commun.*, vol. 161, pp. 132–141, Sep. 2020, doi: 10.1016/j.comcom.2020.07.019.

[29] R. Sihwail, K. Omar, K. Ariffin, and M. Tubishat, "Improved Harris hawks optimization using elite opposition-based learning and novel search mechanism for feature selection," *IEEE Access*, vol. 8, pp. 121127–121145, 2020, doi: 10.1109/ACCESS.2020.3006473.

[30] H. Chen, C. Yang, A. A. Heidari, and X. Zhao, "An efficient double adaptive random spare reinforced whale optimization algorithm," *Expert Syst. Appl.*, vol. 154, Sep. 2020, Art. no. 113018, doi: 10.1016/j.eswa.2019.113018.

[31] S. Li, Q. Gu, W. Gong, and B. Ning, "An enhanced adaptive differential evolution algorithm for parameter extraction of photovoltaic models," *Energy Convers. Manage.*, vol. 205, Feb. 2020, Art. no. 112443, doi: 10.1016/j.enconman.2019.112443.

[32] A. T. Khan, S. Li Senior, P. S. Stanimirovic, and Y. Zhang, "Model-free optimization using eagle perching optimizer," 2018, *arXiv:1807.02754*.

[33] Z. Shu, X.-J. Wu, P. Huang, H. Fan, Z. Liu, and F. Ye, "Multiple graph regularized concept factorization with adaptive weights," *IEEE Access*, vol. 6, pp. 64938–64945, 2018, doi: 10.1109/ACCESS.2018.2876880.

[34] I. Fister, J. Brest, A. Iglesias, A. Galvez, S. Deb, and I. Fister, "On selection of a benchmark by determining the Algorithms' qualities," *IEEE Access*, vol. 9, pp. 51166–51178, 2021, doi: 10.1109/ACCESS.2021.3058285.

[35] M. Jamil and X.-S. Yang, "A literature survey of benchmark functions for global optimisation problems," *Int. J. Math. Modeling Numer. Optim.*, vol. 4, no. 2, pp. 150–194, 2013.

[36] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945. [Online]. Available: http://www.jstor.org/stable/3001968

[37] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, Mar. 2011, doi: 10.1016/j.swevo.2011.02.002.

[38] J. Li, Y. Gao, K. Wang, and Y. Sun, "A dual opposition-based learning for differential evolution with protective mechanism for engineering optimization problems," *Appl. Soft Comput.*, vol. 113, Dec. 2021, Art. no. 107942, doi: 10.1016/j.asoc.2021.107942.

[39] F. M. Gartner, I. R. Graf, and E. Frey, "The time complexity of self-assembly," *Proc. Nat. Acad. Sci. USA*, vol. 119, no. 4, pp. 1–7, Jan. 2022, doi: 10.1073/pnas.2116373119.

[40] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015, doi: 10.1016/j.knosys.2015.07.006.
[41] N. H. Khan, Y. Wang, D. Tian, M. A. Z. Raja, R. Jamal, and Y. Muhammad, "Design of fractional particle swarm optimization gravitational search algorithm for optimal reactive power dispatch problems," *IEEE Access*, vol. 8, pp. 146785–146806, 2020, doi: 10.1109/ACCESS.2020.3014211.
[42] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, Feb. 2016, doi: 10.1016/j.advengsoft.2016.01.008.
[43] S. Arora, S. Singh, and K. Yetilmezsoy, "A modified butterfly optimization algorithm for mechanical design optimization problems," *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 40, no. 1, pp. 1–17, Jan. 2018, doi: 10.1007/s40430-017-0927-1.
[44] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.
[45] A. H. Gandomi, X.-S. Yang, A. H. Alavi, and S. Talatahari, "Bat algorithm for constrained optimization tasks," *Neural Comput. Appl.*, vol. 22, no. 6, pp. 1239–1255, 2013, doi: 10.1007/s00521-012-1028-9.
[46] K. M. Ragsdell and D. T. Phillips, "Optimal design of a class of welded structures using geometric programming," *J. Eng. Ind.*, vol. 98, no. 3, pp. 1021–1025, 1976, doi: 10.1115/1.3438995.
[47] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw. Conf.*, Nov. 1995, pp. 1942–1948.
[48] A. Kaveh and S. Talatahari, "A novel heuristic optimization method: Charged system search," *Acta Mech.*, vol. 213, nos. 3–4, pp. 267–289, Sep. 2010, doi: 10.1007/s00707-009-0270-4.
[49] Q. He and L. Wang, "A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization," *Appl. Math. Comput.*, vol. 186, pp. 1407–1422, Mar. 2007, doi: 10.1016/j.amc.2006.07.134.
[50] E. Sandgren, "Nonlinear integer and discrete programming in mechanical design optimization," *J. Mech. Des.*, vol. 112, no. 2, pp. 223–229, Jun. 1990, doi: 10.1115/1.2912596.

QIXUAN LIU received the B.E. degree in 2020. He is currently pursuing the master's degree with the School of Biological and Agricultural Engineering, Jilin University. His research interests include algorithm optimization and application, model simulation and application.

XIAOYI ZHANG received the B.S. and M.S. degrees in management and the Ph.D. degree in engineering from Jilin University, in 2006, 2008, and 2011, respectively. He is currently an Associate Professor with Jilin University and a Ph.D. Supervisor of agricultural economic management and systems engineering. His research interests include improvement of optimization algorithms, model simulation and application, technical economics, and supply chain management.