

Received February 23, 2021, accepted March 7, 2021, date of publication March 17, 2021, date of current version March 23, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3066180

CMSRAS: A Novel Chaotic Multi-Specular Reflection Optimization Algorithm Considering Shared Nodes

BING MA^{1,3}, PENGMIN LU¹, LUFAN ZHANG², QISONG QI^{3,4}, YIXIN CHEN^{1,3},
YONGTAO HU^{3,5}, MENGMEG WANG^{3,5}, AND GUOZHU WANG^{3,5}

¹Key Laboratory of Road Construction Technology and Equipment, Ministry of Education, Chang'an University, Xi'an 710064, China

²School of Mechanical and Electrical Engineering, Henan University of Technology, Zhengzhou 450001, China

³WEIHUA Group Company Ltd., Changyuan 453400, China

⁴Mechanical Engineering Department, Taiyuan University of Science and Technology, Taiyuan 030024, China

⁵School of Mechanical Engineering, Henan Institute of Technology, Xinxiang 453003, China

Corresponding author: Bing Ma (mmbingchd@163.com)


This work was supported in part by the National Natural Science Foundation of China under Grant 51705132, in part by the Training Program for Young Backbone Teachers of Henan University of Technology, the Science and the Education Department of Henan Province Natural Science Project under Grant 21A460006, in part by the Innovative Funds Plan of Henan University of Technology, Science and Technology Research Project of Henan Province under Grant 202102210061, in part by the Key Scientific Research Projects of Universities in Henan under Grant 19B460001 and Grant 212102210139, and in part by the Horizontal Project of Chang'an University under Grant220225200517.

ABSTRACT Specular reflection algorithm (SRA) was a single population meta-heuristic algorithm inspired by the physical function of mirror. However, similar to most of meta-heuristic algorithms, it had the disadvantages of weak population diversity, stagnation in local optimal and low convergence rate. In order to overcome these shortcomings, a chaotic multi-specular reflection optimization algorithm considering shared nodes (CMSRAS) was proposed by the combination of population strategy with shared node, improved Tent chaos strategy and Gaussian mutation strategy. Initially, a single population SRA was extended to the multi-population with shared node and the population was initialized by improved Tent chaos sequence to improve the quality of the initial solution and the population diversity, and to enhance the global search ability. Meanwhile, to strengthen the local search ability and the convergence accuracy, the Gaussian mutation and improved Tent chaotic disturbance strategy were introduced into SRA. And then the influence law and sensitivity analysis of the CMSRAS algorithm between the initial setting parameters were obtained based on the response surface method and the Sobol's method. Finally, compared with both 12 state-of-the-art algorithms and 8 well-known advanced algorithms, the performance of CMSRAS was evaluated on a comprehensive set of 32 benchmark problems. In addition, CMSRAS was applied to solve the complex optimization problems of engineering structure. The results demonstrated that proposed CMSRAS algorithm outperformed most competitive algorithms and efficiently solve the real-life global optimization problems.

INDEX TERMS Specular reflection algorithm, chaotic multi-specular reflection optimization algorithm, sharing nodes, sensitivity analysis, Tent chaos, Gaussian mutation, real-life global optimization problems.

I. INTRODUCTION

Nowadays, with the continuous development of science and technology, engineering design is developing in the direction of lightweight, green intelligent manufacturing, high reliability etc. Therefore, the optimal design of engineering

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

structures has become the focus of attention of many scholars. However, many engineering structural optimization problems are generally continuous, nonlinear, multi-dimensional and complex constraints. Traditional numerical optimization methods have low computational efficiency and low precision, so it is difficult to find global optimal solutions that meet the structural performance requirements. Due to the low computational efficiency and low accuracy of traditional

numerical optimization methods, it is difficult to find the global optimal solution to meet the requirements of structural performance.

In order to solve such complex engineering structure optimization design problems, humans simulate the behavior of natural biological groups from a biological point of view, and many metaheuristic swarm intelligence optimization algorithms are emerged and have been successfully applied to solve the complex optimization design problems of various engineering structures. Genetic algorithm (GA) was a robust stochastic global search optimization algorithm based on natural selection and natural genetics. Genetic algorithm (GA) was used to solve the shell structure optimization problem [1], flexure hinge mechanism optimization problem [2], path planning problem [3], scheduling problem [4] and the air conditioning fuzzy controller optimization problem [5], respectively. PSO, as a swarm intelligent optimization algorithm simulating the foraging behavior of birds, was widely used in solving design optimization problems of engineering structures [6], [7], BP neural network optimization problem [8], electromagnetics optimization problem [9], feature selection [10], and flexible job-shop scheduling optimization problem [11], respectively. Differential evolution algorithm (DE) was one of the most popular meta-heuristic algorithms with the characteristics of simple structure, fast convergence and strong robustness [12]. Based on the DE, the LIDDE was proposed to solve the social networks problem [13]. The memory-based global differential evolution (MGDE) algorithm was proposed to solve the dynamic economic dispatch problem [14]. The economic load dispatch (ELD) problem was efficiently solved by the ADE-MMS method [15]. The EFDE algorithm was used to solve the dynamic economic emission dispatch (DEED) problem [16]. The Firefly Algorithm (FA) mimicked the social behavior of fireflies based on their flashing characteristics and was used to solve the optimization problems of hybrid continuous/discrete structure [17] and the tracking architecture [18]. CS, which mimicked the brood parasitism behavior of cuckoos, combined with Levy flight, was a new type of intelligent optimization proposed by Gandomi AH *et al.* and widely used to solve structural nonlinear constrained optimization problems [19] and the discrete size optimization of composite steel-concrete box girders [20]. Yang X *et al.* proposed a BA based on the echolocation behavior of bats, which was applied to solve eight nonlinear engineering optimization problems, and achieved good optimization design results [21]. To solve the engineering combination optimization problems, simulated annealing algorithm (SA), simulating the physical annealing process of solid material, was put forward by Kirkpatrick *et al.* [22], [23]. Artificial bee colony (ABC) algorithm was inspired by the intelligent behavior of honey bee swarm to optimize the multivariable functions [24]. Based on the law of gravity and mass interactions, gravitational search algorithm (GSA) was proposed to solve various nonlinear functions [25]. By mimicking the leadership hierarchy and hunting mechanism of grey wolves, grey wolf

optimizer (GWO) was used to solve the classical engineering design problems [26]. Multi-Verse Optimizer (MVO) was inspired by the concepts of white hole, black hole, and wormhole to deal with both challenging test and real engineering problems [27]. Moth-Flame Optimization (MFO) algorithm mimicked the navigation method of moths in nature called transverse orientation [28]. Whale Optimization Algorithm (WOA) mimicked the social behavior of humpback whales which inspired by the bubble-net hunting strategy to solve structural design problem [29]. Based on a mathematical model based on sine and cosine functions, Sine Cosine Algorithm (SCA) was proposed to solve optimization problems [30]. By inspired the social interaction of dragonflies in navigating, searching for foods, and avoiding enemies, the dragonfly algorithm (DA) was used to solve optimization problems [31]. Based on the intelligent behavior of crows, a crow search algorithm (CSA) was successful to solve some classic engineering problems [32]. Salp swarm algorithm (SSA) was inspired by the swarming behavior of salps when navigating and foraging in oceans to efficiently solve engineering design problems [33]. Inspired by the behaviors of searching for prey, encircling, and attacking prey for spotted hyenas, Spotted Hyena Optimizer (SHO) was applied to deal with engineering design problems [34]. Seagull Optimization Algorithm (SOA) mimicked the migration and attacking behaviors of a seagull in nature to solve large-scale industrial engineering problems [35]. A butterfly optimization algorithm (BOA) was proposed, which mimicked the butterfly's food search and mating behavior and was used to solve three classic engineering problems (spring design welding beam design and gear train design) [36]. By the cooperative behavior and chasing style of Harris hawks in nature, Harris Hawks Optimizer (HHO) was used to solve several real-world engineering problems [37]. A new slime mould algorithm (SMA) based on the oscillation mode of slime mould in nature for solving the optimizing engineering problems [38]. Based upon the gradient-based Newton's method and utilize the GSR and LEO, gradient-based optimizer (GBO) was proposed in solving complex real-world engineering problems [39]. The heap-based optimizer (HBO) was proposed by the concept of CRH, which built on three pillars: the interaction between the subordinates and their immediate boss, the interaction between the colleagues, and self-contribution of the employees [40]. The Social Engineering Optimizer (SEO) algorithm was an effective meta-heuristic algorithm inspired by social engineering to solve some optimization problems [41]. The Turbulent Flow of Water-based Optimization (TFWO) algorithm was a state-of-the-art optimization algorithm, which mimicked whirlpools created in turbulent flow of water [42]. The Barnacles Mating Optimizer (BMO) algorithm inspired by the mating behavior of barnacles was proposed to solve optimization problems [43].

Nevertheless, on the one hand, there were differences between the proposed meta-heuristic algorithms, and these algorithms had the common characteristics: exploration and

exploitation behavior[44], [45]. The exploration stage was to search for the optimal solution in the global space randomly. Based on the exploration stage, the exploitation stage was to search for the accurate solution and to improve the accuracy of the optimal solution. However, there was often a certain contradiction between the two stages, how to balance the relationship between the two stages was a relatively difficult problem. On the other hand, another common characteristic of these popular meta-heuristic algorithms was that they couldn't guarantee that the global optimum could be found for all optimization problems which was confirmed by No Free Lunch Theorem (NFL) [46].

In order to be able to handle optimization issues, based on the physical function of mirror, a new specular reflection algorithm (SRA) was proposed by Qi. When the object was hidden, the object could be found by the mirror reflecting lights, and then the object was observed and displayed in the mirror. It was a non-population algorithm so that only a single solution was calculated in each iteration to search for the new solution, which could improve the computational efficiency. Compared with DE and MDE algorithm, SRA had unique advantages, with fewer set parameters and higher solution accuracy, which can also better deal with the complex engineering structure optimization problems [47]. SRA was used in the robust optimization design and reliability robust optimization design of crane girder structure, and it was proved that the algorithm can effectively deal with the structure robust optimization design [48]. Compared with PSO, FOA and SA, SRA was an effective and superior algorithm, but for solving high dimensional, multi-peak complex functions, it was easy to fall into local optima, and the convergence accuracy of SRA was lower [49]. At present, there is little research on SRA. Compared some traditional algorithms, it has advantages of fast convergence rate and higher solution accuracy. However, similar to other meta-heuristic algorithms, SRA has the disadvantages of single population, jumping into local optimum, low robustness, weak ergodicity etc. How to improve the performance of metaheuristic swarm intelligence optimization algorithms has been a hot issue problem by the way to the convergence accuracy, global search ability, the ability to resist local optimization etc. One way is to improve the performance of meta-heuristic algorithms by using shared concepts. To solve the problem of premature convergence frequently appearing in ABC algorithm and convergence slowly of ABC, ABC algorithm with sharing factor was proposed. The results showed that this algorithm had higher convergence property compared with ABC algorithm [50]. In order to improve the convergence accuracy and late search ability of crow search algorithm, a sharing mechanism was introduced into crow search algorithm, shared crow algorithm using multi-segment perturbation was proposed. The results showed that the comprehensive performance of this algorithm was better than other meta-heuristic algorithms [51]. A multi-subpopulation based on symbiosis and non-uniform Gaussian mutation salp swarm algorithm (MSNSSA) was proposed to the overcome the disadvantages

of slow convergence rate and low precision in salp swarm algorithm (SSA) [52]. Another way is to improve the performance of meta-heuristic algorithm by using chaos theory and Gaussian mutation. The ABC algorithm based on self-adaptive Tent chaos was proposed to improve the performance of ABC algorithm, and the results show that it had a better performance compared with ABC algorithm [53]. Chaos optimization algorithm based on Tent map was proposed with a fast search rate [54]. The gravitational search algorithm based on improved Tent chaos (ITC-GSA) was put forward, which could effectively overcome the shortcomings of the GSA's vulnerability to premature convergence and local optimization, and improve the algorithm's convergence speed and optimization accuracy [55]. A new improved population migration algorithm was proposed by adding Gaussian mutation and the steepest descent algorithms, and the simulation results demonstrated that the convergence rate and global convergence ability of population migration algorithm was improved [56].

Currently, to the best of our knowledge, there is no work proposed in literatures to improve the exploitation and the exploration of SRA. Therefore, in this paper, a novel chaotic multi-specular reflection optimization algorithm considering shared nodes (CMSRAS) was proposed by the combination of population strategy with shared node, improved Tent chaos strategy and Gaussian mutation strategy, respectively. The main contributions of this paper were summarized as follows:

(a) a novel chaotic multi-specular reflection optimization algorithm considering shared nodes (CMSRAS) was proposed.

(b) Combined with the RSM method and Sobol's method, the influence law and sensitivity analysis of the CMSRAS algorithm were attained.

The rest of this paper is organized as follows: SRA algorithm was described in detail in section 2. The proposed CMSRAS algorithm, the pseudocode and flow chart of CMSRAS algorithm, and computational complexity analysis of CMSRAS algorithm were described in detail in section 3, respectively. In section 4, the influence of algorithm parameters on the performance of CMSRAS algorithm and its sensitivity analysis were obtained. In section 5, the performance of CMSRAS algorithm was evaluated compared with other competitive meta-heuristic algorithms. In the last section, conclusions and prospects were presented.

II. DESCRIPTION OF SRA ALGORITHM

According to the physical laws of mirrors reflection, the SRA algorithm was proposed by Qi *et al.* [47]–[49]. The physical model of SRA was shown in Fig. 1. It was observed that SRA consists of eye, mirror and object, respectively. Through the reflection behavior and reversing function of the mirror, the search range of the solution was expanded to achieve the purpose of searching for the global optimal solution. For n dimensional minimum optimization problem, the detailed steps of SRA were obtained as follows:

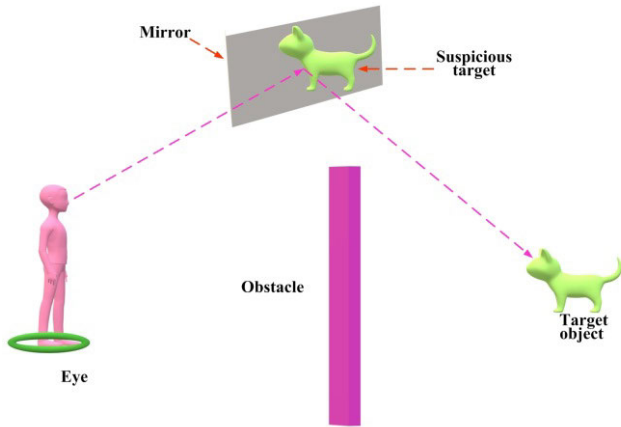


FIGURE 1. Physical model of SRA.

Step 1: In the independent variable feasible region, randomly generated three groups of initialization solution $X_i = [x_i^1, x_i^2, \dots, x_i^n]$, $i = 1, 2, 3$, and maximum number of iterations Max_I , search step factor c and convergence iteration accuracy ζ were determined, respectively.

Step 2: Evaluate of initial fitness value $f(X_i)$, and then sort it, when $f(X_1) \leq f(X_2) \leq f(X_3)$, let $X_1 = X^{eye}$, $X_2 = X^{mirror}$, $X_3 = X^{object}$.

Step 3: Updated the optimal solution position $x_{New_1}^n$ and $x_{New_2}^n$ by (1), respectively.

$$\begin{cases} x_{New_1}^n = x_{eye}^n + c \cdot r (x_{eye}^n - x_{object}^n) \\ x_{New_2}^n = x_{eye}^n + c \cdot r (x_{eye}^n - x_{mirror}^n - x_{object}^n) \end{cases} \quad (1)$$

where c was search step factor and r was random number.

Step 4: Updated the new optimal solution x_{New}^n by (2).

$$\begin{cases} x_{New}^n = x_{New_1}^D & \text{if } f(x_{New_1}^D) \leq f(x_{New_2}^D) \\ x_{New}^n = x_{New_2}^D & \text{if } f(x_{New_1}^D) \geq f(x_{New_2}^D) \end{cases} \quad (2)$$

Step 5: If it met the constraints of ζ or Max_I , output the global optimum x_{New}^n otherwise go to step 2, and continue to calculate the global optimal optimum. The flow chart of SRA was shown in Fig. 2.

III. THE PROPOSED CMSRAS ALGORITHM

SRA is an efficient and general random search optimization algorithm with high global search capability. However, it has the disadvantages of single population, slow convergence in the later stage, easily fall into local optimum. In order to overcome the shortcomings of SRA, the CMSRAS algorithm is proposed based on the multi-population strategy with shared node, improved Tent chaos strategy and Gaussian mutation strategy, respectively.

A. IMPROVEMENT STRATEGIES

(I) Multi-population strategy with shared node

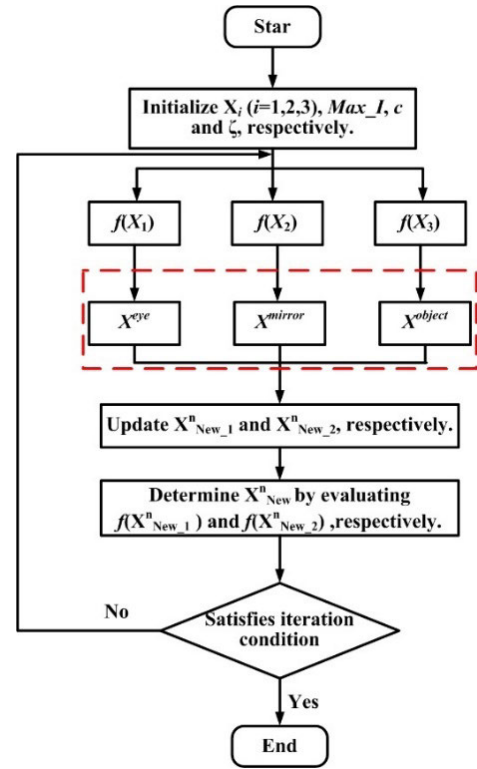


FIGURE 2. The flow chart of SRA.

SRA has a single population and weak population diversity, which easily leads to low ability of jumping out of local optimum and global optimum search. To enrich the population diversity of SRA, the single SRA model is modified into a multi-specular reflection optimization algorithm model based on a shared node, as shown in Fig. 3. The same as SRA is that its main components are eyes, suspicious targets, mirrors and target. Inversely, in the CMSRAS model, the population of eyes, mirrors, and suspicious targets are expanded, respectively.

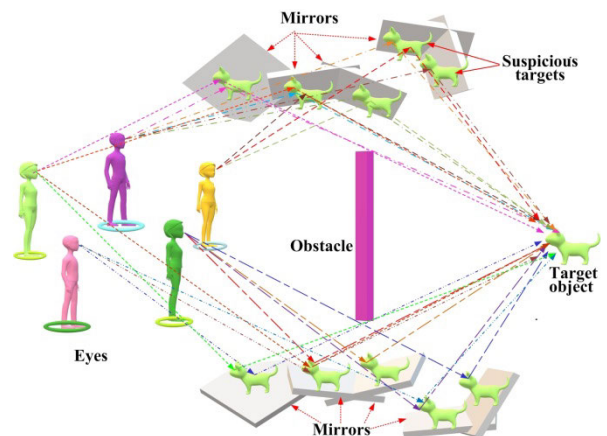


FIGURE 3. Physical model of CMSRAS.

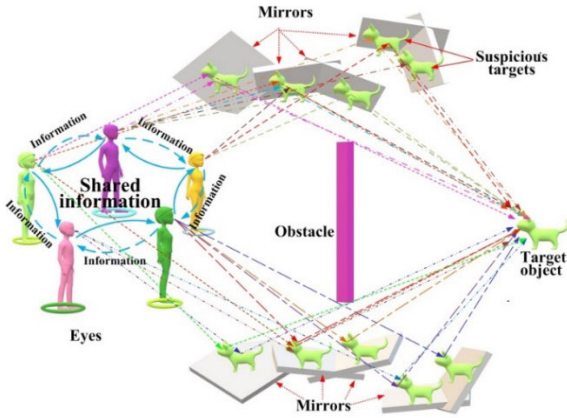


FIGURE 4. Information sharing of CMSRAS.

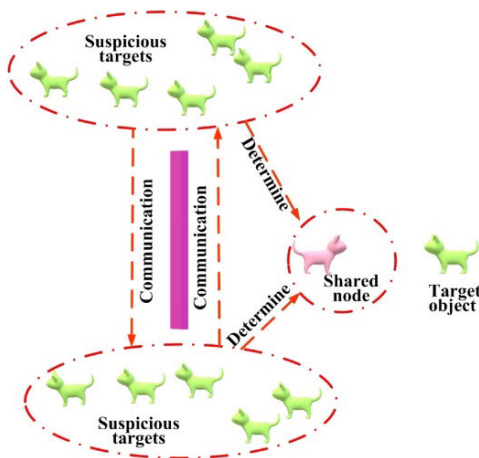


FIGURE 5. Shared node selection.

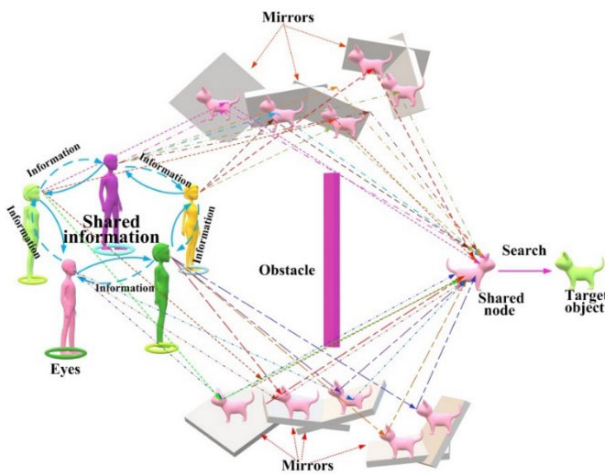


FIGURE 6. The iterative process of CMSRAS by starting with the shared node.

A shared node is obtained by the selecting the suspicious targets according to the survival of the fittest, and to improve the efficiency of the algorithm. The specific selection process of shared nodes is shown in Fig. 4 to Fig. 6. And then a

shared node is obtained by the selecting the suspicious targets according to the survival of the fittest, and to improve the efficiency of the algorithm. The specific selection process of shared nodes is shown in Fig. 4 to Fig. 6. In Fig. 4, the number of eyes is n , and it is the same as the number of mirrors. Different eyes can find the different target objects through the reflection function of different mirrors. Suspicious targets are made up of the different target objects, which are close to the real target objects. The number of suspicious targets is also n . The specular reflection rings are composed of eyes, mirrors and suspicious targets. Eyes can use the specular reflection ring to continuously search for the target, through the reflection and reversing behavior of the mirrors. However, this search method has the disadvantages of a certain blind selectivity and low efficiency and stability, because it is unable to accurately determine which suspicious target is closest to the real target. In order to further improve the efficiency of CMSRAS algorithm and ensure that each search is in the best direction, eyes and mirrors should have self-learning experiences (i.e. searching information such as the best viewing angle, environmental visibility, lighting conditions, mirror size, mirror reflection path, etc.) and share them with each other. And then through the way of information sharing between eyes and mirrors, the optimal location of eyes and mirrors is selected to determine the best suspected target, namely shared node. In Fig. 5, the number of suspicious targets is n , which are obtained by the Fig. 4. According to the law of survival of the fittest, shared node is determined by the way of the exchange and learning among n suspect targets. And the specular reflection rings with shared node are composed of shared node, eyes and mirrors, respectively. In Fig. 6, when the share node is determined, and then the specular reflection rings with shared node are also obtained. The number of specular reflection rings with shared node is n , and eyes can use the specular reflection rings with shared node to continuously search for the global optimal solution, through the reflection and reversing behavior of the mirrors. In the search process, the specular reflection rings with shared node are continuously updated, namely eyes, mirrors and shared node are continuously updated. The detailed steps were obtained as follows:

Step 1: The population number of the specular reflection ring n , the maximum iterations m and iterative precision ξ is defined, respectively.

Step 2: Randomly search for $2n + 1$ initial variable value in the feasible region $x_i(i= 1, 2, \dots, 2n+1)$.

Step 3: Calculate the fitness value of the objective function for each feasible solution $f(x_i)$, ($i= 1, 2, \dots, 2n+1$). According to the fitness value $f(x_i)$, select the optimal fitness value as a shared node, the median fitness value as $eye(i)(i= 1, 2, \dots, n)$ and the worst fitness value as $Mirror(i)(i= 1, 2, \dots, n)$ for n specular reflection rings. Meanwhile, the shared node and determination rules of specular reflection ring are shown in Table 1 in detail.

TABLE 1. The shared node and determination rules of specular reflection ring.

ID	Judgement conditions	Shared node	Mirrors	Eyes
1	$if\ f(x_{2n-1}) \leq f(x_{2n}) \leq f(x_{2n+1})$	$x_{Shared\ node}^n = x_{2n-1}$	$x_{Mirror}^n = x_{2n}$	$x_{eye}^n = x_{2n+1}$
2	$if\ f(x_{2n+1}) \leq f(x_{2n}) \leq f(x_{2n-1})$	$x_{Shared\ node}^n = x_{2n+1}$	$x_{Mirror}^n = x_{2n}$	$x_{eye}^n = x_{2n-1}$
3	$if\ f(x_{2n-1}) \leq f(x_{2n+1}) \leq f(x_{2n})$	$x_{Shared\ node}^n = x_{2n-1}$	$x_{Mirror}^n = x_{2n+1}$	$x_{eye}^n = x_{2n}$
4	$if\ f(x_{2n}) \leq f(x_{2n+1}) \leq f(x_{2n-1})$	$x_{Shared\ node}^n = x_{2n}$	$x_{Mirror}^n = x_{2n+1}$	$x_{eye}^n = x_{2n-1}$
5	$if\ f(x_{2n}) \leq f(x_{2n-1}) \leq f(x_{2n+1})$	$x_{Shared\ node}^n = x_{2n}$	$x_{Mirror}^n = x_{2n-1}$	$x_{eye}^n = x_{2n+1}$
6	$if\ f(x_{2n+1}) \leq f(x_{2n-1}) \leq f(x_{2n})$	$x_{Shared\ node}^n = x_{2n+1}$	$x_{Mirror}^n = x_{2n-1}$	$x_{eye}^n = x_{2n}$

TABLE 2. The updated rules of specular reflection rings and shared node.

ID	Judgement conditions	Shared node	Mirrors	Eyes
1	$if\ f(x_{New}^n) \leq f(x_{Shared\ node}^n) \leq f(x_{Mirror}^n)$	$x_{Shared\ node}^n = x_{New}^n$	$x_{Mirror}^n = x_{Shared\ node}^n$	$x_{eye}^n = x_{Mirror}^n$
2	$if\ f(x_{Shared\ node}^n) \leq f(x_{New}^n) \leq f(x_{Mirror}^n)$	$x_{Shared\ node}^n = x_{Shared\ node}^n$	$x_{Mirror}^n = x_{New}^n$	$x_{eye}^n = x_{Mirror}^n$
3	$if\ f(x_{Shared\ node}^n) \leq f(x_{Mirror}^n) \leq f(x_{New}^n)$	$x_{Shared\ node}^n = x_{Shared\ node}^n$	$x_{Mirror}^n = x_{Mirror}^n$	$x_{eye}^n = x_{New}^n$

Step 4: Based on the shared node, (1) is modified by (3). And then new solutions $x_{New_1}^n$ and $x_{New_2}^n$ are obtained by (3) and (4), respectively.

$$\begin{aligned}
 X_{New_1}^n &= x_{Shared\ node}^n + c_1 r_1 (x_{Mirror}^n - x_{eye}^n) \\
 &\quad + c_2 r_2 (x_{Shared\ node}^n - x_{Mirror}^n) \\
 &\quad + c_3 r_3 (x_{Shared\ node}^n - x_{eye}^n) \quad (3)
 \end{aligned}$$

$$\begin{aligned}
 X_{New_2}^n &= x_{Shared\ node}^n + c_1 r_1 (x_{Mirror}^n - x_{eye}^n) \\
 &\quad + c_2 r_2 (x_{Shared\ node}^n - x_{Mirror}^n) \\
 &\quad + c_3 r_3 (x_{Shared\ node}^n - x_{eye}^n) \quad (4)
 \end{aligned}$$

where c_1, c_2 and c_3 is search step size coefficient, respectively. r_1, r_2 and r_3 is random number at $[-1, 1]$.

Step 5: According to (5), two new specular reflection rings are selected for iterative calculation.

$$\begin{cases}
 X_{New}^n = X_{New_1}^n & if\ f(X_{New_1}^n) < f(X_{New_2}^n) \\
 X_{New}^n = X_{New_2}^n & if\ f(X_{New_1}^n) > f(X_{New_2}^n) \\
 X_{New}^n = X_{New_2}^n = X_{New_1}^n & if\ f(X_{New_1}^n) = f(X_{New_2}^n)
 \end{cases} \quad (5)$$

Step 6: The updated rules of specular reflection rings and shared node is shown in Table 2. According to the Table 2, specular reflection rings are updated accurately and efficiently.

The simplified model and iteration process of SRA and CMSRAS algorithm are described, as shown in Fig. 7 and Fig. 8, respectively. In Fig. 7, the SRA algorithm model is composed of eye, mirror and suspected object, respectively. The specular reflection ring is composed of eye, mirror and suspected target, and the eye can use the reflection and reversing function of the mirror to continuously search for the final target by the specular reflection ring. Based on the

multi-population strategy with shared node, the SRA model is extended to the CMSRAS model. In the CMSRAS model, the population number of eyes, mirrors, and suspected targets are expanded, respectively. The equivalent suspected target is determined by the selection of suspected targets, namely shared node. The specular reflection ring is also modified by the specular reflection rings with shared node which are composed of shared node, eyes and mirrors, respectively. Eyes can efficiently find the final object by the way of the specular reflection rings with shared node. In Fig. 8, through the initial calculation of all fitness values, eyes, mirrors and shared node are defined by the worst solution, near-optimal solution and optimal solution, respectively. The reflection rings are composed of shared node, eyes and mirrors (i.e. the worst solution, near-optimal solution and optimal solution). It shows that the iteration process of CMSRAS algorithm is more complex than SRA. In this way, a group of n initial reflection rings with shared node is generated. In addition, the shared node and reflection ring are updated in each iteration. In the same way, a final optimal solution can be searched after n iterations.

Compared with SRA, CMSRAS algorithm has the following advantages: (a) CMSRAS algorithm mainly consists of three behaviors: the selection behavior of mirror reflection path, a shared node and mirror reversing, respectively. (b) The selection behavior of mirror reflection path is used to fully enhance the ergodicity of CMSRAS algorithm. (c) The selection behavior of a shared node to increase the diversity of the population. (d) The reversing behavior of multiple mirrors can refine the local optimum and improve the search ability of the optimal solution, and to improve the global convergence efficiency and accuracy of the algorithm and avoid falling into the local optimum.

(II) Population initialization based on improved Tent chaos strategy

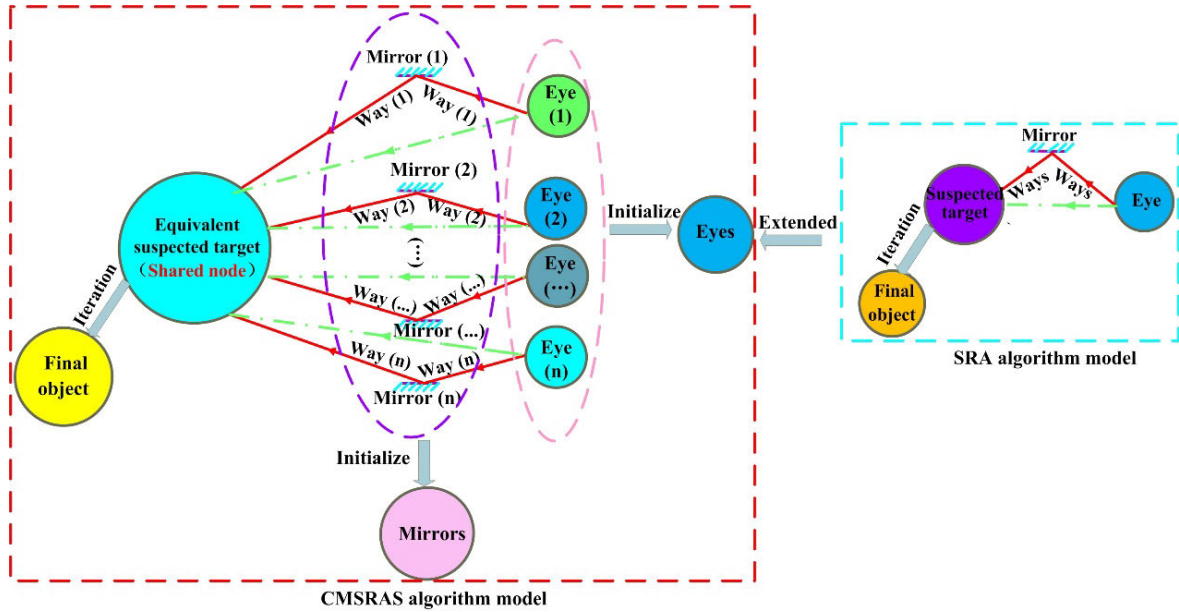


FIGURE 7. Simplified model of CMSRAS.

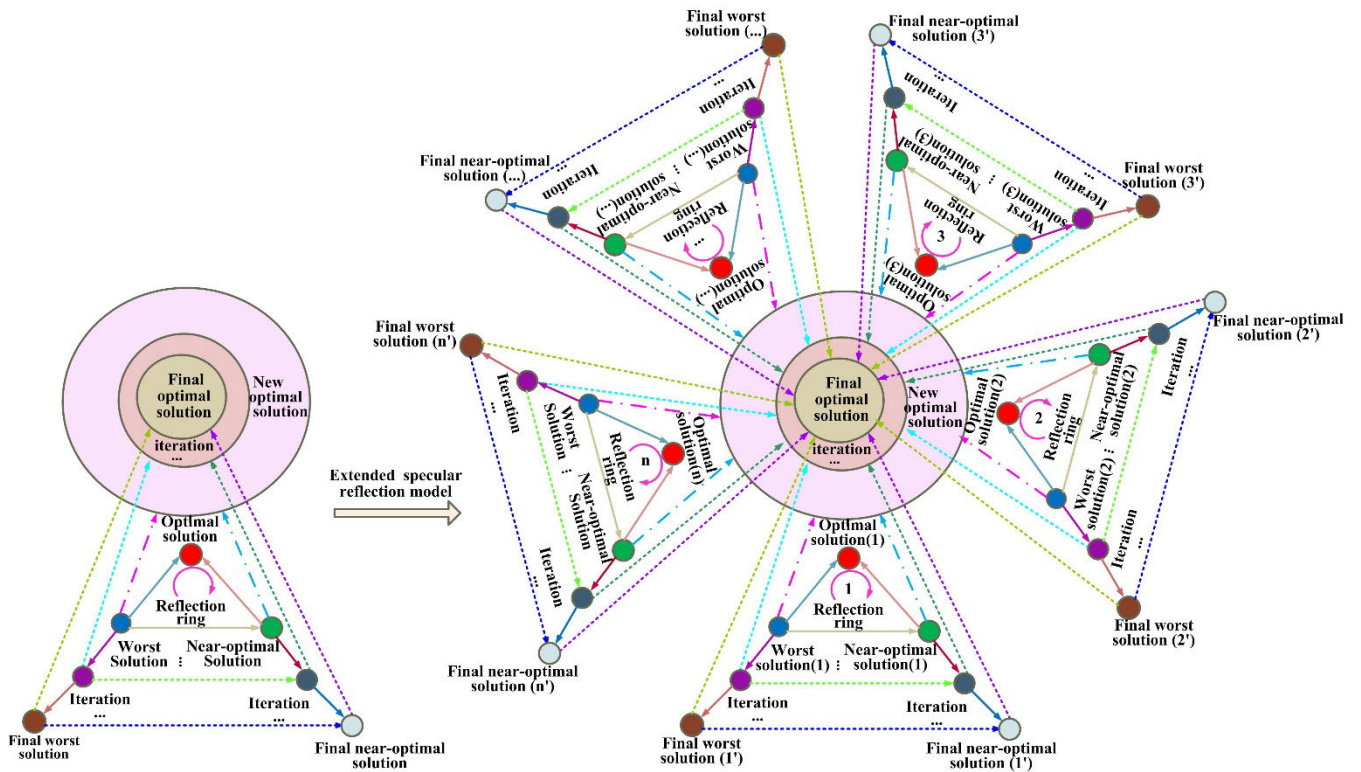


FIGURE 8. The iterative process of CMSRAS.

Based on the multi-population strategy with shared node, the population diversity of CMSRAS algorithm is enriched. Chaos is a kind of nonlinear phenomenon in nature, and CMSRAS algorithm searches for global optimal solutions in chaotic space.

Because the Tent chaotic sequence has the characteristics of randomness, property ergodicity and regularity, therefore, in order to further improve the population diversity and the ability to jump out of local optimal and global search ability of CMSRAS algorithm, in this part, population is initialized

by the way of improved Tent chaos strategy. The form of chaotic mapping has a great influence on the optimization performance of the algorithm [53]. Compared with Logistic chaotic map, the Tent chaotic map has more uniform ergodicity and faster search speed [54]. The expression for the Tent chaotic map is expressed by (6).

$$x_{i+1} = \begin{cases} 2x_i & 0 \leq x \leq 0.5 \\ 2(1 - x_i) & 0.5 < x \leq 1 \end{cases} \quad (6)$$

By the Bernoulli shift method and (6), x_{i+1} is obtained as shown in (7).

$$x_{i+1} = (2x_i) \bmod 1 \quad (7)$$

Due to the existence of small period and unstable periodic points in Tent chaotic map, in order to ensure the randomness, ergodicity and regularity of Tent chaotic map, avoid falling into small period or unstable periodic point, (6) can be improved by the $\text{rand}(0, 1) \times 1/n$, as shown in (8) [55].

$$x_{i+1} = \begin{cases} 2x_i + \text{rand}(0, 1) \times \frac{1}{n} & 0 \leq x \leq 0.5 \\ 2(1 - x_i) + \text{rand}(0, 1) \times \frac{1}{n} & 0.5 < x \leq 1 \end{cases} \quad (8)$$

By the Bernoulli shift method and (8), x_{i+1} is obtained as shown in (9).

$$x_{i+1} = (2x_i) \bmod 1 + \text{rand}(0, 1) \times \frac{1}{n} \quad (9)$$

where n is the population number of specular reflection ring.

The steps of population initialization based on improved Tent chaos strategy as follows:

Step 1: The initial value x_0 is generated randomly in $[0, 1]$, and let $i = 0$.

Step 2: set the maximum number of iterations is \max_i . And according to (8), loop iteration is calculated for i times, chaotic sequence x_d is obtained.

Step 3: If $i < \max_i$, save the x_d .

(III) Improved Tent chaotic disturbance strategy

In order to further improve the ability of jumping out of local optimum and the convergence precision of global optimization, the improved Tent chaotic disturbance strategy is introduced into CMSRAS algorithm. And then specific steps of improved Tent chaotic disturbance are as follows:

Step 1: Chaotic sequence x_d is obtained in (II) section.

Step 2: According to (10), x_d is carried into the search space of the corresponding variable.

$$x_d^{new} = \min(x_d^{new}) + (\max(x_d^{new}) - \min(x_d^{new}))x_d \quad (10)$$

where $\min(x_d^{new})$ and $\max(x_d^{new})$ is the maximum and minimum of x_d^{new} , respectively.

Step 3: Perturbation search for individuals by (11).

$$x'^{new} = (x' + x^{new})/2 \quad (11)$$

where x' is disturbed individuals, x^{new} is chaotic disturbance quantity, and x'^{new} is individual after chaotic disturbance, respectively.

(IV) Gaussian mutation strategy

In the CMSRAS algorithm, the position update mainly depends on the interaction between groups, and the individual has no mutation mechanism. Therefore, the individuals easily fall into the local optimum, which leads the algorithm to premature convergence, reduced population diversity and low optimization accuracy. In order to improve the ability of individual variation, the Gaussian mutation strategy is introduced into the CMSRAS algorithm. The Gaussian mutation is to add a random number that obeys the normal distribution $N(\mu, \sigma^2)$ to replace the original parameter value [56], as shown in (12).

$$x'^{new} = x^{new}(1 + N(0, 1)) \quad (12)$$

where x^{new} is the original parameter value, $N(0, 1)$ is random number with a standard normal distribution, and x'^{new} is the parameter value after Gaussian mutation.

The Gaussian mutation based on normal distribution can ensure the individual to search in the local neighborhood of the individual, and the local search ability is greatly improved, which can improve the ability and robustness of the CMSRAS algorithm and avoid the local optimum.

B. PSEUDOCODE AND FLOW CHART OF CMSRAS ALGORITHM

The pseudocode and flow chart of the proposed CMSRAS algorithm are reported in Algorithm 1 and Fig. 9, respectively.

Algorithm 1 Pseudo Code of CMSRAS

Initialize the parameters: specular reflection ring n , $\max_iteration$. variable dimension Dim

Initialize the population x_i ($i = 1, 2, \dots, 2n + 1$) by (9) and (10)

Specular reflection ring (shared node, eyes, mirrors, respectively.) is obtained by $f(x_i)$.

While ($k \leq \max_iteration$)

For each specular reflection ring in population **do**
 Updated the positions of specular reflection ring by (3) and (4)

End for

 Updated fitness f_i and calculated mean fitness f_{ave}

For each specular reflection ring in population **do**

If ($f_i < f_{ave}$)

 Update specular reflection ring by Eq. (12)

Else

 Update specular reflection ring by Eq. (11)

End if

For each specular reflection ring in population **do**

 Update specular reflection ring by Eq. (5)

End for

End for

$k = k + 1$

End while

Return the best solution found

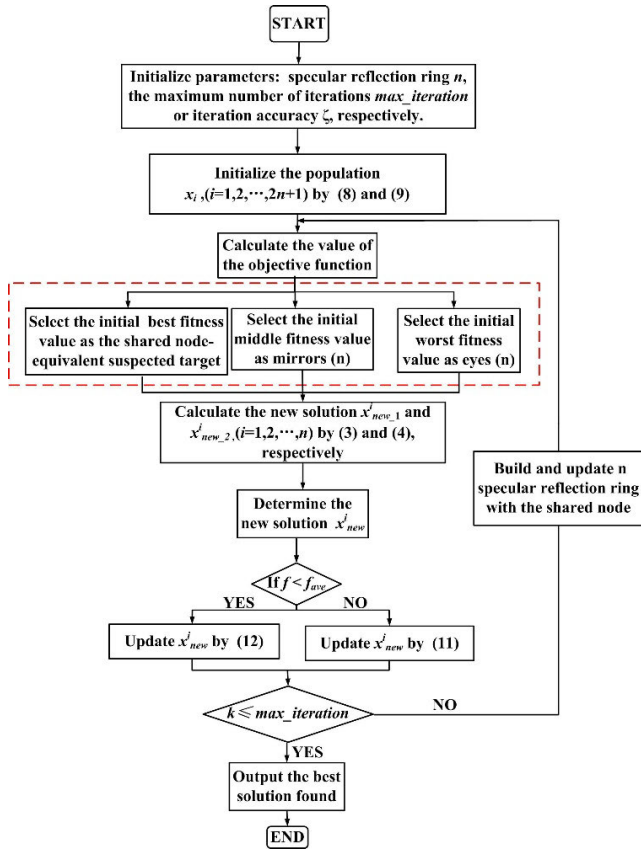


FIGURE 9. The flow chart of proposed CMSRAS algorithm.

C. COMPUTATIONAL COMPLEXITY ANALYSIS OF CMSRAS ALGORITHM

Note that the computational complexity of the CMSRAS algorithm mainly depends on three parts: initialization, definition of the specular reflection rings and specular reflection rings update. Among them, N denotes the number of specular reflection rings, D denotes the dimension of functions, and T denotes the maximum number of iterations. The computational complexity of initialization is $O((2 \times N + 1) \times D)$, and the computational complexity of definition of the specular reflection rings is $O(2 \times N + 1)$, and the computational complexity of specular reflection rings update is $O((N \times D) \times (N + 1))$. Therefore, the total complexity of the CMSRAS algorithm is $O((2 \times N + 1) \times (1 + D)) + O(T \times N \times D \times (N + 1))$.

IV. CHANGE TREND AND SENSITIVITY ANALYSIS OF PARAMETERS IN CMSRAS ALGORITHM

Because there are many parameters in the multi mirror optimization algorithm, such as the population number, the design variable dimension, step size factors, etc., whether these parameters are set reasonably or not determines the performance of the CMSRAS algorithm. However, there is a recessive relationship between these parameters and the performance of the algorithm. In order to study the influence of the parameters on the performance of the

CMSRAS algorithm, it is necessary to convert the implicit relationship into explicit relationship. Response surface analysis method (RSM) is a popular method that can realize the conversion of implicit relationship into explicit relationship [57], [58].

In this paper, by selecting the step size factors c_1, c_2 and c_3 , the number of mirror populations n and the design variable dimension D are selected as factor indexes, respectively. Meanwhile, the average iteration times N , convergence time T and optimal solution are taken as the objective factors, respectively.

The selected test function is shown in (13), and the image is shown in Fig. 10. And then set $0.1 \leq c_1 \leq 1.7, 0.1 \leq c_2 \leq 1.7, 0.1 \leq c_3 \leq 1.7, 10 \leq n \leq 50, n \in N^+, 10 \leq D \leq 100, D \in N^+$, respectively. At the same time, the independently calculation times are set to 30, the maximum iteration times are 500 and the iteration accuracy is 0.

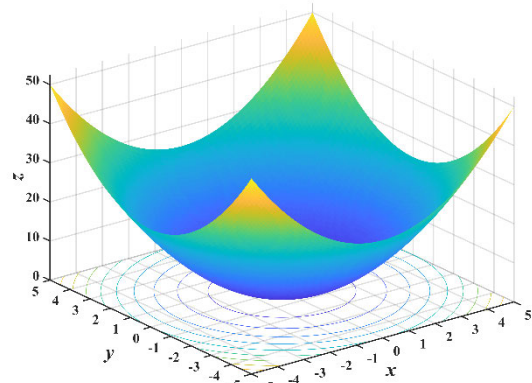


FIGURE 10. The test function.

Based on the RSM, the orthogonal test analysis results are obtained, as shown in Table 3. It is found that the accuracy of all optimal solutions has reached 0. Therefore, only the relationship between c_1, c_2, c_3, n, D and N, T is analyzed, respectively. Explicit objective functions of average convergence time and average number of iterations are determined, as shown in (14) and (15), respectively.

$$f(x) = \sum_{i=1}^D x_i^2 \tag{13}$$

$$T = 2.27150 - 2.06974 \times c_1 - 2.14435 \times c_2 - 1.11918 \times c_3 - 6.08949E - 003 \times D + 0.35613 \times n + 0.35613 \times c_1 \times c_2 + 0.34417 \times c_1 \times c_3 + 0.010766 \times c_1 \times D + 0.018506 \times c_1 \times n + 0.45426 \times c_2 \times c_3 + 9.3963E - 003 \times c_2 \times D + 0.024198 \times c_2 \times n - 2.12222E - 003 \times c_3 \times D + 0.015033 \times c_3 \times n + 4.90213E - 004 \times n \times D + 0.64025 \times c_1^2 + 0.55815 \times c_2^2 + 0.079309 \times c_3^2 + 1.45701E - 004 \times D^2 - 6.75825E - 004 \times n^2 \tag{14}$$

$$N = 5778.28124 - 531.97259 \times c_1 - 684.7207$$

TABLE 3. The orthogonal test analysis results.

ID	c_1	c_2	c_3	D	n	T	N	Average optimal solution
1	0.1	0.1	0.9	55	30	1.055267	812.7	0
2	1.7	0.1	0.9	55	30	2.826967	2204.8	0
3	0.1	1.7	0.9	55	30	2.76483	2089.63	0
4	1.7	1.7	0.9	55	30	5.448233	3669.2333	0
5	0.9	0.9	0.1	10	30	0.781	1710.6	0
6	0.9	0.9	1.7	10	30	0.8213	1763.766	0
7	0.9	0.9	0.1	100	30	4.1801	1758.433	0
8	0.9	0.9	1.7	100	30	3.9148	1615.733	0
9	0.9	0.1	0.9	55	10	1.309	3262.333	0
10	0.9	1.7	0.9	55	10	1.399367	3394.8333	0
11	0.9	0.1	0.9	55	50	2.28137	989.2333	0
12	0.9	1.7	0.9	55	50	3.9204	1670.333	0
13	0.1	0.9	0.1	55	30	2.093733	1604.466	0
14	1.7	0.9	0.1	55	30	3.10006	2319.3333	0
15	0.1	0.9	1.7	55	30	1.990566	1505.133	0
16	1.7	0.9	1.7	55	30	3.8779666	2674.4333	0
17	0.9	0.9	0.9	10	10	0.5285	3785.533	0
18	0.9	0.9	0.9	100	10	2.8089	3258.8666	0
19	0.9	0.9	0.9	10	50	1.0198	1138.9333	0
20	0.9	0.9	0.9	100	50	5.0649667	1117.8333	0
21	0.9	0.1	0.1	55	30	2.1850667	1678.2667	0
22	0.9	1.7	0.1	55	30	2.8749333	2157.96667	0
23	0.9	0.1	1.7	55	30	1.8847333	1437.7333	0
24	0.9	1.7	1.7	55	30	3.7375	2663.5	0
25	0.1	0.9	0.9	10	30	0.698833	1539.5	0
26	1.7	0.9	0.9	10	30	1.0654	2364.5333	0
27	0.1	0.9	0.9	100	30	4.0705333	1483.5	0
28	1.7	0.9	0.9	100	30	5.987466	2184.9333	0
29	0.9	0.9	0.1	55	10	1.6144333	3770.5	0
30	0.9	0.9	1.7	55	10	1.11656667	2575.6	0
31	0.9	0.9	0.1	55	50	2.6456	1110.0333	0
32	0.9	0.9	1.7	55	50	3.10986667	1278.26667	0
33	0.1	0.9	0.9	55	10	1.3867	3258.06667	0
34	1.7	0.9	0.9	55	10	1.484433	3416.3333	0
35	0.1	0.9	0.9	55	50	2.55156667	1045.6333	0
36	1.7	0.9	0.9	55	50	3.8337	1575.1	0
37	0.9	0.1	0.9	10	30	0.7243	1542.433	0
38	0.9	1.7	0.9	10	30	1.0737333	2314.8667	0
39	0.9	0.1	0.9	100	30	4.20876667	1519.46667	0
40	0.9	1.7	0.9	100	30	5.9112667	2125.5	0
41	0.9	0.9	0.9	55	30	2.2468333	1629.633	0
42	0.9	0.9	0.9	55	30	2.2468333	1629.633	0
43	0.9	0.9	0.9	55	30	2.2468333	1629.633	0
44	0.9	0.9	0.9	55	30	2.2468333	1629.633	0
45	0.9	0.9	0.9	55	30	2.2468333	1629.633	0
46	0.9	0.9	0.9	55	30	2.2468333	1629.633	0
47	0.9	0.9	0.9	55	30	2.2468333	1629.633	0
48	0.9	0.9	0.9	55	30	2.2468333	1629.633	0
49	0.9	0.9	0.9	55	30	2.2468333	1629.633	0
50	0.9	0.9	0.9	55	30	2.2468333	1629.633	0

$$\begin{aligned}
 & \times c_2 - 1266.86679 \times c_3 - 5.41298 \\
 & \times D - 167.60062 \times n + 73.24348 \times c_1 \times c_2 \\
 & + 177.51289 \times c_1 \times c_3 - 0.85833 \times c_1 \times D \\
 & + 5.8 \times c_1 \times n + 291.43232 \times c_2 \times c_3 \\
 & - 1.15556 \times c_2 \times D + 8.57187 \times c_2 \\
 & \times n - 1.36018 \times c_3 \times D + 21.29896 \times c_3 \times n \\
 & + 0.14044 \times n \times D + 406.58302 \times c_1^2 \\
 & + 378.65336 \times c_2^2 + 130.07723 \times c_3^2 \\
 & + 0.024677 \times D^2 + 1.25472 \times n^2 \tag{15}
 \end{aligned}$$

According to (14) and (15), the change trends of average convergence time T and average number of iterations N with c_1, c_2, c_3, n and D can be determined respectively, as shown in Fig. 11 and 12.

As n and D is constant, it can be seen in Fig. 11(a), 11(b) and 11(c) that T decreases rapidly and then increases slowly with the increase of c_1 and c_2 , but decreases rapidly with the increase of c_3 . In Fig. 11(a), when $0 \leq c_1 \leq 0.9, 0 \leq c_2 \leq 0.8, 0 \leq c_3 \leq 0.95, T$ decreases with the increase of c_1 and the decrease rate v_{c11} becomes smaller and smaller. When $0 \leq c_1 \leq 0.3, 0 \leq c_2 \leq 1.7, 0 \leq c_3 \leq 1.7, T$

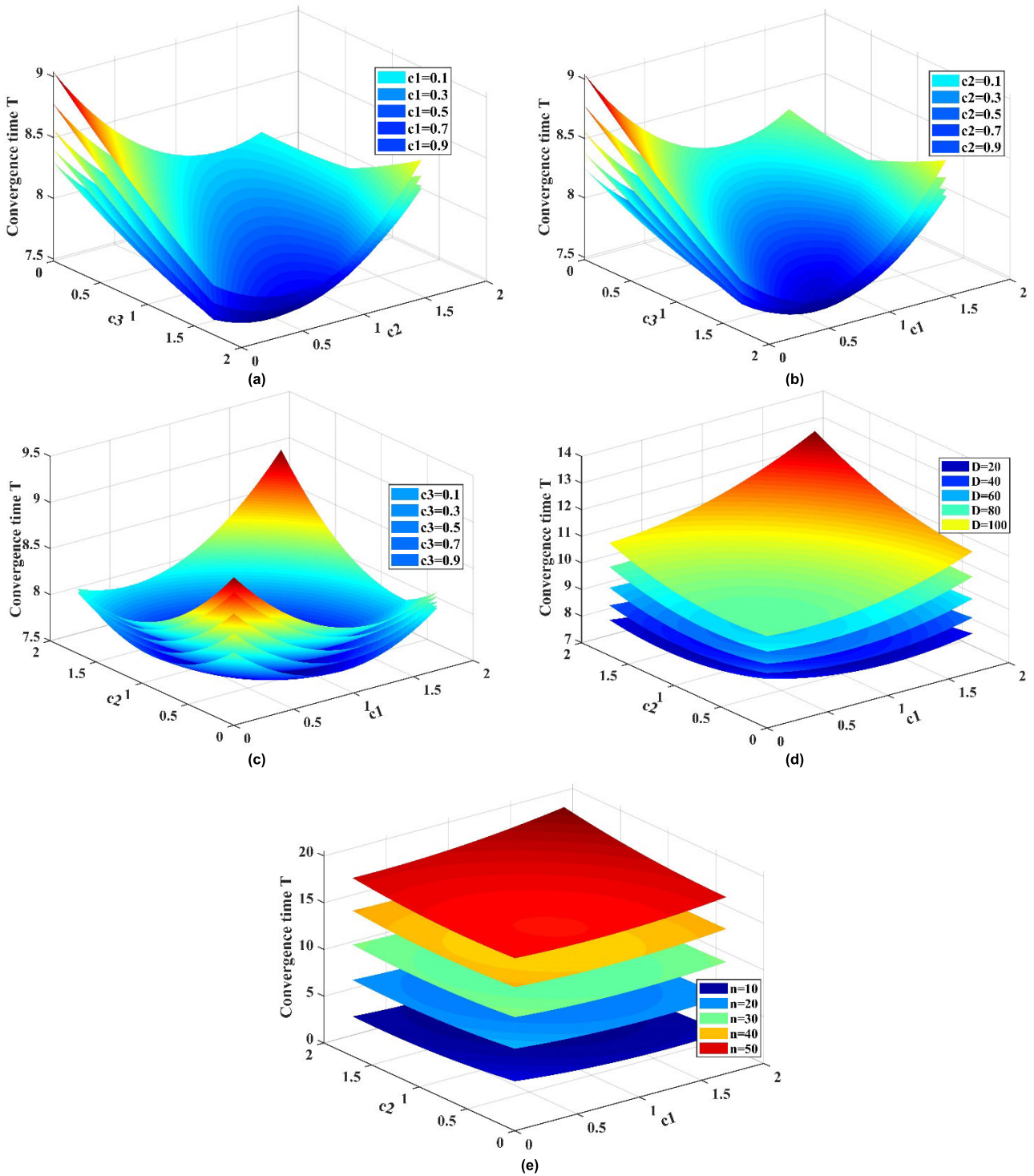


FIGURE 11. Change trend of average convergence time T with (a) c_1 , (b) c_2 , (c) c_3 , (d) D , (e) n .

decreases with the increase of c_1 , and the decreasing rate v'_{c11} becomes smaller and smaller and $v'_{c11} < v_{c11}$. However, when $0.5 \leq c_1 \leq 0.9$, $1.3 \leq c_2 \leq 1.7$, $0.4 \leq c_3 \leq 1.7$, T increases with the increase of c_1 and the change rate v_{c11} is getting bigger and bigger. In Fig. 11(b), when $0 \leq c_1 \leq 0.9$, $0 \leq c_2 \leq 0.6$, $0 \leq c_3 \leq 0.8$, T decreases with the

increase of c_2 and the decrease rate v'_{c21} becomes smaller and smaller. However, when $1.5 \leq c_1 \leq 1.7$, $0.1 \leq c_2 \leq 0.9$, $1.4 \leq c_3 \leq 1.7$, T increases with the increase of c_2 and the change rate v_{c21} is getting faster and faster. In Fig. 11(c), when $0 \leq c_1 \leq 0.9$, $0 \leq c_2 \leq 0.9$, $0 \leq c_3 \leq 0.9$, T decreases with the increase of c_3 and the decrease rate v_{c31} becomes

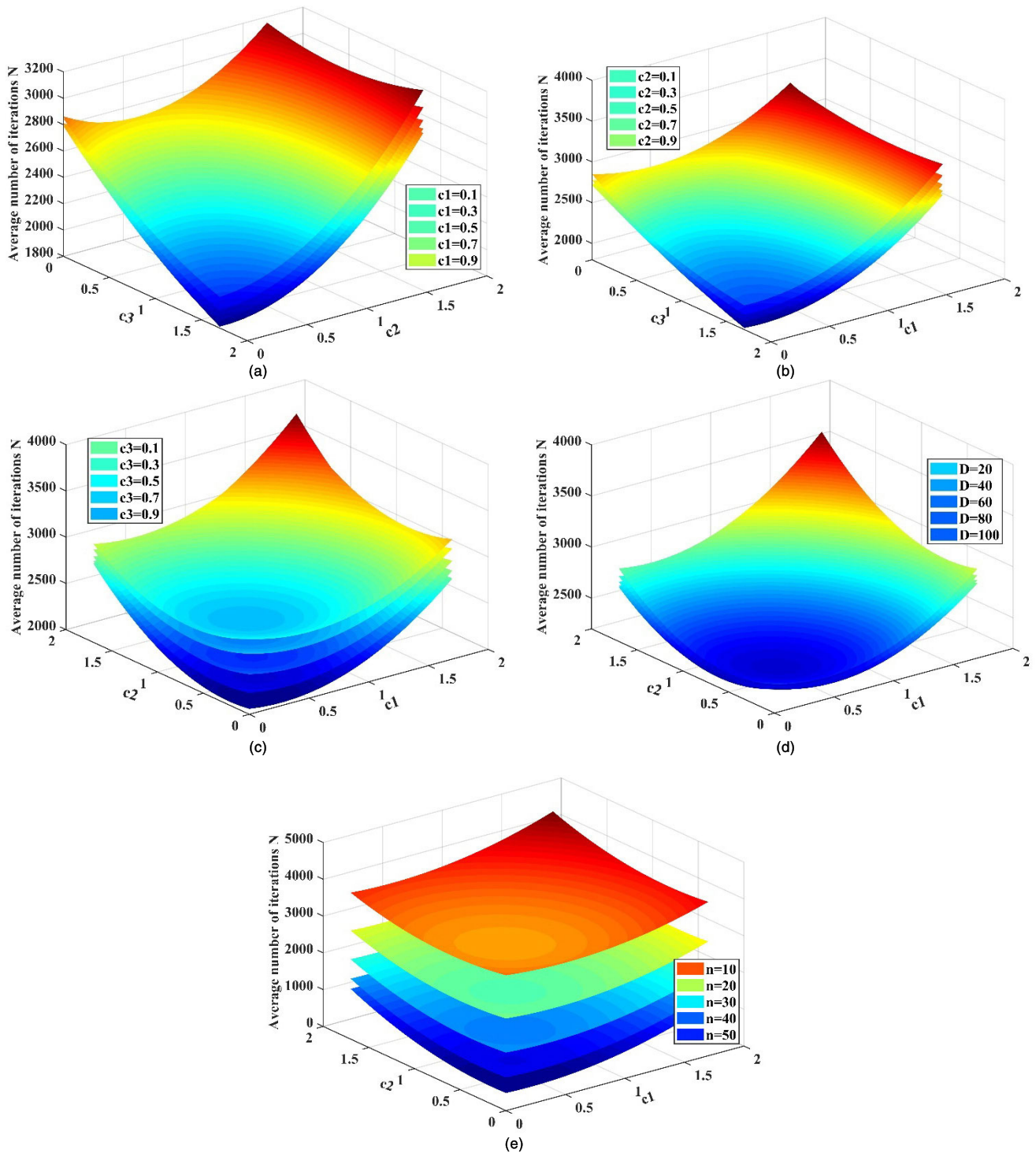


FIGURE 12. Change trend of average number of iteration N with (a) c_1 , (b) c_2 , (c) c_3 , (d) D , (e) n .

larger and larger. However, when $0.9 \leq c_1 \leq 1.7$, $0.9 \leq c_2 \leq 1.7$, $0.1 \leq c_3 \leq 0.9$, T increases with the increase of c_3 and the change rate v_{c31} becomes larger and larger. In Fig. 11(d) and 11(e), when c_1 , c_2 and c_3 is constant, T increases with the increase of n and D , and the decrease rate v_{n1} and v_{D1} remain nearly constant and $v_{n1} > v_{D1}$. When n and D is

constant, it can be seen in Fig. 12(a), 12(b) and 12(c) that N decreases slowly and then increases rapidly with the increase of c_1 and c_2 , but decreases rapidly with the increase of c_3 . In Fig. 12(a), when $0 \leq c_1 \leq 0.5$, $0 \leq c_2 \leq 1.4$, $0 \leq c_3 \leq 6$, N decreases with the increase of c_1 and the decrease rate v_{c11} becomes smaller and smaller. However, when

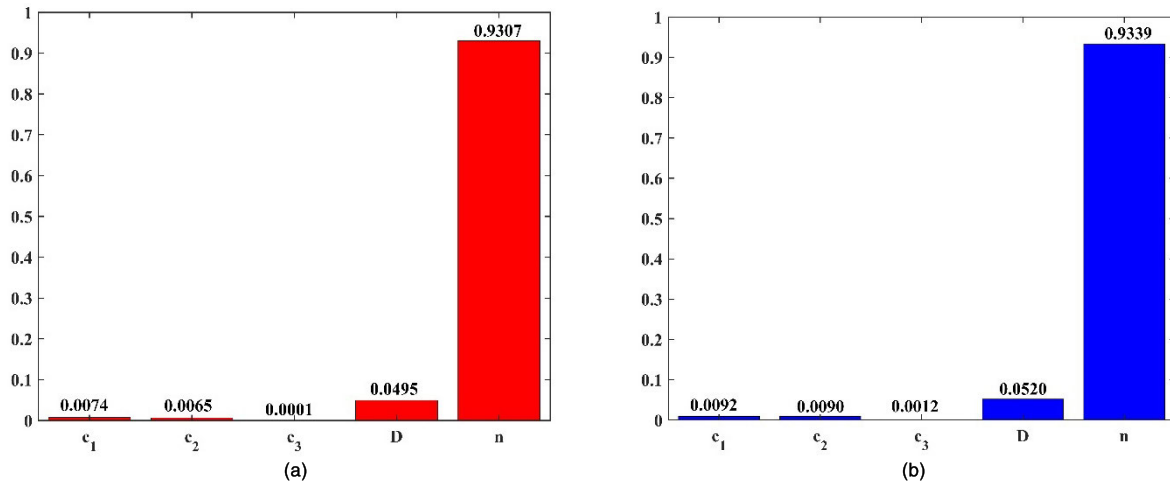


FIGURE 13. First order effects (a) and total effects (b) of average convergence time T using Sobol's method of sensitivity analysis.

$0.5 \leq c_1 \leq 0.9, 0 \leq c_2 \leq 1.7, 1.5 \leq c_3 \leq 1.7, T$ increases with the increase of c_1 , and the increasing rate v'_{c12} becomes larger and larger. In Fig. 12(b), when $0 \leq c_1 \leq 1.7, 0 \leq c_2 \leq 0.5, 0 \leq c_3 \leq 0.6, N$ decreases with the increase of c_2 and the decrease rate v'_{c22} becomes smaller and smaller. However, when $0 \leq c_1 \leq 1.7, 0.1 \leq c_2 \leq 0.9, 1.5 \leq c_3 \leq 1.7, N$ increases with the increase of c_2 and the increase rate v'_{c22} is getting faster and faster. In Fig. 12(c), when $0 \leq c_1 \leq 1.7, 0 \leq c_2 \leq 1.7, 0.1 \leq c_3 \leq 0.9, N$ increases with the increase of c_3 and the increase rate v'_{c32} becomes uniformly. However, when $0 \leq c_1 \leq 1.7, 0 \leq c_2 \leq 1.7, 0 \leq c_3 \leq 0.3, N$ decreases with the increase of c_3 and the decrease rate v'_{c32} becomes smaller and smaller. In Fig. 12(d) and 12(e), when c_1, c_2 and c_3 is constant, it shows that N decreases rapidly with the increase of D but decreases rapidly and then slowly with the increase of n .

All in all, these results show that the reasonable parameter setting plays a crucial role in the performance of CMSRAS algorithm.

Parameter sensitivity analysis is helpful to set the parameters reasonably and ensure the performance of CMSRAS algorithm. Sobol's method is a global sensitivity analysis method based on variance.

In practical application, Sobol's method is relatively easy to implement by using Monte Carlo simulation method. The first order and total sensitivity indexes of Sobol's method are relatively easy to obtain. The detailed calculation steps are shown in Ref. [59].

Therefore, in this paper, Sobol's method is used to determine the sensitivity of setting parameters of CMSRAS algorithm.

It supposes that the parameters c_1, c_2, c_3, n and D obey uniform distribution and $0.1 \leq c_1 \leq 1.7, 0.1 \leq c_2 \leq 1.7, 0.1 \leq c_3 \leq 1.7, 10 \leq n \leq 50, 10 \leq D \leq 100$ and the simulation times by Monte Carlo method are 20000, respectively.

According to (14) and (15), the first-order effect sensitivity coefficient and total effect sensitivity coefficient of the setting parameters to the average convergence time and the average number of iterations are calculated respectively, and the results are shown in Fig. 13 and Fig. 14.

It can be seen from Fig. 13(a) that the first-order effect sensitivity coefficient of the average convergence time in c_1, c_2, c_3, D and n is 0.0074, 0.0065, 0.001, 0.0495, 0.9307, respectively. The result shows that n and D are more sensitive to the average convergence time, followed by c_1, c_2 , and c_3 . Similarly, it can be seen from Fig. 14(a) that the first-order effect sensitivity coefficient of the average number of iterations in c_1, c_2, c_3, D and n is 0.1179, 0.1015, 0.0049, 0.0027, 0.7259, respectively. The results show that n and c_1 are more sensitive to the average convergence time, followed by c_3, c_2 , and D . It can be seen from Fig. 13(b) that the total effect sensitivity coefficient of the average convergence time in c_1, c_2, c_3, D and n is 0.0092, 0.0090, 0.0012, 0.0520, 0.9339, respectively. The result shows that n and D are more sensitive to the average convergence time, followed by c_1, c_2 and c_3 . Similarly, it can be seen from Fig. 14(b) that the total effect sensitivity coefficient of the average number of iterations in c_1, c_2, c_3, D and n is 0.1240, 0.1153, 0.0420, 0.0068, 0.7602, respectively. The results show that n and c_1 are more sensitive to the average convergence time, followed by c_2, c_3 and D .

In summary, these results indicated that c_1, D and n are more sensitive to the performance of CMSRAS algorithm.

V. PERFORMANCE EVALUATION OF CMSRAS ALGORITHM

In this section, we compared the CMSRAS with some competitive meta-heuristic algorithms by 32 benchmark functions. The experiments were run on the operating system of Windows 10, the CPU of Intel (R) Xeon (R) Gold5118 CPU@2.3Hz 2.29Hz and the memory of 64G.

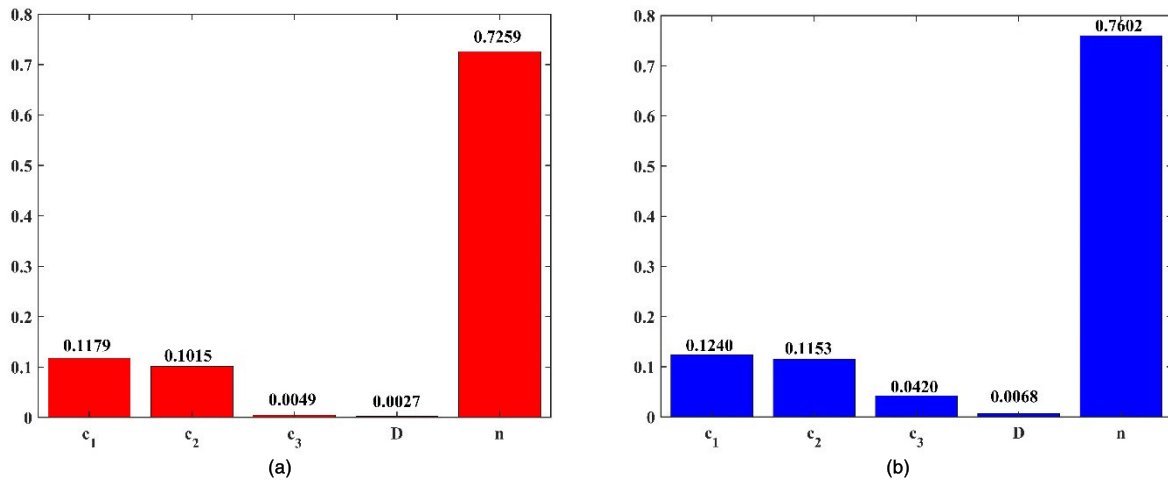


FIGURE 14. First order effects (a) and total effects (b) of average number of iteration N using Sobol’s method of sensitivity analysis.

TABLE 4. Setting parameters of the algorithm.

Function	Dim	Range	f_{min}
$F_1(x) = \sum_{i=1}^n x_i^2$	n	[-100,100]	0
$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	n	[-10,10]	0
$F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	n	[-100,100]	0
$F_4(x) = \max_i \{ x_i , 1 < x < n \}$	n	[-100,100]	0
$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	n	[-30,30]	0
$F_6(x) = \sum_{i=1}^n [(x_i + 0.5)]^2$	n	[-100,100]	0
$F_7(x) = \sum_{i=1}^n ix_i^4 + random[0,1]$	n	[-1.28,1.28]	0

TABLE 5. Description of unimodal test functions.

Function	Dim	Range	f_{min}
$F_8(x) = \sum_{i=1}^n -x_i \sin \sqrt{ x_i }$	n	[-500,500]	-418.9829×Dim
$F_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	n	[-5.12,5.12]	0
$F_{10}(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + e$	n	[-32,32]	0
$F_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	n	[-600,600]	0
$F_{12}(x) = \frac{\pi}{n} \{ \sin(\pi y_1) \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}$	n	[-50,50]	0
$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$			
$F_{13}(x) = \sum_{i=1}^n 0.1 \{ \sin^2(3\pi x_i) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	n	[-50,50]	0

All the algorithm codes are written in M file by MATLAB R2019a version.

A. BENCHMARK FUNCTIONS SET AND COMPARED ALGORITHM

In order to verify the performance of the CMSRAS algorithm proposed in this paper, some diverse subsets of benchmark functions are selected as shown in Tables 4-7, which

include unimodal functions (F1-F7), multimodal functions (F8-F13), fixed-dimension unimodal functions (F14-F27) and fixed-dimension multimodal functions (F28-F32), respectively. These functions are widely used to test the various characteristics of the proposed algorithm, such as the ability of fast convergence ability, global convergence, avoiding the local optimum and premature convergence, respectively. In Tables 4-7, Dim is the dimension of

TABLE 6. Description of multimodal test functions.

Function	Dim	Range	f_{min}
$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6}\right)^{-1}$	2	[-65,65]	1
$F_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[-5,5]	0.00030
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[-5,5]	-1.0316
$F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10 \left(1 - \frac{1}{8\pi}\right) \cos x_1 + 10$	2	[-5,5]	0.398
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[-2,2]	3
$F_{19}(x) = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2)$	3	[1,3]	-3.86
$F_{20}(x) = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2)$	6	[0,1]	-3.32
$F_{21}(x) = -\sum_{i=1}^5 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.1532
$F_{22}(x) = -\sum_{i=1}^7 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.4028
$F_{23}(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.5363
$F_{24}(x) = \sin^2(3\pi x_1) + (x_1 - 1)^2 [1 + \sin^2(3\pi x_2)] + (x_1 - 1)^2 [1 + \sin^2(2\pi x_2)]$	2	[-10,10]	0
$F_{25}(x) = -\cos(x_1)\cos(x_2)\exp[-(x_1 - \pi)^2 - (x_2 - \pi)^2]$	2	[-100,100]	-1
$F_{26}(x) = (x_1^2 + x_2^2)^{0.25} [50(x_1^2 + x_2^2)^{0.1} + 1]$	2	[-100,100]	0
$F_{27}(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7$	2	[-100,100]	0

TABLE 7. Description of fixed-dimension multimodal test functions.

Function	Dim	Range	f_{min}
$F_{28}(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$	2	[-4.5,4.5]	0
$F_{29}(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	2	[-10,10]	0
$F_{30}(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	2	[-10,10]	0
$F_{31}(x) = (x_1^2 + x_2^2 - 2x_1)^2 + 0.25x_1$	2	[-1,5]	-0.00379
$F_{32}(x) = 100(x_2 - x_1^3)^2 + (1 - x_1)^2$	2	[-1,2,1,2]	0

benchmark functions, and Range is the definition of benchmark functions and f_{min} is the global optimum of benchmark functions, respectively. The results and performance of the proposed CMSRAS algorithm is compared with some well-known meta-heuristic algorithms including both traditional meta-heuristic algorithms: PSO [6], CS [19], DA [31], GWO [26], MFO [28], HHO [37], MVO [27], SMA [38], SCA [30], SOA [35], WOA [29], SRA [47]–[49] and advanced meta-heuristic algorithms: TAPSO [61], MPSO [62], IPSO [63], I-GWO [64], AGPSO1 [60], AGPSO2 [60], AGPSO3 [60], GWOCs [65]. The parameter setup of all traditional and advanced meta-heuristic algorithms is detailed in Table 8. The parameters of all algorithms are set by more commonly-used or popular parameters in literatures.

In order to ensure the fairness of competitive experiments, all the algorithms were carried out under the same experimental conditions. In this paper, the population and the maximum iterations of all algorithms were set to 30 and 2000, respectively. In addition, for the purpose of avoiding the error caused by random factors, all the algorithms were independently run by 30 times for all tested benchmark functions, and at the same time, the average value (AVG) and standard deviation (STD) were selected as the evaluation index of the experimental results.

TABLE 8. Description of fixed-dimension unimodal test functions.

Algorithms	Parameter settings
PSO	$w_{Max}=0.9; w_{Min}=0.6; c_1=c_2=2, v_{max}=6$
CS	$p_a=0.25$
DA	$w = 0.9-0.2, s = 0.1, a = 0.1, c = 0.7, f=1, e=1$
GWO	$a=[2,0]$
MFO	$b=1; t=[-1,1]; a=[-1,-2]$
HHO	$beta=1.5; E0=[-1,1]; J=[-2,2]$
MVO	Wormhole Existence Prob. [0.2, 1] Traveling Distance Rate [0.6, 1]
SMA	$z=0.03$
SCA	$A=2$
SOA	$A=[2,0]; f_c=2$
WOA	$a_1=[2,0]; a_2=[-2,-1]; b=1$
SRA	$\xi=1.68$
CMSRAS	$c_1=0.4; c_2=0.6; c_3=1.68$
TAPSO	$w_{Max}=0.9; w_{Min}=0.6; v_{max}=6$
MPSO	$w_{Max}=0.9; w_{Min}=0.6; v_{max}=6$
IPSO	$w_{Max}=0.9; w_{Min}=0.6; v_{max}=6$
I-GWO	$a=[2,0]$
AGPSO1	$w_{Max}=0.9; w_{Min}=0.6; v_{max}=6$
AGPSO2	$w_{Max}=0.9; w_{Min}=0.6; v_{max}=6$
AGPSO3	$w_{Max}=0.9; w_{Min}=0.6; v_{max}=6$
GWOCs	$beta=1.5; a=[2,0]; L=0.001.$

B. QUALITATIVE ANALYSIS OF CMSRAS ALGORITHM

In order to demonstrate the convergence analysis of the CMSRAS algorithm, the search history, the trajectory of mirrors in the 1st dimension, the average fitness of mirrors

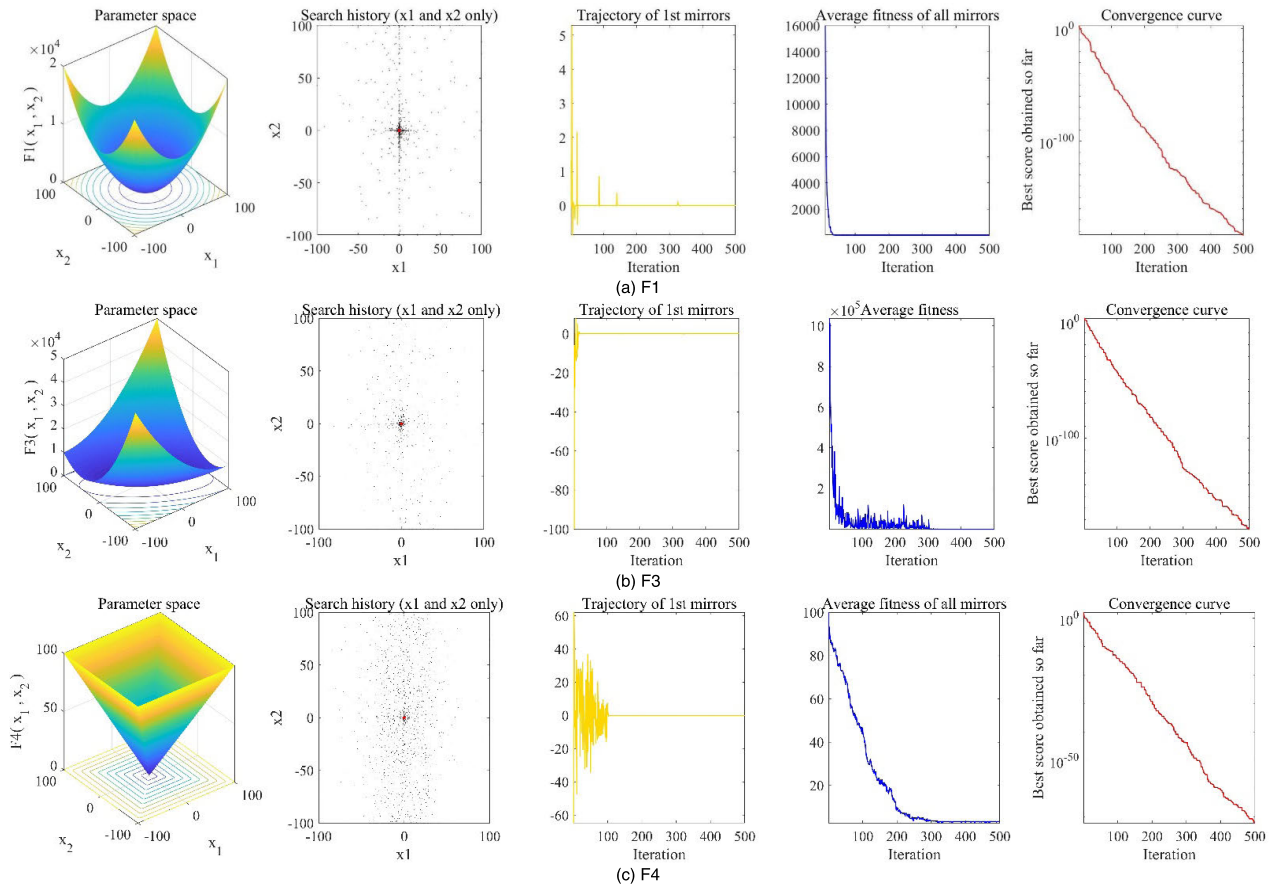


FIGURE 15. Qualitative results of unimodal problems.

and the convergence curve are selected as the evaluation index in the 2D environment as shown in Fig. 15 and Fig. 16, respectively. The search history shows the location and distribution of mirrors during the iteration process. The trajectory of the 1st mirror shows the value of the first variable in each iteration. The average fitness indicates the average objective value of all mirrors in each iteration. The convergence curve indicates the optimal objective value of all mirrors in the iteration process.

From the search history in Fig. 15 and Fig. 16, it shows that the mirrors are similarly gathered near the optimal solution. Meanwhile, the optimal solution is precisely searched in the search area by the frequently reversing and rapidly reflecting behavior of the mirror. For unimodal functions, the distribution of mirrors is relatively discrete, and the phenomenon of local optimum aggregation is not obvious. However, for multimodal functions, the distribution of mirrors is mainly concentrated in multiple regions with local optimum, which fully demonstrates that mirrors can realize the tradeoff between multiple local optimums.

As it can be seen in Fig. 15 and Fig. 16, the trajectory of 1st mirrors indicates that the preliminary exploratory behavior of mirrors. Through the larger oscillation in the initial stage and the smaller oscillation in the later stage, mirrors can

improve the convergence speed and the search accuracy of the optimal solution [66]. In the initial stage, the space search is larger than that in the later stage, and even at 60% of the exploration space. Compared with the unimodal functions, the position of mirrors fluctuates greatly in the late stage for the multimodal functions, and varies with the value of the benchmark functions. These results show that mirrors are more versatile and have higher robustness in different functional functions.

From the average convergence curve in Fig. 15 and Fig. 16, the results show that the average convergence curve of the mirror decreases first rapidly and then slowly with the increase of iterations, and the amplitude of oscillation attenuation is relatively small. Thus, the ability of fast convergence in the early stage and accurate search in the later stage are ensured.

According to the convergence curve in Fig. 15 and Fig. 16, it is obvious that CMSRAS can reveal an accelerated in the iteration stage, the ability of shifting from exploration to exploitation is higher and the rate of convergence is faster.

C. EXPLOITATION COMPETENCE ANALYSIS

(I) CMSRAS compared with SRA

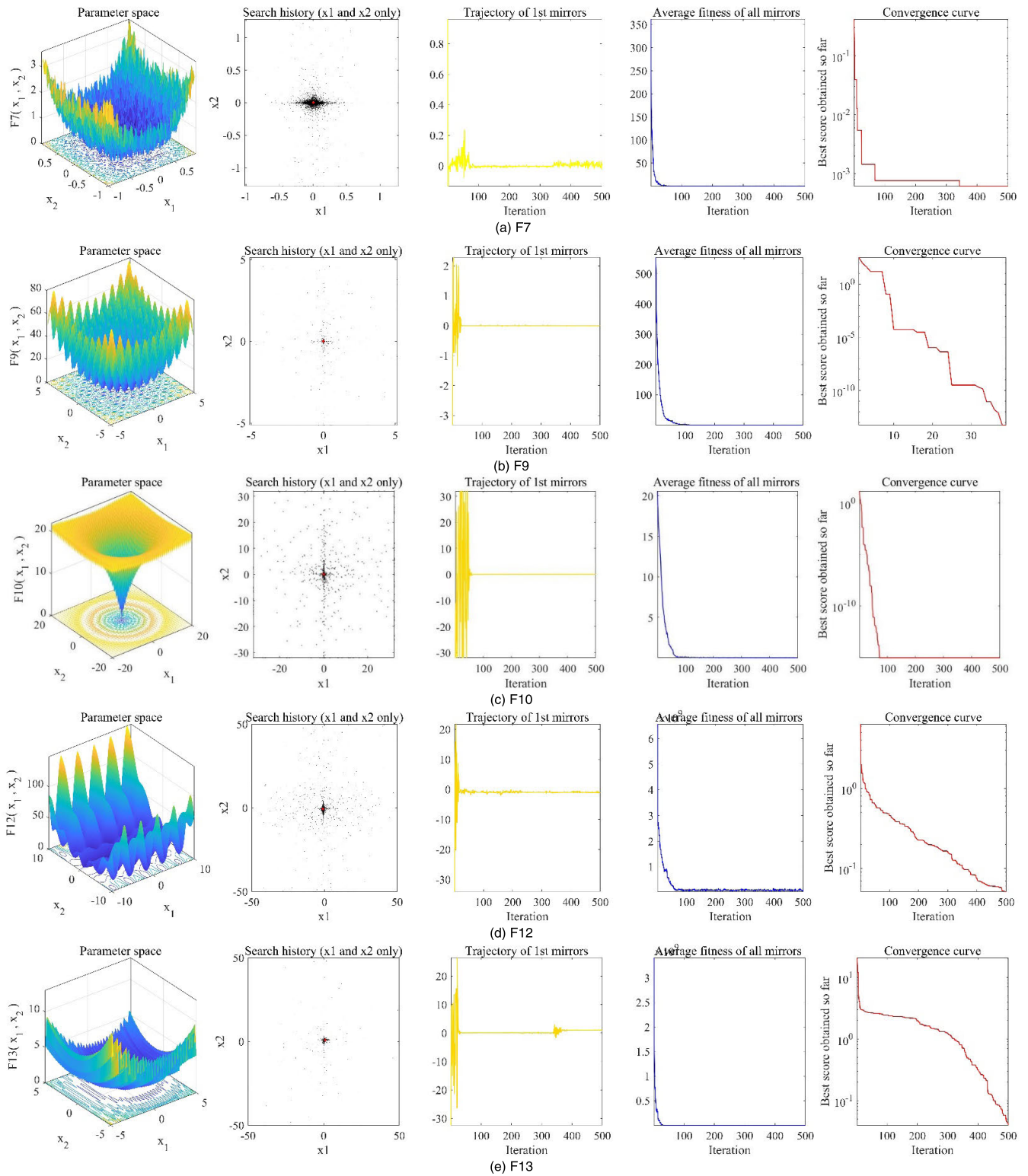


FIGURE 16. Qualitative results of multimodal problems.

Unimodal functions can be well used to test the exploitation ability of algorithms. Therefore, in this section, in order to compare performance of CMSRAS with basic SRA more

comprehensively, the exploitation ability test was carried out. In different dimensions of search space and population size, the performance of both CMSRAS and SRA is performed

TABLE 9. Results on unimodal benchmark functions by different population size.

F1	AVG		STD		F2	AVG		STD	
	CMSRAS	SRA	CMSRAS	SRA		CMSRAS	SRA	CMSRAS	SRA
10	4.521E-239		0.000		10	5.028E-138		2.754E-137	
30	0.000	8.69E-04	0.000	7.125E-04	30	8.726E-198	8.088E-03	0.000	2.112E-03
50	0.000		0.000		50	7.058E-237		0.000	
F3	AVG		STD		F4	AVG		STD	
	CMSRAS	SRA	CMSRAS	SRA		CMSRAS	SRA	CMSRAS	SRA
10	1.547E-230		0.000		10	2.888E-120		1.581E-119	
30	0.000	1.121E+04	0.000	3.751E+03	30	1.529E-179	1.815E-01	0.000	7.788E-2
50	0.000		0.000		50	1.059E-216		0.000	
F5	AVG		STD		F6	AVG		STD	
	CMSRAS	SRA	CMSRAS	SRA		CMSRAS	SRA	CMSRAS	SRA
10	2.218E-28		1.158E-27		10	1.791E-10		2.306E-10	
30	0.000	6.487	0.000	11.687	30	3.483E-14	8.791E-2	5.4E-14	8.558E-2
50	0.000		0.000		50	1.219E-16		1.663E-16	
F7	AVG		STD						
	CMSRAS	SRA	CMSRAS	SRA					
10	1.782E-04		1.818E-04						
30	1.241E-04	0.4132	1.232E-04	0.6895					
50	4.832E-05		3.088E-05						

TABLE 10. Results on unimodal benchmark functions by different dimensions.

F1	AVG		STD		F2	AVG		STD	
	CMSRAS	SRA	CMSRAS	SRA		CMSRAS	SRA	CMSRAS	SRA
30	0.000	8.692E-04	0.000	6.719E-04	30	4.359E-201	9.27E-03	0.000	3.074E-03
100	0.000	1.124E-01	0.000	2.511E-02	100	1.096E-198	2.267E-01	0.000	2.096E-02
200	0.000	8.517E-01	0.000	1.601E-01	200	2.221E-196	8.554E-01	0.000	6.483E-02
F3	AVG		STD		F4	AVG		STD	
	CMSRAS	SRA	CMSRAS	SRA		CMSRAS	SRA	CMSRAS	SRA
30	0.000	1.127E+04	0.000	4.976E+03	30	9.054E-176	2.097E-01	0.000	1.167E-01
100	0.000	1.282E+05	0.000	2.437E+04	100	8.087E-46	2.2008	3.961E-45	1.097
200	0.000	4.858e+05	0.000	8.012e+04	200	9.922E-27	8.064	5.434E-26	2.942
F5	AVG		STD		F6	AVG		STD	
	CMSRAS	SRA	CMSRAS	SRA		CMSRAS	SRA	CMSRAS	SRA
30	0.000	5.703	0.000	10.124	30	2.519E-16	8.009E-02	6.021E-16	7.0541E-02
100	0.000	9.151	0.000	13.005	100	2.441E-02	8.09104	7.217E-03	9.9801E-01
200	0.000	8.541	0.000	12.287	200	2.647	30.527	0.611	1.5957
F7	AVG		STD						
	CMSRAS	SRA	CMSRAS	SRA					
30	6.636E-05	4.3311E-01	7.847E-05	6.6933E-01					
100	7.503E-05	5.368E-01	8.121E-05	2.7872E-01					
200	5.832E-05	7.6049E-01	4.778E-05	3.1853E-01					

by the 12 unimodal benchmark functions (F₁-F₇ and F₂₈-F₃₂), respectively. In this experiment, the maximum number of iterations and calculated times was set to 2000 and 30, respectively. The experiment is divided into two parts: in the first part, for F₁-F₇ benchmark functions, the dimension of search space was set to 30 and the population size of CMSRAS was set to 10, 30, 50, respectively. Meanwhile, the population size of CMSRAS was set to 30, and the dimension of search space was set to 30, 100, 200, respectively. In the second part, for F₂₈-F₃₂ benchmark functions, under a fixed dimension, the population size of CMSRAS was set to 10, 30, 50, respectively. The average optimum (AVG) and

standard deviation (STD) of the attained results over 30 times independent runs as shown in Table 9-11, respectively. The convergence curves of unimodal benchmark functions by the different population sizes and dimensions are shown in Fig. 17 and 18, respectively.

As it can be seen from Table 9 and 11, as the population increases, for F₁, F₃, F₅, F₂₈, F₂₉, F₃₀, F₃₁ and F₃₂, CMSRAS obtains the global optimum and the STD is respectively smaller. In addition, for F₁-F₇ and F₂₈-F₃₂, the AVG and STD obtained by CMSRAS are getting better and better. However, the AVG and STD obtained by SRA are worse than that of CMSRAS, and it is easy to fall into local optimum.

TABLE 11. Results on fixed dimension unimodal benchmark functions by different population size.

F28	AVG		STD		F29	AVG		STD	
	CMSRAS	SRA	CMSRAS	SRA		CMSRAS	SRA	CMSRAS	SRA
10	5.0804E-02		0.1933		10	0.000		0.000	
30	0.000	0.6838	0.000	0.879	30	0.000	8.643E-04	0.000	1.501E-03
50	0.000		0.000		50	0.000		0.000	
F30	AVG		STD		F31	AVG		STD	
	CMSRAS	SRA	CMSRAS	SRA		CMSRAS	SRA	CMSRAS	SRA
10	7.657E-244		0.000		10	-0.0037912		1.764E-18	
30	0.000	5.275E-03	0.000	8.345E-03	30	-0.0037912	-0.002738	1.764E-18	3.692E-03
50	0.000		0.000		50	-0.0037912		1.764E-18	
F32	AVG		STD						
	CMSRAS	SRA	CMSRAS	SRA					
10	2.379E-21		1.22E-20						
30	1.824E-31	0.7689	9.096E-31	1.224					
50	0.000		0.000						

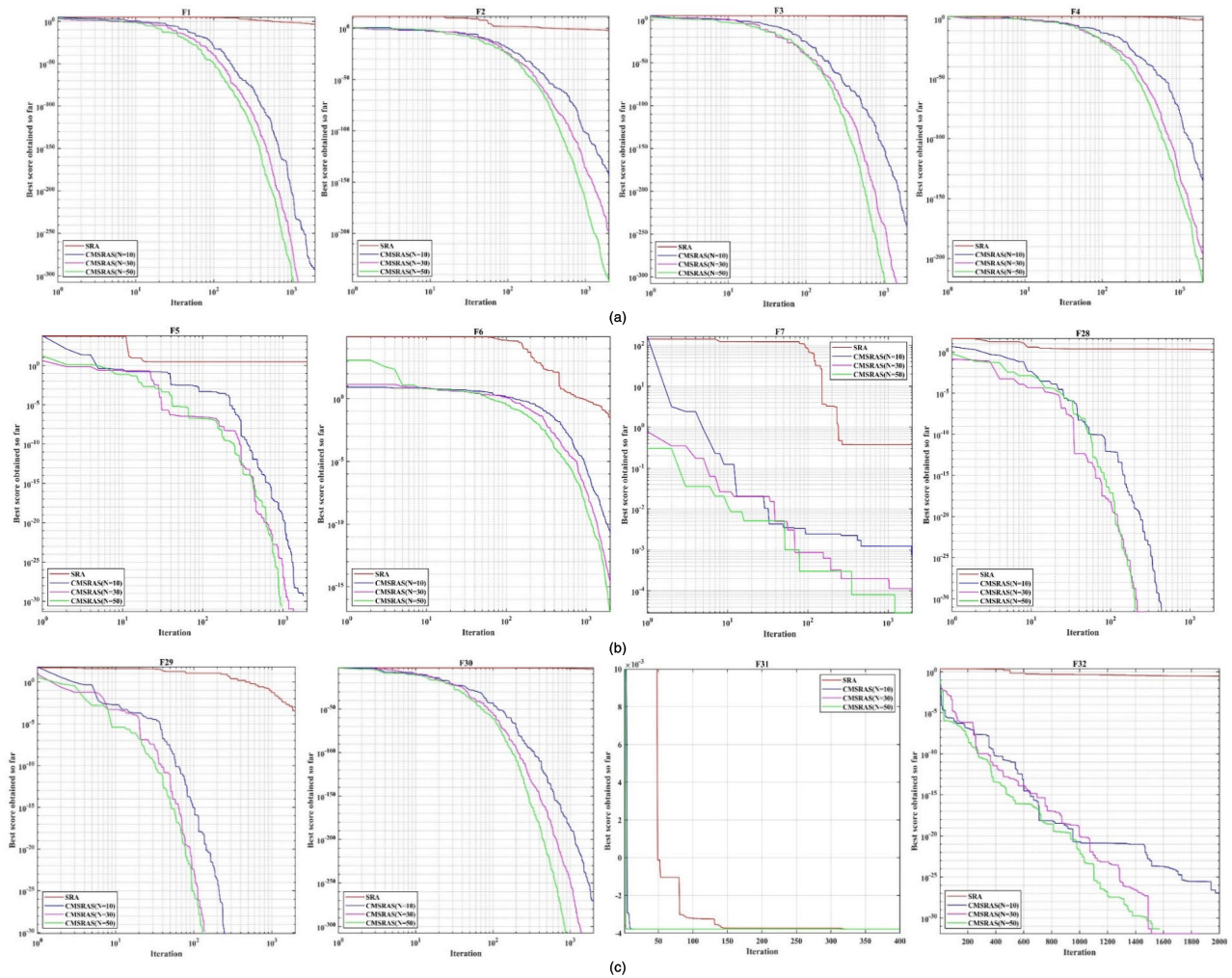


FIGURE 17. The convergence curves of unimodal benchmark functions by different population sizes.

The unimodal benchmark functions are more challenging to be addressed as the dimension of search space increases. As per results in Table 10, as the dimension increases, for F₁, F₃ and F₅, CMSRAS can obtain the global optimum

and the STD is 0. Meanwhile, for F₁-F₇, the AVG and STD of CMSRAS is getting worse and worse but better than that of SRA with the increase of the dimension of search space.

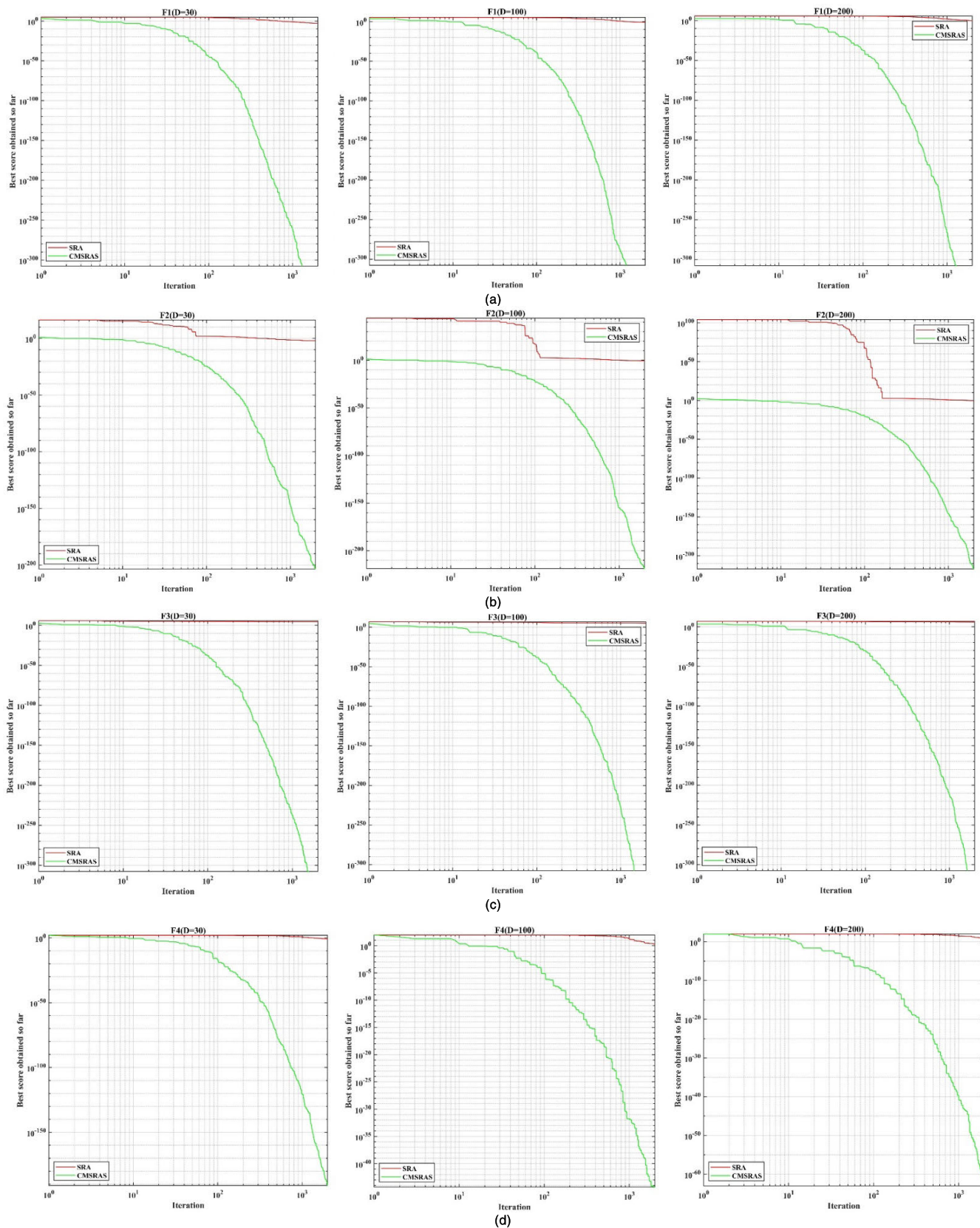


FIGURE 18. The convergence curves of unimodal benchmark functions by different dimensions.

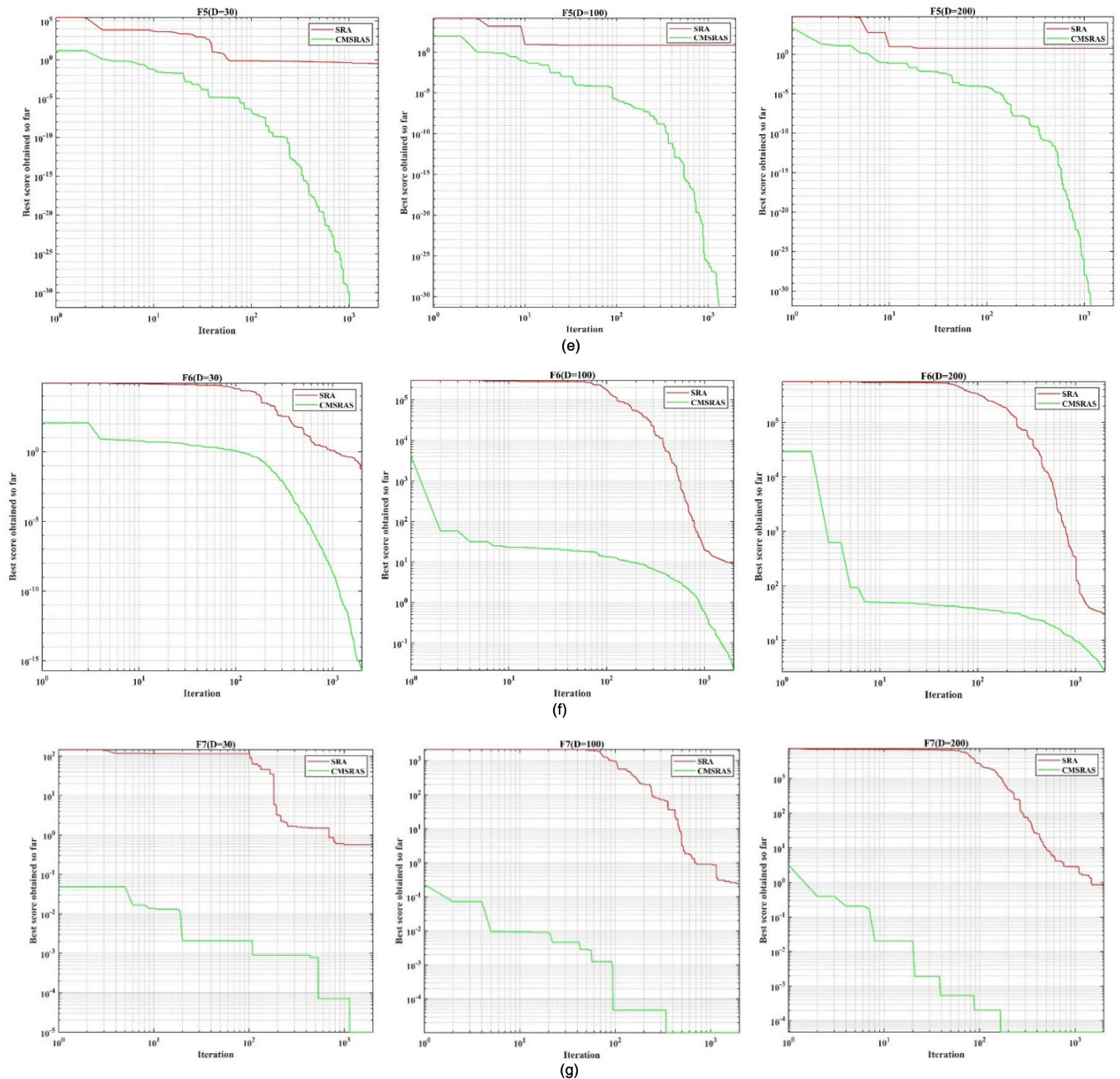


FIGURE 18. (Continued.) The convergence curves of unimodal benchmark functions by different dimensions.

As it can be seen in Fig. 17 and 18, for all unimodal benchmark functions, the convergence rate of CMSRAS is faster than that of SRA. With the increase of population size, the convergence rate of CMSRAS is gradually increased. Inversely, as the dimension of search space increases, the convergence rate of CMSRAS is gradually decreased for most of unimodal benchmark functions. The results of convergence curves show that SRA is more easily to fall into local optimum than CMSRAS.

To sum up, the exploitation ability of CMSRAS is superior to that of SRA, whether in low dimensions or in high

dimensions. That may be because the combination of population strategy with shared node, improved tent chaotic mutation strategy and Gaussian mutation strategy increases the population diversity, making CMSRAS more likely to jump out of the local optimum and obtain a better solution with a fast convergence rate.

(II) CMSRAS compared with traditional and advanced algorithms

In this section, in order to evaluate performance of CMSRAS compared with other algorithms more comprehensively, the exploitation ability test was performed

TABLE 12. Comparison results of unimodal functions with traditional algorithms.

Algorithms	F1		F2		F3	
	AVG	STD	AVG	STD	AVG	STD
CMSRAS	0.000	0.000	1.735E-196	0.000	0.000	0.000
PSO	1.05E-02	1.021E-02	6.527E-02	3.993E-02	3.351E+02	1.112E+01
CS	4.929E-10	3.944E-10	1.802E-04	1.152E-04	1.465E+01	7.192
DA	6.092E+02	3.385E+02	1.388E+01	4.764	7.402E+03	5.395E+03
GWO	8.823E-122	2.259E-121	8.367E-71	1.292E-70	1.13E-32	5.551E-32
MFO	4.11E-11	2.12E-10	8.647	1.001E+01	1.276E+04	9.843E+03
HHO	0.000	0.000	1.202E-189	0.000	1.367E-259	0.000
MVO	7.696E-02	3.133E-02	2.021E-01	5.752E-02	9.68	4.394
SMA	0.000	0.000	4.986E-238	0.000	0.000	0.000
SCA	4.593E-07	1.889E-06	3.008E-11	5.984E-11	1.196E+03	1.662E+03
SOA	0.000	0.000	0.000	0.000	2.134E+03	2.851E+03
WOA	4.411E-311	0.000	4.582E-212	0.000	7.538E+03	5.279E+03
Algorithms	F4		F5		F6	
	AVG	STD	AVG	STD	AVG	STD
CMSRAS	4.762E-255	0.000	0.000	0.000	1.178E-13	3.170E-13
PSO	1.272E+00	2.213E-01	2.286E-19	1.047E-18	7.235E-03	5.657E-03
CS	2.969E+00	1.255E+01	0.000	0.000	4.396E-10	2.858E-10
DA	1.633E+01	5.1784E+01	8.167E-01	3.708E+00	5.129E+02	1.909E+02
GWO	1.274E-29	2.64E-29	5.592E-08	5.722E-08	4.352E-01	3.475E-01
MFO	6.3044+01	9.287E+00	1.468E-02	6.148E-02	1.684E-09	8.237E-09
HHO	2.179E-181	0.000	3.078E-09	1.161E-08	8.849E-06	1.182E-05
MVO	4.571E-01	1.594E-01	5.529E-06	6.811E-06	7.652E-02	2.091E-02
SMA	2.226E-221	0.000	3.431E-08	7.372E-08	2.543E-04	9.538E-05
SCA	5.161E+00	6.288E+00	3.929E-04	4.782E-04	4.213E+00	3.511E-01
SOA	1.926E-95	1.055E-94	1.141E-01	1.956E-01	1.728E-01	8.931E-01
WOA	2.961E+01	2.713E+01	2.026E-08	3.31E-08	2.224E-03	1.494E-03
Algorithms	F7					
	AVG	STD				
CMSRAS	8.230E-05	7.025E-05				
PSO	4.212E-01	2.512E-01				
CS	2.514E-05	9.203E-03				
DA	1.804E-01	1.096E-01				
GWO	3.438E-04	2.137E-04				
MFO	1.585E+00	4.231E+00				
HHO	3.678E-05	3.259E-05				
MVO	1.259E-02	4.655E-03				
SMA	5.83E-05	4.276E-05				
SCA	1.291E-01	1.141E-02				
SOA	5.976E-05	5.797E-05				
WOA	6.268E-04	5.995E-04				

by the 12 unimodal benchmark functions (F₁-F₇ and F₂₈-F₃₂), respectively. In this test, the maximum number of iterations, the dimension of search space and the population size were set to 2000, 30 and 30 respectively. The average optimum (AVG) and standard deviation (STD) the attained results over 30 times independent runs as shown in Table 12-15, respectively. The convergence curves of unimodal benchmark functions by traditional and advanced algorithms are shown in Fig. 21 and 22, respectively.

As it can be seen in Table 12 and 14, for F₁, the CMSRAS, HHO, SMA and SOA can obtain the global optimum with the best STD, and the AVG and STD of CMSRAS are superior to PSO, CS, DA, GWO, MFO, MVO, SCA and WOA, respectively. For F₂, SMA can obtain the global optimum is superior to other traditional algorithms. Meanwhile, the AVG and STD of CMSRAS are superior to PSO, CS, DA, GWO, MFO, MVO, HHO and SOA, respectively. For F₃, both CMSRAS and SMA can obtain the global optimum with the best STD, which is superior to PSO, CS, DA, GWO, MFO, MVO, SCA,

TABLE 13. Comparison results of unimodal functions with advanced algorithms.

Algorithms	F1		F2		F3	
	AVG	STD	AVG	STD	AVG	STD
CMSRAS	0.000	0.000	2.915E-189	0.000	0.000	0.000
TAPSO	1.669E-12	8.379E-12	8.085E-05	2.093E-04	1.599E-01	2.451E-01
MPSO	4.052E-11	1.268E-10	2.333E+01	6.261E+00	1.627E-01	2.117E-01
IPSO	1.676E-13	2.953E-13	3.334E-01	1.825E+00	2.904E-01	6.782E-01
I-GWO	4.540E-124	1.61E-123	3.266E-75	4.928E-75	2.001E-23	7.421E-23
AGPSO1	1.598E-10	3.349E-10	6.711E-01	2.535E+00	1.559E+00	2.304E+00
AGPSO2	7.549E-10	3.356E-09	1.004E+00	3.049E+00	6.640E-01	4.92E-01
AGPSO3	1.819E-13	4.496E-13	9.468E-03	2.335E-03	2.062E-01	3.063E-01
GWOCs	9.125E-123	2.114E-122	2.152E-71	1.844E-71	1.647E-33	7.513E-33
Algorithms	F4		F5		F6	
	AVG	STD	AVG	STD	AVG	STD
CMSRAS	6.174E-253	0.000	0.000	0.000	1.171E-12	1.44E-12
TAPSO	6.652E-02	3.083E-02	0.000	0.000	1.5E-11	7.353E-11
MPSO	1.043E-01	4.394E-02	0.000	0.000	1.57E-12	5.441E-12
IPSO	6.857E-02	3.109E-02	0.000	0.000	5.989E-13	2.368E-12
I-GWO	1.465E-23	3.065E-23	1.831E-13	3.143E-13	3.067E-11	2.018E-11
AGPSO1	3.837E-01	1.789E-01	0.000	0.000	6.34E-10	1.934E-09
AGPSO2	2.522E-01	1.165E-01	0.000	0.000	6.303E-10	3.032E-09
AGPSO3	8.939E-01	4.308E-01	0.000	0.000	1.761E-13	6.953E-13
GWOCs	2.742E-30	5.9E-30	6.802E-08	7.201E-08	5.391E-01	2.279E-01
Algorithms	F7					
	AVG	STD				
CMSRAS	6.403E-05	1.69E-05				
TAPSO	2.053E-02	6.996E-03				
MPSO	3.798E-01	1.534E+00				
IPSO	2.058E-02	8.326E-03				
I-GWO	6.343E-04	2.209E-04				
AGPSO1	2.979E-01	1.469E+00				
AGPSO2	2.831E-02	1.324E-02				
AGPSO3	2.562E-02	1.045E-02				
GWOCs	3.455E-04	1.533E-04				

HHO, WOA and SOA, respectively. For F_5 and F_{32} , both CMSRAS and CS can obtain the global optimum with the best STD, which is superior to other algorithms. For F_7 , the HHO can obtain the best results, and the AVG and STD of CMSRAS is superior to PSO, DA, GWO, MFO, MVO, SCA and WOA, respectively. For F_4 and F_5 , CMSRAS can attain the best results compared with other algorithms. For F_{28} and F_{29} , CMSRAS, PSO and CS can attain the global optimum with the best STD, which is superior to other algorithms. For F_{30} , the global optimum is obtained by CMSRAS, GWO, HHO, SMA, SOA and WOA. For F_{31} , the AVG and STD of SMA are ranked first. As per results in Table 13 and 15, for F_1 , F_2 , F_3 , F_4 and F_7 , the results of CMSRAS are optimal. For F_{28} , F_{29} and F_{31} , the AVG and STD of CMSRAS are the smallest in parallel compared with other algorithms, but that is better than GWOCs. For F_{30} , CMSRAS, I-GWO and GWOCs can attain the best results and that is better than other algorithms. For F_{31} , the AVG and STD of CMSRAS are the smallest in parallel compared with other algorithms, but

that is better than I-GWO, AGPSO1, AGPSO2, AGPSO3 and GWOCs, respectively.

As it can be seen in Fig. 21 and 22, it is visually observed that CMSRAS has the fastest convergence rate than other competitive algorithms in F_3 , F_4 , F_5 , F_6 , F_7 , F_{28} , F_{29} and F_{31} , respectively. All in all, compared with traditional and advanced algorithms, CMSRAS has an advantage in unimodal functions.

D. EXPLORATION COMPETENCE ANALYSIS

(I) CMSRAS compared with SRA

The ability to exploit and avoid falling into local optimum is often evaluated by multimodal functions. In this section, in order to compare performance of CMSRAS with basic SRA more comprehensively, the experiments were performed by the 20 multimodal benchmark functions (F_8 - F_{27}). In this experiment, the maximum number of iterations and calculated times was set to 2000 and 30, respectively. The experiments were divided into two parts: in the first part, for

TABLE 14. Comparison results of fixed dimension unimodal functions with traditional algorithms.

Algorithms	F28		F29		F30	
	AVG	STD	AVG	STD	AVG	STD
CMSRAS	0.000	0.000	0.000	0.000	0.000	0.000
PSO	0.000	0.000	0.000	0.000	8.274E-56	4.516E-55
CS	0.000	0.000	0.000	0.000	5.928E-64	3.243E-63
DA	2.815E-04	1.527E-03	1.931E-05	6.335E-05	1.283E-07	3.182E-07
GWO	5.08E-02	1.933E-01	2.519E-08	2.808E-08	0.000	0.000
MFO	3.573E-20	2.808E-20	1.028E-19	9.833E-20	7.678E-122	4.205E-121
HHO	8.379E-13	3.260E-12	2.469E-06	3.652E-06	0.000	0.000
MVO	1.524E-01	3.1E-01	4.357E-08	4.496E-08	2.906E-10	2.922E-10
SMA	3.139E-10	6.231E-10	2.027E-10	4.241E-10	0.000	0.000
SCA	6.259E-05	6.055E-05	2.331E-04	2.420E-04	3.722E-215	0.000
SOA	2.032E-01	3.427E-01	4.578E-01	7.809E-01	0.000	0.000
WOA	5.411E-12	1.216E-11	2.052E-05	1.777E-05	0.000	0.000
Algorithms	F31		F32			
	AVG	STD	AVG	STD		
CMSRAS	-3.79E-02	1.764E-18	0.000	0.000		
PSO	-3.79E-02	1.764E-18	7.227E-15	3.282E-14		
CS	-3.79E-02	1.764E-18	0.000	0.000		
DA	-3.79E-02	8.843E-07	1.049E-03	1.762E-03		
GWO	-3.79E-02	2.641E-11	1.254E-07	1.647E-07		
MFO	-3.79E-02	1.764E-18	4.058E-03	5.945E-03		
HHO	-3.79E-02	1.481E-11	3.732E-07	1.155E-06		
MVO	-3.79E-02	1.796E-09	1.933E-08	3.228E-08		
SMA	-3.79E-02	1.50E-18	1.386E-08	2.462E-08		
SCA	-3.79E-02	2.323E-10	7.328E-05	7.429E-05		
SOA	-3.67E-02	1.808E-04	3.021E-03	8.280E-03		
WOA	-3.79E-02	1.530E-10	1.429E-06	3.507E-06		

TABLE 15. Comparison results of fixed dimension unimodal functions with advanced algorithms.

Algorithms	F28		F29		F30	
	AVG	STD	AVG	STD	AVG	STD
CMSRAS	0.000	0.000	0.000	0.000	0.000	0.000
TAPSO	0.000	0.000	0.000	0.000	8.014E-129	3.464E-128
MPSO	0.000	0.000	0.000	0.000	8.086E-117	2.528E-116
IPSO	0.000	0.000	0.000	0.000	2.967E-115	1.470E-114
I-GWO	0.000	0.000	0.000	0.000	0.000	0.000
AGPSO1	0.000	0.000	0.000	0.000	2.649E-107	1.447E-106
AGPSO2	0.000	0.000	0.000	0.000	4.282E-110	2.142E-109
AGPSO3	0.000	0.000	0.000	0.000	3.491E-179	0.000
GWCS	2.540E-02	1.391E-02	3.098E-08	2.843E-08	0.000	0.000
Algorithms	F31		F32			
	AVG	STD	AVG	STD		
CMSRAS	-3.79E-02	1.764E-18	0.000	0.000		
TAPSO	-3.79E-02	1.764E-18	0.000	0.000		
MPSO	-3.79E-02	1.764E-18	0.000	0.000		
IPSO	-3.79E-02	1.764E-18	0.000	0.000		
I-GWO	-3.79E-02	1.764E-18	1.365E-11	1.894E-11		
AGPSO1	-3.79E-02	1.764E-18	9.522E-27	2.832E-26		
AGPSO2	-3.79E-02	1.764E-18	1.643E-33	9.001E-33		
AGPSO3	-3.79E-02	1.764E-18	1.643E-33	9.001E-33		
GWCS	-3.79E-02	3.091E-11	9.689E-08	1.032E-07		

F₈-F₁₃ benchmark functions, the population size of CMSRAS was set to 10, 30, 50, respectively. and then the dimension of search space was set to 30, 100, 200,

respectively. In the second part, for F₁₄-F₂₇ benchmark functions, under a fixed dimension, the population size of CMSRAS was set to 10, 30, 50, respectively. The AVG

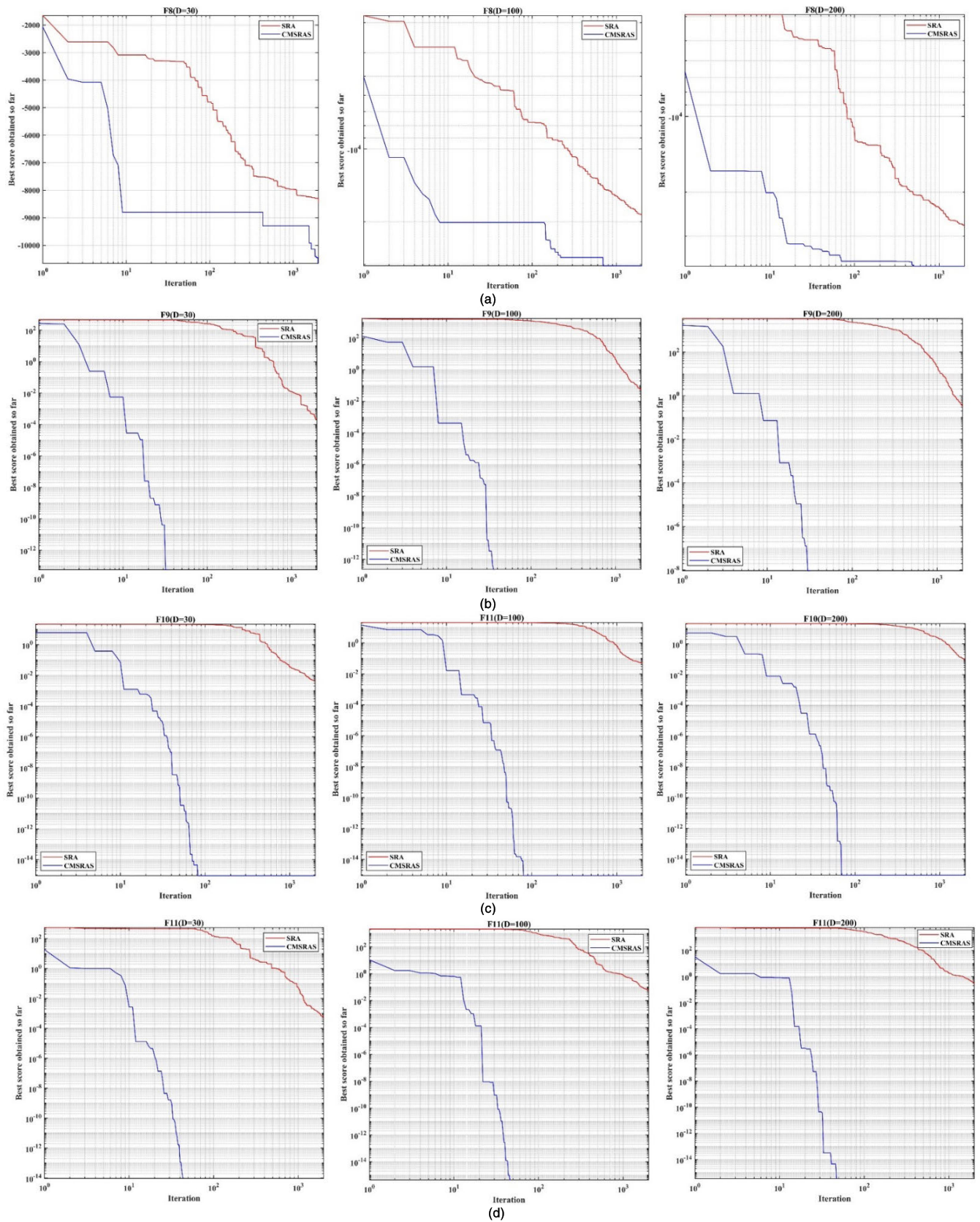


FIGURE 19. The convergence curves of multimodal benchmark functions by different dimensions.

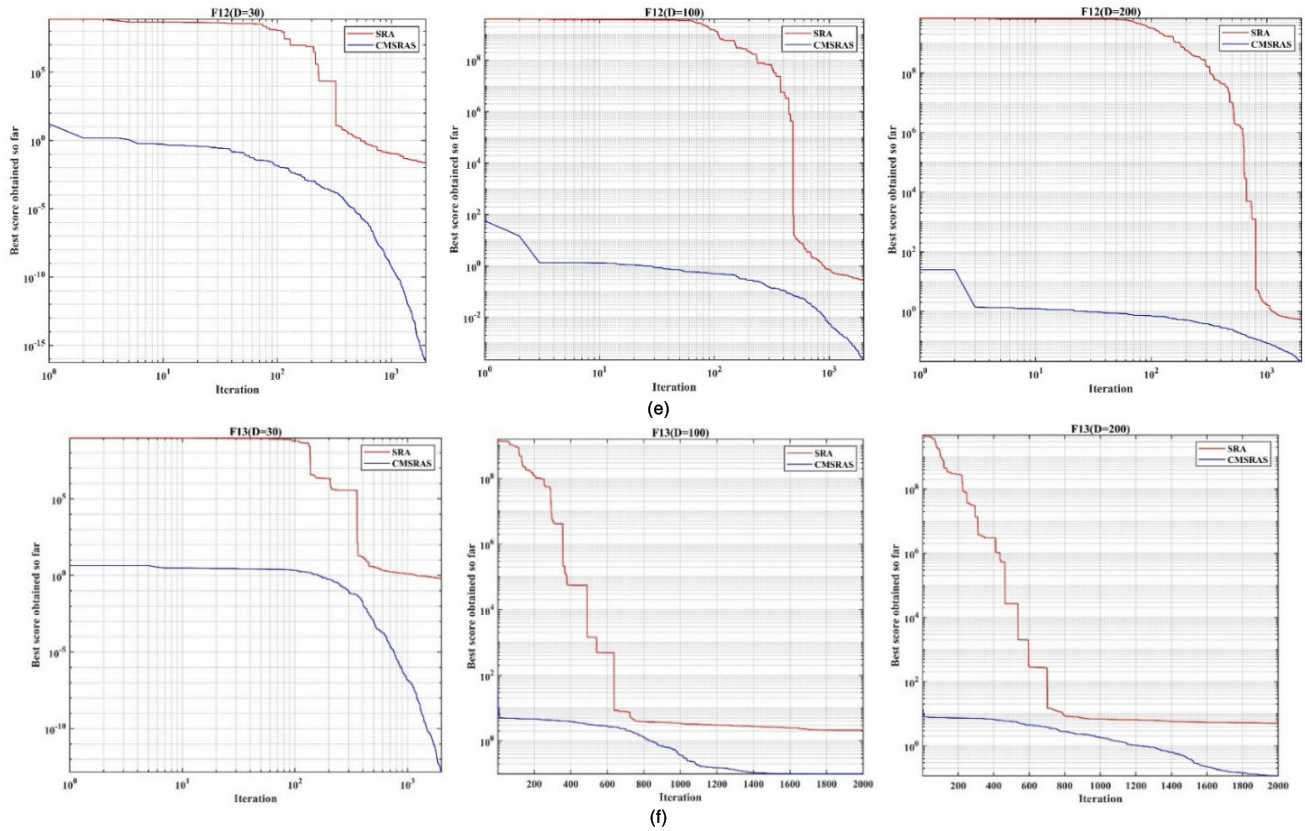


FIGURE 19. (Continued.) The convergence curves of multimodal benchmark functions by different dimensions.

TABLE 16. Results on multimodal benchmark functions by different population size.

F8	AVG		STD		F9	AVG		STD	
	CMSRAS	SRA	CMSRAS	SRA		CMSRAS	SRA	CMSRAS	SRA
10	-9.181E+03		1.451E+02		10	0.000		0.000	
30	-1.040E+04	-7.962E+03	1.244E+02	8.006E+02	30	0.000	4.682E-04	0.000	2.931E-04
50	-1.095E+04		1.220E+02		50	0.000		0.000	
F10	AVG		STD		F11	AVG		STD	
	CMSRAS	SRA	CMSRAS	SRA		CMSRAS	SRA	CMSRAS	SRA
10	8.8818E-16		0.000		10	0.000		0.000	
30	8.8818E-16	6.732E-03	0.000	2.525E-03	30	0.000	2.622E-02	0.000	2.7594E-02
50	8.8818E-16		0.000		50	0.000		0.000	
F12	AVG		STD		F13	AVG		STD	
	CMSRAS	SRA	CMSRAS	SRA		CMSRAS	SRA	CMSRAS	SRA
10	1.8377E-12		2.4719E-12		10	1.919E-11		3.891E-12	
30	7.4797E-16	2.0808E-02	1.3146E-15	2.557E-02	30	2.087E-15	2.901E-01	3.473E-14	3.601E-01
50	1.4982E-18		3.403E-18		50	6.141E-16		2.854E-16	

and STD of the attained results over 30 times independent runs as shown in Table 16-18, respectively. The convergence curves of multimodal benchmark functions by different population sizes and dimensions are shown in Fig. 19 and 20, respectively.

The data in Table 16 and 18 demonstrates that CMSRAS obtains the global optimum with lowest STD in F9, F11, F14, F16, F17, F18, F19, F20, F21, F22, F23, F26 and F27, respectively. However, SRA can attain the global optimum in F16, F17 and F19, the STD of SRA is less than CMSRAS.

As the population increases, the AVG and STD obtained by CMSRAS are getting better and better in F8, F12, F13 and F15, respectively. These results show that CMSRAS can still exhibit significant advantages compared to SRA, such as ranking first among other multimodal benchmark functions other than F16 and F27.

As per results in Table 17, with the increase of the dimension of search space, for F9 and F11, CMSRAS can obtain the global optimum and the STD is 0. Meanwhile, for F12 and F13, the AVG and STD of CMSRAS is getting worse and

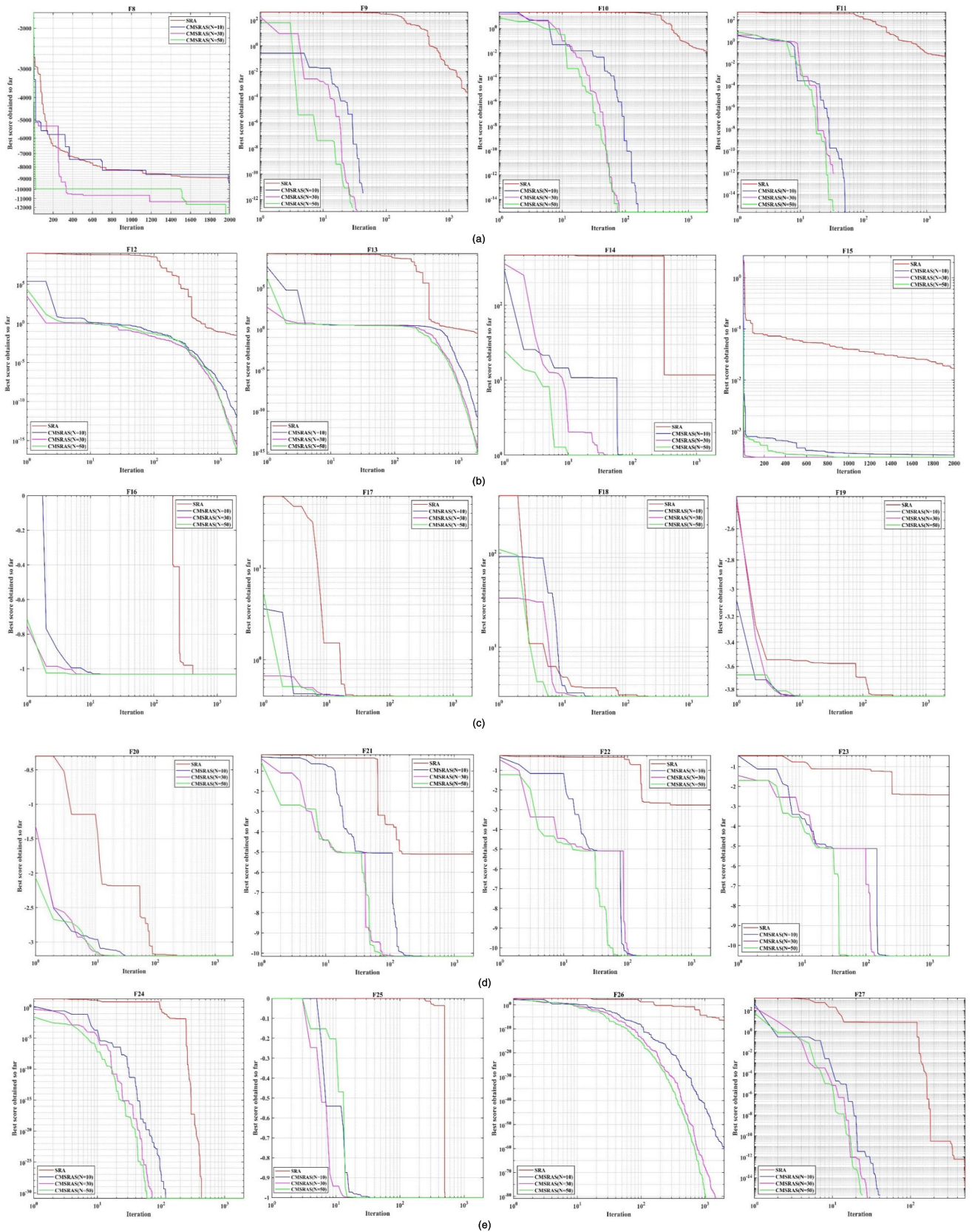


FIGURE 20. The convergence curves of multimodal benchmark functions by different population sizes.

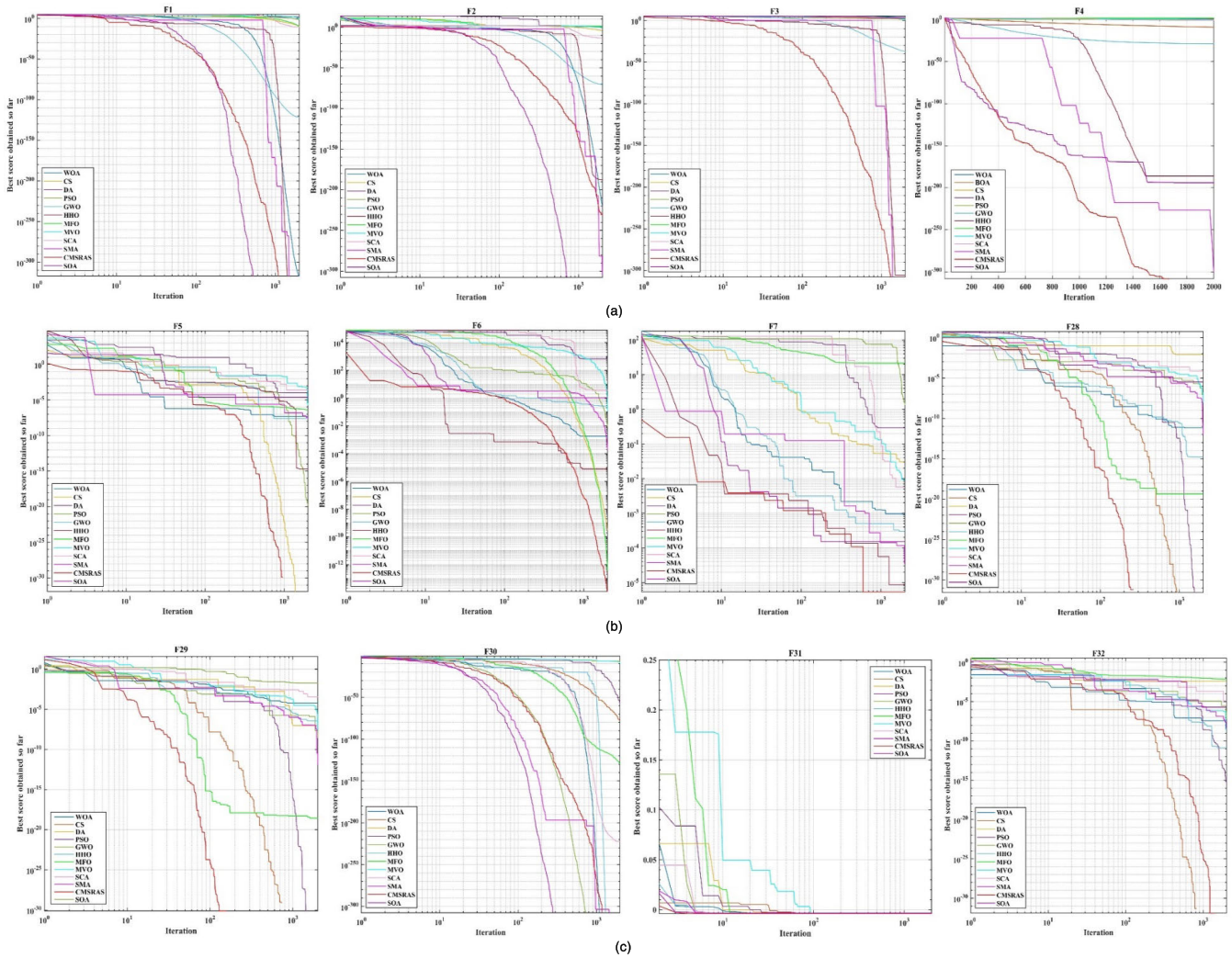


FIGURE 21. The convergence curves of unimodal benchmark functions by traditional algorithms.

TABLE 17. Results on multimodal benchmark functions by different dimensions.

F8	AVG		STD		F9	AVG		STD	
	CMSRAS	SRA	CMSRAS	SRA		CMSRAS	SRA	CMSRAS	SRA
30	-1.06E+04	-7.91E+03	2.26E+02	7.58E+02	30	0.000	4.319E-04	0.000	2.861E-04
100	-3.12E+04	-1.83E+04	6.47E+02	1.25E+03	100	0.000	6.411E-02	0.000	1.034E-02
200	-3.72E+04	-2.67E+04	3.47E+02	1.36E+03	200	0.000	4.551E-01	0.000	1.2136E-01
F10	AVG		STD		F11	AVG		STD	
	CMSRAS	SRA	CMSRAS	SRA		CMSRAS	SRA	CMSRAS	SRA
30	8.882E-16	7.036E-03	0.000	2.192E-03	30	0.000	2.107E-02	0.000	2.634E-02
100	8.882E-16	4.868E-02	0.000	4.445E-03	100	0.000	7.087E-02	0.000	2.1589E-02
200	8.882E-16	1.016E-01	0.000	8.528E-03	200	0.000	2.334E-01	0.000	3.998E-02
F12	AVG		STD		F13	AVG		STD	
	CMSRAS	SRA	CMSRAS	SRA		CMSRAS	SRA	CMSRAS	SRA
30	6.058E-16	2.685E-02	1.2196E-15	3.542E-02	30	7.0668E-12	0.4592	1.0311E-11	0.1608
100	1.909E-04	2.748E-01	7.921E-05	4.521E-02	100	1.244E-06	2.0482	1.629E-06	3.106E-01
200	1.633E-02	6.117E-01	4.766E-02	5.449E-02	200	5.688E-02	18.108	8.4007E-02	2.949E-01

worse but better than that of SRA with the increase of the dimension of search space. Especially for F₈-F₁₃, the AVG and STD of CMSRAS is better than SRA. In other words,

the ability of both exploration and avoiding falling into the local optimum of CMSRAS is superior to SRA. That may be because the reversing behavior and disturbance behavior of

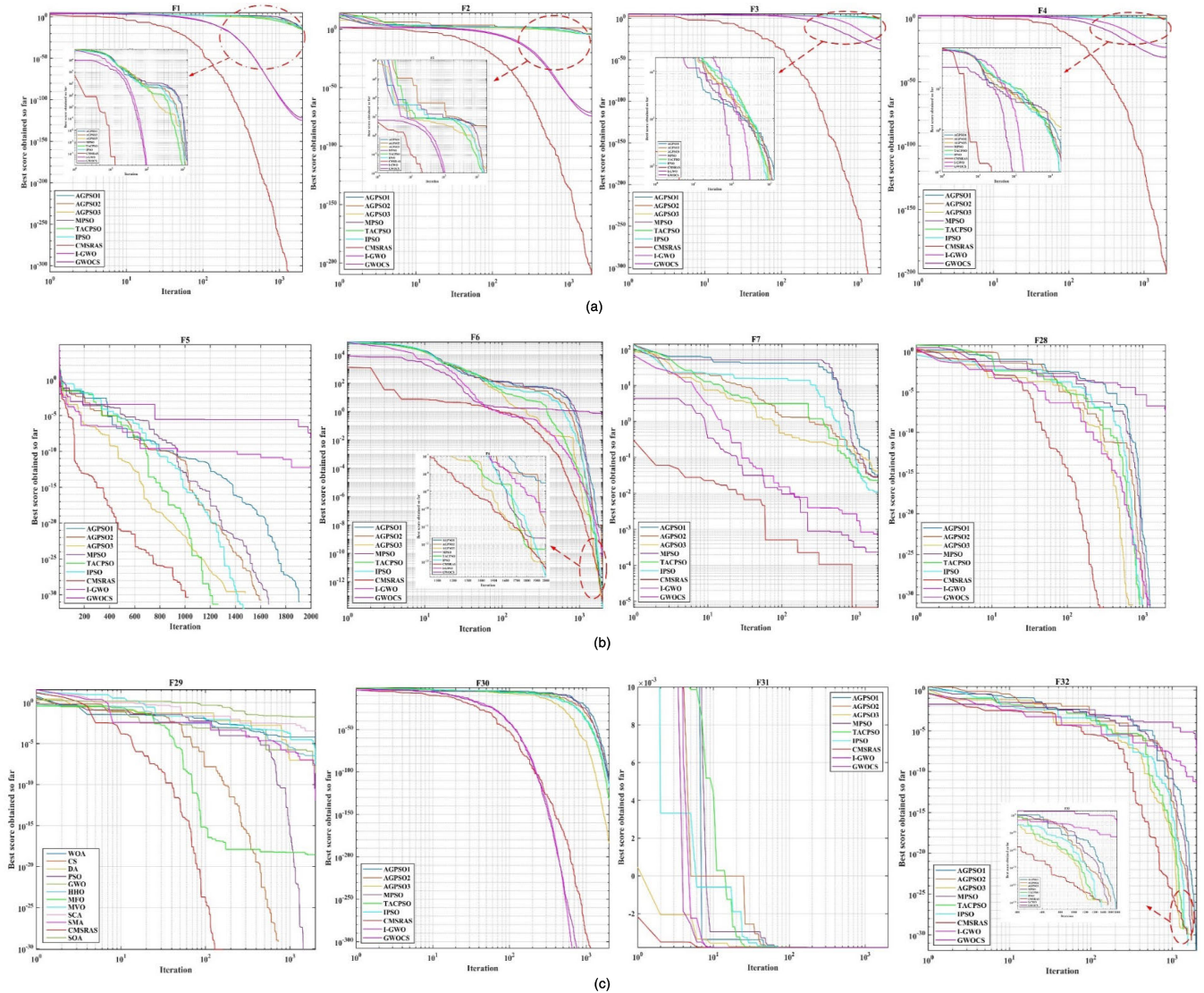


FIGURE 22. The convergence curves of unimodal benchmark functions by advanced algorithms.

multiple mirrors can refine the local optimum and improve the search ability of the optimal solution, and to improve the global convergence efficiency and accuracy of the algorithm and avoid falling into the local optimum.

As it can be seen in Fig. 19 and 20, for F₈-F₂₇, the convergence rate of CMSRAS is faster than that of SRA. With the increase of population size, the convergence rate of CMSRAS is gradually increased. That may be because the population diversity of CMSRAS is enriched by the selection behavior of a shared node. As the dimension of search space increases, the convergence rate of CMSRAS is gradually decreased. The results of convergence curves show that SRA is more easily to fall into local optimum than CMSRAS.

In conclusion, for high-dimensional multimodal functions, CMSRAS is obviously better than SRA in terms of in terms of the ability to exploit and avoid falling into local optimum.

(II) CMSRAS compared with traditional and advanced algorithms

In this section, in order to compare performance of CMSRAS with other algorithms more comprehensively, the experiments were performed by the 20 multimodal benchmark functions (F₈-F₂₇) to evaluate the ability of exploration and avoiding the local optimum. Initially, the maximum number of iterations, the dimensions and the population size were set to 2000, 30 and 30 respectively. The AVG and STD of the attained results over 30 times independent runs as shown in Table 19-22. The convergence curves of multimodal benchmark functions by the traditional and advanced algorithms are shown in Fig. 23 and 24, respectively.

Compared with traditional and advanced algorithms, the data in Table 19-22 represents that CMSRAS is still competitive in multimodal benchmark functions. As it can be seen in Table 19 and 21, for F₉, F₁₀, F₁₁, F₁₄, F₁₆, F₁₉, F₂₄,

TABLE 18. Results on fixed dimension multimodal benchmark functions by different population size.

F14	AVG		STD		F15	AVG		STD	
	CMSRAS	SRA	CMSRAS	SRA		CMSRAS	SRA	CMSRAS	SRA
10	1.0641		3.6224E-01		10	5.1841E-04		6.6274E-04	
30	0.998	9.737	0.000	4.2088	30	4.8279E-04	2.064E-02	6.5133E-04	2.3299E-02
50	0.998		0.000		50	3.38E-04		1.6718E-04	
F16	AVG		STD		F17	AVG		STD	
	CMSRAS	SRA	CMSRAS	SRA		CMSRAS	SRA	CMSRAS	SRA
10	-1.0316		6.712E-16		10	0.3978		0.000	
30	-1.0316	-1.0316	6.7752E-16	4.8261E-16	30	0.3978	0.3978	0.000	3.0717E-06
50	-1.0316		6.7752E-16		50	0.3978		0.000	
F18	AVG		STD		F19	AVG		STD	
	CMSRAS	SRA	CMSRAS	SRA		CMSRAS	SRA	CMSRAS	SRA
10	3.000		6.0599E-16		10	-3.86		9.362E-16	
30	3.000	3.000	1.3297E-16	2.9608E-04	30	-3.86	-3.86	9.362E-16	4.44E-16
50	3.000		1.3168E-16		50	-3.86		9.362E-16	
F20	AVG		STD		F21	AVG		STD	
	CMSRAS	SRA	CMSRAS	SRA		CMSRAS	SRA	CMSRAS	SRA
10	-3.32		0.000		10	-10.03		0.47218	
30	-3.32	-3.238	0.000	5.743E-02	30	-10.1532	-5.97287	1.7763E-15	3.3703
50	-3.32		0.000		50	-10.1532		1.4503E-15	
F22	AVG		STD		F23	AVG		STD	
	CMSRAS	SRA	CMSRAS	SRA		CMSRAS	SRA	CMSRAS	SRA
10	-9.9576		1.838		10	-9.8663		2.119	
30	-10.4028	-4.6646	5.713E-16	2.7715	30	-10.5364	-3.4495	1.899E-15	2.5845
50	-10.4028		1.189E-15		50	-10.5364		1.6747E-15	
F24	AVG		STD		F25	AVG		STD	
	CMSRAS	SRA	CMSRAS	SRA		CMSRAS	SRA	CMSRAS	SRA
10	1.3497E-31		6.6808E-47		10	-0.8333		0.379	
30	1.3497E-31	0.1303	6.6808E-47	0.4961	30	-1	-0.7	0.000	0.483
50	1.3497E-31		6.6808E-47		50	-1		0.000	
F26	AVG		STD		F27	AVG		STD	
	CMSRAS	SRA	CMSRAS	SRA		CMSRAS	SRA	CMSRAS	SRA
10	3.0119E-57		1.6096E-56		10	0.000		0.000	
30	0.000	5.01E-05	0.000	2.182E-04	30	0.000	0.000	0.000	0.000
50	0.000		0.000		50	0.000		0.000	

F25, F26 and F27, the AVG and STD of CMSRAS were the smallest in parallel compared with other algorithms, but that was better than most traditional algorithms. For F12, F13, F20, F21, F22 and F23, it shows that the results of CMSRAS are optimal, which indicates that CMSRAS can still maintain its advantages compared to the traditional algorithms and reflect CMSRAS's capability to avoid local optimum solutions.

As per results in Table 20 and 22, for F9, F10, F11, F12, F13, F18, F20, F21, F22 and F23, compared with other advanced algorithms, the AVG and STD of CMSRAS was ranked first. For F16, F17, F19, F25, F24 and F27, the AVG and STD of CMSRAS was equal to or close to other advanced algorithms. However, for F8, the AVG of CMSRAS was inferior to I-GWO and GWOCS, but the STD of CMSRAS was superior to all advanced algorithms. In addition, for F15, the AVG of CMSRAS was better than TAPSO, MPSO, IPSO, AGPSO1, AGPSO2 and GWOCS, but that was worse than I-GWO and AGPSO3. The STD of CMSRAS was superior to other advanced algorithms except I-GWO. For F14, the AVG and STD of both CMSRAS and I-GWO were better than other advanced algorithms.

Compared with traditional algorithms, Fig. 23 shows that CMSRAS can find a superior solution at a relatively

fast convergence tendency in multimodal functions such as F8-F15, F16-F19, F21-F25 and F27. In addition, in Fig. 24, it shows that CMSRAS also can find a superior solution at a relatively fast convergence tendency in multimodal functions such as F9-F19, F21-F25 and F27.

To sum up, CMSRAS can achieve superior faster than most of other counterparts, thus well coordinating the ability of exploration and exploration, and CMSRAS can avoid falling into local optimum with fast convergence.

E. SIGNIFICANCE OF SUPERIOR ANALYSIS

There are often some shortcomings in evaluating the performance of CMSRAS algorithm based on the AVG and STD. In order to accurately evaluate the performance of the CMSRAS algorithm, a statistical test is needed to determine whether there are statistically significant differences between the CMSRAS algorithm and other competitive algorithms. In this section, the Wilcoxon rank-sum test with 5% degree is carefully performed on the results of 30 independent runs [67]. Table 23 shows the obtained *p*-value, *h*-value and *z*-value of the Wilcoxon rank-sum test with 5% significance. The term of NaN means that both algorithms are successful in determining optimal points of a specific function in all the

TABLE 19. Comparison results of multimodal functions with traditional algorithms.

Algorithms	F8		F9		F10	
	AVG	STD	AVG	STD	AVG	STD
CMSRAS	-9.3358E+03	1.975E+02	0.000	0.000	8.8818E-16	0.000
PSO	-6.7400E+03	9.967E+02	7.0344E+01	2.178E+01	8.454E+02	9.192E-02
CS	-9.2459E+03	2.786E+02	5.522E+01	8.251	4.510E-01	7.114E-01
DA	-5.8867E+03	6.169E+02	1.402E+02	3.967E+01	7.309E+00	1.166E+00
GWO	-6.2080E+03	1.0126E+03	1.351E-01	5.14E-01	8.704E-15	1.957E-15
MFO	-8.9889E+03	8.327E+02	1.49E+02	2.791E+01	1.391E+01	8.301E+00
HHO	-1.25694E+04	3.8551E-02	0.000	0.000	8.8818E-16	0.000
MVO	-8.0986E+03	7.266E+02	1.038E+02	2.843E+01	7.731E-01	7.676E-01
SMA	-1.25694E+04	1.614E-02	0.000	0.000	8.8818E-16	0.000
SCA	-4.0603E+03	2.743E+02	7.735E+00	1.857E+01	1.590E+01	7.496E+00
SOA	-1.25694E+04	1.65E-06	0.000	0.000	8.8818E-16	0.000
WOA	-1.13055E+04	1.4799E+03	0.000	0.000	4.322E-15	2.184E-15
Algorithms	F11		F12		F13	
	AVG	AVG	AVG	STD	AVG	STD
CMSRAS	0.000	0.000	7.233E-16	1.920E-15	2.553E-12	6.322E-12
PSO	1.158E-02	9.121E-03	6.488E-05	7.534E-05	5.893E-03	5.948E-03
CS	9.61E-05	2.237E-04	2.753E-01	4.379E-01	1.72E-06	6.306E-06
DA	6.665E+00	3.035E+00	2.867E+01	6.885E+01	3.801E+03	6.601E+03
GWO	0.000	0.000	3.569E-02	2.306E-02	5.261E-01	1.942E-01
MFO	3.042E+00	1.652E+00	4.631E-01	8.436E-01	1.844E-01	7.075E-01
HHO	0.000	0.000	5.846E-07	1.103E-06	7.248E-06	1.331E-05
MVO	3.073E-01	6.852E-02	1.071E+00	1.037E+00	2.203E-02	2.444E-02
SMA	0.000	0.000	2.535E-04	3.072E-04	1.896E-04	9.073E-05
SCA	1.195E-01	2.101E-01	1.0704E+00	2.347E+00	2.309E+00	1.417E-01
SOA	0.000	0.000	1.390E-01	1.352E-01	6.881E-01	5.728E-01
WOA	2.198E-03	8.416E-03	2.034E-02	5.938E-02	4.258E-02	5.327E-02

TABLE 20. Comparison results of multimodal functions with advanced algorithms.

Algorithms	F8		F9		F10	
	AVG	AVG	AVG	STD	AVG	STD
CMSRAS	-1.0518E+04	1.553E+02	0.000	0.000	8.8818E-16	0
TAPSO	-6.959E+03	8.583E+02	5.881E+01	2.233E+01	4.468E-02	2.447E-01
MPSO	-6.729E+03	7.931E+02	9.167E+01	3.245E+01	5.926E-02	3.246E-01
IPSO	-6.751E+03	5.978E+02	7.232E+01	2.189E+01	5.005E-02	2.741E-01
I-GWO	-1.0523E+04	7.996E+02	1.476E+01	8.046E+00	7.993E-15	9.329E-16
AGPSO1	-6.615E+03	8.033E+02	5.97E+01	1.938E+01	1.068E-01	3.316E-01
AGPSO2	-6.556E+03	8.529E+02	6.341E+01	1.974E+01	4.469E-02	2.447E-01
AGPSO3	-6.637E+03	8.705E+02	4.344E+01	1.236E+01	7.046E-01	6.638E-01
GWOCs	-1.0295E+04	2.018E+03	2.545E-01	1.394E+00	9.177E-15	2.154E-15
Algorithms	F11		F12		F13	
	AVG	AVG	AVG	STD	AVG	STD
CMSRAS	0.000	0.000	6.16E-16	9.744E-16	3.256E-12	9.939E-12
TAPSO	6.337E-02	6.459E-02	1.727E-02	5.501E-02	5.493E-03	4.493E-02
MPSO	1.19E-02	1.193E-02	2.351E-13	1.261E-12	3.631E-03	9.009E-03
IPSO	9.187E-03	1.484E-02	3.455E-03	1.892E-02	1.098E-03	5.908E-03
I-GWO	1.559E-03	4.36E-03	3.456E-03	1.892E-02	3.296E-03	3.352E-03
AGPSO1	6.814E-03	9.091E-03	1.708E-11	6.454E-11	2.532E-03	1.805E-02
AGPSO2	8.784E-03	9.709E-03	3.481E-12	8.615E-12	2.563E-03	5.424E-03
AGPSO3	9.602E-03	9.124E-03	1.363E-12	7.127E-12	1.098E-03	4.726E-03
GWOCs	1.274E-03	3.947E-03	3.323E-02	2.004E-02	4.738E-01	3.352E-03

runs and the statistical Wilcoxon test is not applicable. And the non-parametric Friedman’s test [68] was utilized. The average rank of the results of the algorithms on 32 benchmark functions is shown in Table 24. As it can be seen from Table 23, for F₁, CMSRAS is significantly better than other

competitive algorithms, except HHO, SMA, SOA. For F₂, CMSRAS is significantly better than all competitive algorithms. For F₃ and F₄, CMSRAS is significantly better compared to other algorithms, except SMA. For F₅, CMSRAS is significantly superior to other algorithms, except CS, TAPSO,

TABLE 21. Comparison results of fixed dimension multimodal functions with traditional algorithms.

Algorithms	F14		F15		F16	
	AVG	AVG	AVG	STD	AVG	STD
CMSRAS	9.98E-01	0.000	3.388E-04	1.670E-04	-1.0316	6.775E-16
PSO	3.359E+00	2.971E+00	7.683E-04	2.318E-04	-1.0316	6.775E-16
CS	9.98E-01	0.000	3.074E-04	1.769E-19	-1.0316	6.775E-16
DA	9.98E-01	2.1785E-10	2.206E-03	3.699E-03	-1.0316	1.278E-05
GWO	4.845E+00	4.479E+00	5.720E-03	8.984E-03	-1.0316	1.595E-09
MFO	2.314E+00	2.065E+00	1.419E-03	3.583E-03	-1.0316	6.775E-16
HHO	9.98E-01	1.092E-10	3.462E-04	1.661E-04	-1.0316	3.657E-13
MVO	9.98E-01	1.963E-12	6.584E-03	1.232E-02	-1.0316	2.38E-08
SMA	9.98E-01	7.086E-15	3.877E-04	1.539E-04	-1.0316	3.961E-12
SCA	1.461E+00	8.535E-01	7.366E-04	3.945E-04	-1.0316	2.05E-05
SOA	5.422E+00	4.844E+00	1.606E-03	1.535E-03	-1.0316	5.28E-03
WOA	1.621E+00	1.861E+00	5.769E-04	3.204E-04	-1.0316	4.503E-12
Algorithms	F17		F18		F19	
	AVG	STD	AVG	STD	AVG	STD
CMSRAS	3.979E-01	0.000	3E+00	1.533E-16	-3.862E+00	2.71E-15
PSO	3.979E-01	0.000	3E+00	6.059E-16	-3.862E+00	2.682E-15
CS	3.979E-01	0.000	3E+00	1.319E-15	-3.862E+00	2.71E-15
DA	3.979E-01	1.102E-05	3E+00	3.187E-05	-3.862E+00	2.748E-05
GWO	3.979E-01	2.044E-07	5.7E+00	1.478E+01	-3.862E+00	1.353E-03
MFO	3.979E-01	0.000	3E+00	2.041E-15	-3.862E+00	2.371E-15
HHO	3.979E-01	2.106E-08	3E+00	2.099E-09	-3.862E+00	1.295E-03
MVO	3.979E-01	3.137E-08	3E+00	1.616E-07	-3.862E+00	8.896E-08
SMA	3.979E-01	6.151E-10	3E+00	1.737E-13	-3.862E+00	3.075E-09
SCA	3.982E-01	3.698E-04	3E+00	3.952E-06	-3.855E+00	2.427E-03
SOA	3.980E-01	1.492E-04	1.524E+01	1.428E+00	-3.761E+00	1.957E-01
WOA	3.979E-01	4.469E-08	3E+00	1.831E-06	-3.861E+00	2.304E-03
Algorithms	F20		F21		F22	
	AVG	STD	AVG	STD	AVG	STD
CMSRAS	-3.321	1.108E-15	-10.1532	6.564E-15	-10.4029	7.054E-16
PSO	-3.262	6.046E-02	-8.2976	2.480E+00	-9.7953	1.887E+00
CS	-3.321	1.342E-15	-10.1532	7.226E-15	-10.4029	1.475E-15
DA	-3.254	7.809E-02	-9.6333	1.542E+00	-8.4666	2.810E+00
GWO	-3.259	8.422E-02	-9.6463	1.546E+00	-10.4028	7.220E-05
MFO	-3.250	6.596E-02	-7.2246	3.293E+00	-7.3223	3.433E+00
HHO	-3.166	8.037E-02	-5.5631	1.549E+00	-5.6148	1.609E+00
MVO	-3.270	5.998E-02	-7.9546	2.557E+00	-9.4437	2.216E+00
SMA	-3.238	5.541E-02	-10.1532	1.708E-05	-10.4029	2.549E-05
SCA	-2.826	4.48E-01	-3.0953	2.707E+00	-3.8418	2.349E+00
SOA	-2.874	4.038E-01	-4.4579	2.783E+00	-3.8012	2.023E+00
WOA	-3.253	7.923E-02	-9.5042	2.078E+00	-9.8254	1.772E+00
Algorithms	F23		F24		F25	
	AVG	STD	AVG	STD	AVG	STD
CMSRAS	-10.5363	2.856E-15	1.349E-31	6.680E-47	-1	0.000
PSO	-9.9971	1.645E+00	1.349E-31	6.680E-47	-1	0.000
CS	-10.5364	1.806E-15	1.349E-31	6.68E-47	-1	0.000
DA	-9.3445	2.469E+00	2.239E-29	1.134E-28	-1	7.511E-06
GWO	-10.5363	4.729E-05	3.558E-10	6.814E-10	-1	3.913E-08
MFO	-7.2839	3.824E+00	2.492E-23	5.966E-23	-1	0.000
HHO	-5.1282	4.028E-04	1.349E-31	6.680E-47	-1	2.331E-07
MVO	-9.6429	2.031E+00	3.216E-09	6.001E-09	-8.99E-01	3.051E-01
SMA	-10.5363	2.940E-05	1.036E-14	2.370E-14	-9.92E-01	2.345Ee-02
SCA	-4.7077	1.607E+00	8.660E-08	1.545E-07	-9.99E-01	5.532E-04
SOA	-3.8423	2.510E+00	6.485E-10	2.046E-09	-9.68E-01	3.50E-02
WOA	-10.3547	9.870E-01	6.608E-17	2.531E-16	-1	6.226E-09
Algorithms	F26		F27			
	AVG	STD	AVG	STD		
CMSRAS	0.000	0.000	0.000	0.000		
PSO	2.247E-18	4.203E-18	0.000	0.000		
CS	2.990E-17	4.898E-17	0.000	0.000		
DA	2.361E-01	8.661E-01	2.409E-03	9.614E-03		
GWO	0.000	0.000	0.000	0.000		
MFO	0.000	0.000	0.000	0.000		
HHO	0.000	0.000	0.000	0.000		
MVO	4.065E-01	1.771E-01	2.688E-05	2.555E-05		
SMA	0.000	0.000	0.000	0.000		
SCA	8.566E-70	2.641E-69	0.000	0.000		
SOA	0.000	0.000	0.000	0.000		
WOA	0.000	0.000	0.000	0.000		

TABLE 22. Comparison results of fixed-dimension multimodal functions with advanced algorithms.

Algorithms	F14		F15		F16	
	AVG	STD	AVG	STD	AVG	STD
CMSRAS	0.998	0.000	3.378E-04	1.584E-04	-1.0316	6.775E-16
TAPSO	1.163 E+00	3.767E-01	4.4657E-04	3.001E-04	-1.0316	6.712E-16
MPSO	1.916 E+00	2.160E+00	1.819E-03	4.027E-03	-1.0316	6.712E-16
IPSO	1.163 E+00	3.767E-01	8.065E-04	1.469E-03	-1.0316	6.712E-16
I-GWO	0.998	0.000	3.074E-04	1.255E-10	-1.0316	6.775E-16
AGPSO1	2.116 E+00	1.929E+00	1.221E-04	3.639E-03	-1.0316	6.584E-16
AGPSO2	1.130 E+00	3.436E-01	1.068E-03	3.652E-03	-1.0316	6.712E-16
AGPSO3	1.460 E+00	9.610E-01	3.233E-04	8.675E-05	-1.0316	6.712E-16
GWOCs	4.326 E+00	4.196 E+00	3.808E-04	2.835E-04	-1.0316	1.704E-09
Algorithms	F17		F18		F19	
	AVG	STD	AVG	STD	AVG	STD
CMSRAS	3.978E-01	0.000	3E+00	8.49E-16	-3.862E+00	2.710E-15
TAPSO	3.978E-01	0.000	3E+00	1.421E-15	-3.862E+00	2.710E-15
MPSO	3.978E-01	0.000	3E+00	1.314E-15	-3.862E+00	2.682E-15
IPSO	3.978E-01	0.000	3E+00	1.272E-15	-3.862E+00	2.696E-15
I-GWO	3.978E-01	0.000	3E+00	1.862E-15	-3.862E+00	2.710E-15
AGPSO1	3.978E-01	0.000	3E+00	2.313E-15	-3.862E+00	2.696E-15
AGPSO2	3.978E-01	0.000	3E+00	1.269E-15	-3.862E+00	2.710E-15
AGPSO3	3.978E-01	0.000	3E+00	1.538E-15	-3.862E+00	2.710E-15
GWOCs	3.978E-01	8.906E-08	3E+00	1.929E-06	-3.862E+00	2.038E-06
Algorithms	F20		F21		F22	
	AVG	STD	AVG	STD	AVG	STD
CMSRAS	-3.321E+00	1.806E-15	-10.1532	5.770E-15	-10.4029	8.079E-16
TAPSO	-3.260E+00	6.395E-02	-5.6353	2.964E+00	-9.7180	2.120E+00
MPSO	-3.245E+00	6.581E-02	-8.4690	2.422E+00	-9.5239	1.999E+00
IPSO	-3.272E+00	6.304E-02	-7.8026	2.795E+00	-9.4757	2.121E+00
I-GWO	-3.294E+00	5.114E-02	-9.8367	1.207E+00	-10.4029	3.459E-15
AGPSO1	-3.253E+00	6.059E-02	-9.6479	1.541E+00	-10.4029	1.094E-15
AGPSO2	-3.270E+00	6.015E-02	-8.3869	2.575E+00	-9.2666	2.345E+00
AGPSO3	-3.278E+00	5.827E-02	-8.5553	2.518E+00	-9.8755	1.609E+00
GWOCs	-3.321E+00	7.789E-07	-8.9726	2.176E+00	-9.1653	2.281E+00
Algorithms	F23		F24		F25	
	AVG	STD	AVG	STD	AVG	STD
CMSRAS	-10.5363	2.086E-15	1.349E-31	6.68E-47	-1	0.000
TAPSO	-9.7448	2.085E+00	1.349E-31	6.680E-47	-1	0.000
MPSO	-10.3561	9.873E-01	1.349E-31	6.680E-47	-1	0.000
IPSO	-9.8466	2.133E+00	1.349E-31	6.680E-47	-1	0.000
I-GWO	-10.5364	1.979E-15	1.349E-31	6.680E-47	-1	0.000
AGPSO1	-10.2809	1.399E+00	1.349E-31	6.680E-47	-1	0.000
AGPSO2	-10.5364	1.894E-15	1.349E-31	6.680E-47	-1	0.000
AGPSO3	-9.7448	2.085E+00	1.349E-31	6.680E-47	-1	0.000
GWOCs	-10.1773	1.366E+00	3.675E-10	6.470E-10	-1	5.148E-08
Algorithms	F26		F27			
	AVG	STD	AVG	STD		
CMSRAS	0.000	0.000	0.000	0.000		
TAPSO	7.665E-35	7.651E-35	0.000	0.000		
MPSO	4.964E-32	1.523E-31	0.000	0.000		
IPSO	4.813E-32	5.717E-32	0.000	0.000		
I-GWO	0.000	0.000	0.000	0.000		
AGPSO1	1.839E-31	2.588E-31	0.000	0.000		
AGPSO2	1.248E-31	1.440E-31	0.000	0.000		
AGPSO3	7.126E-48	1.237E-47	0.000	0.000		
GWOCs	0.000	0.000	0.000	0.000		

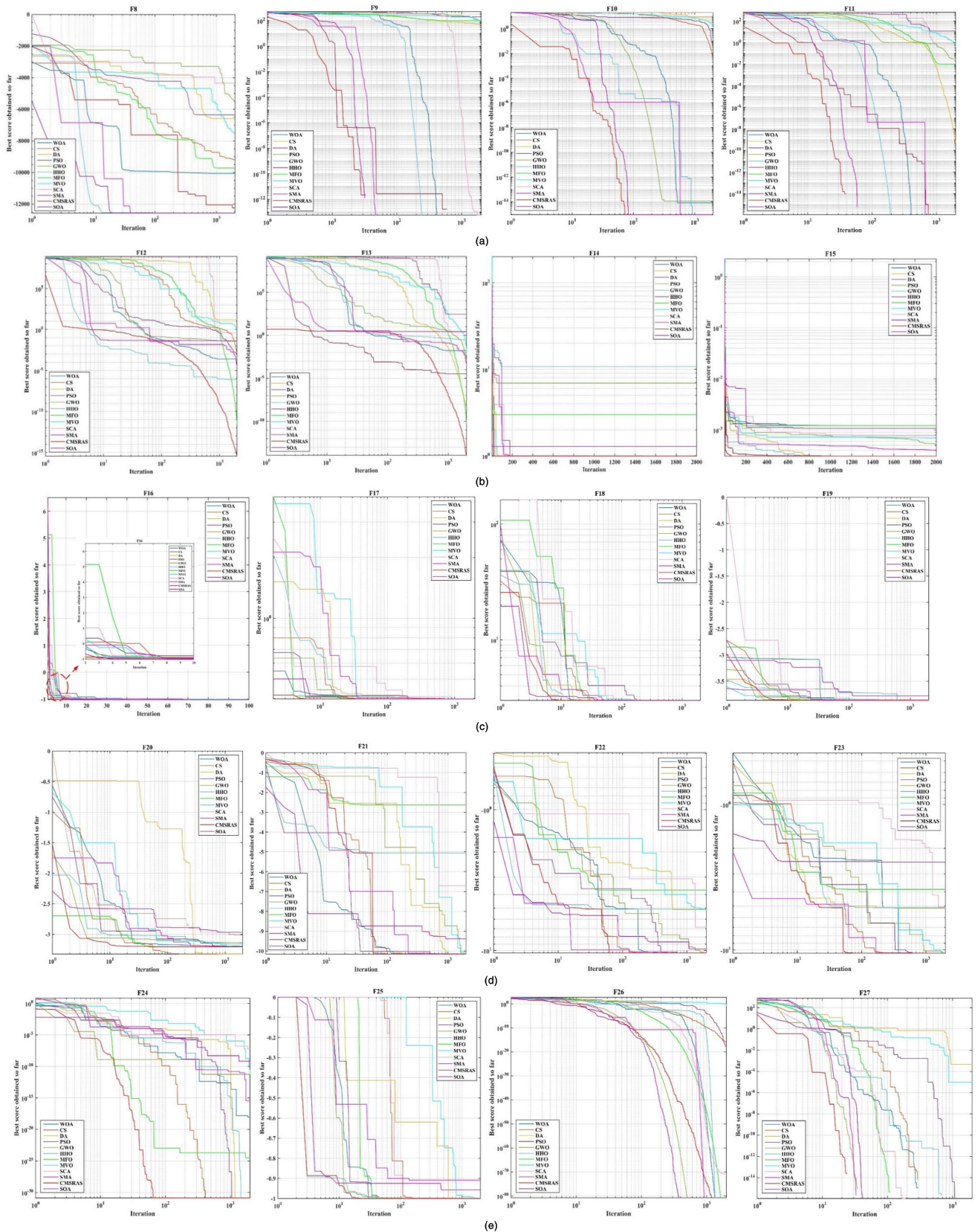


FIGURE 23. The convergence curves of multimodal benchmark functions by traditional algorithms.

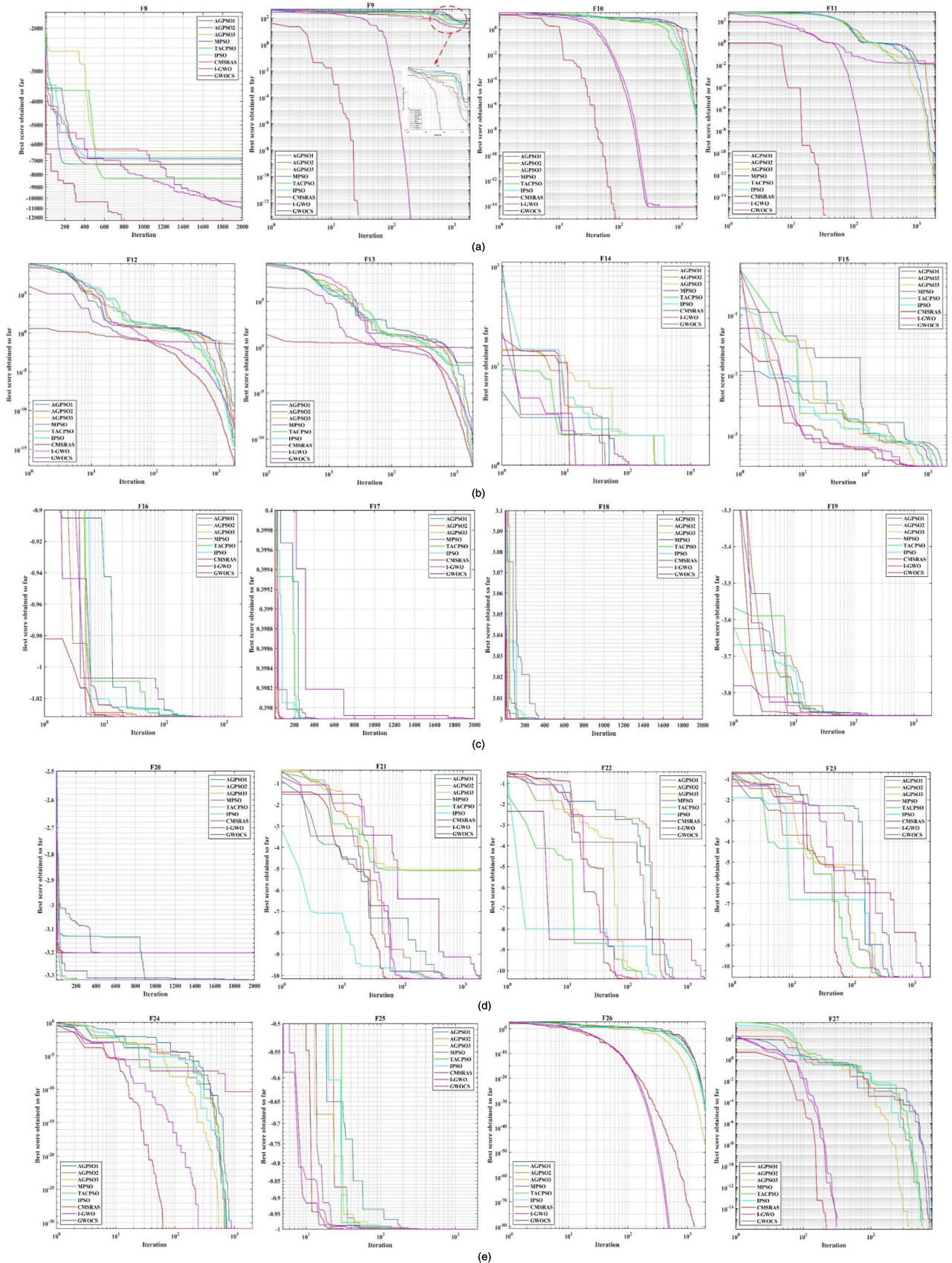


FIGURE 24. The convergence curves of multimodal benchmark functions by advanced algorithms.

MPSO, IPSO, AGPSO1, AGPSO2 and AGPSO3. For F_6 , CMSRAS is significantly superior to most other algorithms, but inferior to TAPSO, MPSO, IPSO and AGPSO3. For F_7 , CMSRAS is significantly better than most other algorithms, but less than SMA and SOA. For F_8 , CMSRAS is significantly less than MFO. For F_9 - F_{11} , CMSRAS is not significantly better compared to HHO, SMA, SOA and GWOCS. For F_{12} , CMSRAS is not significantly better compared with MPSO, IPSO and AGPSO3. For F_{13} , CMSRAS is significantly better compared with other competitors, expect MPSO, IPSO and TAPSO. For F_{14} and F_{16} , CMSRAS is not significantly superior to CS, HHO, SMA and I-GWO. For F_{15} , CMSRAS is significantly better compared with other competitors, expect GWOCS. For F_{17} and F_{19} , CMSRAS is significantly superior to other competitors, except CS, PSO, MFO, I-GWO and GWOCS. For F_{18} and F_{24} , CMSRAS is not significantly better compared to PSO, CS, TAPSO, MPSO, IPSO, I-GWO, AGPSO1, AGPSO2 and AGPSO3. For F_{20} , CMSRAS is not significantly superior to DA, MFO and MVO. For F_{21} , CMSRAS is not significantly better compared to CS and I-GWO. CMSRAS is not significantly superior to PSO, CS and I-GWO on F_{22} and F_{23} . CMSRAS is not significantly superior to PSO, CS MFO, TAPSO, MPSO, IPSO, AGPSO1, AGPSO2, AGPSO3, I-GWO and GWOCS on F_{25} and F_{27} . CMSRAS is significantly better compared to DA, GWO, MFO, HHO, SMA, MVO, SCA, SOA, WOA and GWOCS on F_{28} and F_{29} . CMSRAS is not significantly better compared to HHO, MVO, I-GWO and GWOCS on F_{30} and F_{31} . For F_{26} , CMSRAS is significantly better compared to other competitors, except GWO, MFO, HHO, SMA, WOA, SOA, I-GWO and GWOCS. For F_{32} , CMSRAS is significantly better compared to PSO, DA, GWO, MFO, HHO, MVO, SMA, SCA, WOA, SOA, I-GWO and GWOCS. Therefore, the statistical results of p -value, h -value and z -value detected that the solutions of CMSRAS are significantly better than those realized by other competitive algorithms in most cases. As it can be seen from in Table 24, the results show that CMSARS is ranked first compared to other competitive algorithms for 32 benchmark functions. Accordingly, CMSRAS has the best performance among all these competitive algorithms from a statistical point of view.

F. AVERAGE TIME-CONSUMING ANALYSIS

In the section, for 32 benchmark functions, CMSRAS algorithm was compared with other 13 competitive algorithms in the average time-consuming experiment mentioned above. Under the same lab environment, the average time-consuming experiment was obtained by running independently 30 times for each benchmark function, and the results of the average time-consuming were shown in Table 25. As can be observed from the data in Table 25, the computation of CMSRAS took a relatively longer time than SRA. This may be due to the enlargement of the population and the addition of some mutation strategy. However, it can be

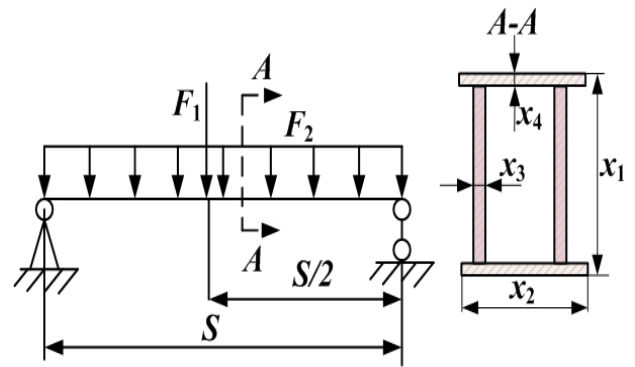


FIGURE 25. Simplified model of the crane box girder.

seen from the experimental results that CMSRAS is significantly better than SRA and some competitive algorithms in most cases. Therefore, the improved strategies introduced into SRA are worth it. In addition, CMSRAS can still outperform some algorithms while taking less time, such as DA, SMA, and IGWO. Generally, even if it is relatively time-consuming, CMSRAS still has better advantages over other algorithms.

VI. ENGINEERING STRUCTURE OPTIMIZATION CASES

A. CASE I

Cranes are widely used in industrial and mining enterprises, ports, construction sites, aerospace, energy construction and other aspects. As the main load-bearing structure (subject to lifting loads, self-weight, external load, etc.), a box-shaped welded structure is generally adopted, and the simplified structure model and the definition and value of design parameters are described, as shown in Fig. 25 and Table 26, respectively.

a) Design variable selection and objective function determination

x_1, x_2, x_3 and x_4 are selected as optimization design variables and written in vector form as follows: $X = [x_1 x_2 x_3 x_4]^T$. In general, the minimum cross section area of the beam is selected as the optimization objective function, as shown in (16).

$$\min f(X) = x_1 x_3 + x_2 x_4 \tag{16}$$

b) Determining constraints

The design of the crane box girder should satisfy the requirements of strength, stability and stiffness (3S), respectively. Therefore, the constraint conditions can be obtained from the aspects of 3S and geometric dimension.

(1) Strength constraint

$$g_1(X) = \frac{3S}{4} \left[\frac{F_1 + 7.8 \times 10^{-5}(x_1 x_3 + x_2 x_4)}{3x_1 x_2 x_4 + x_1^2 x_3} + \frac{F_2}{3x_1 x_2 x_3 + x_2^2 x_4} \right] - 140 \leq 0 \tag{17}$$

TABLE 23. Results of wilcoxon’s rank-sum test on 32 benchmark functions.

CMSRAS Vs.	Wilcoxon’s rank sum test	F1	F2	F3	F4	F5	F6	F7	F8	
Traditional algorithms	PSO	<i>p</i> -value	1.21E-12	3.02E-11	1.21E-12	3.00E-11	1.21E-12	3.01E-11	3.01E-11	2.78E-07
		<i>h</i> -value	1	1	1	1	1	1	1	1
		<i>z</i> -value	-7.104	-6.645	-7.104	-6.646	-7.104	-6.645	-6.645	-5.131
	CS	<i>p</i> -value	1.21E-12	3.02E-11	1.21E-12	3.00E-11	NaN	3.02E-11	3.02E-11	7.81E-01
		<i>h</i> -value	1	1	1	1	0	1	1	0
		<i>z</i> -value	-7.104	-6.645	-7.104	-6.646	NaN	-6.645	-6.645	-0.273
	DA	<i>p</i> -value	1.21E-12	3.02E-11	1.21E-12	2.99E-11	1.21E-12	3.02E-11	3.02E-11	7.12E-09
		<i>h</i> -value	1	1	1	1	1	1	1	1
		<i>z</i> -value	-7.104	-6.645	-7.104	-6.646	-7.104	-6.645	-6.645	-5.788
	GWO	<i>p</i> -value	1.21E-12	3.01E-11	1.21E-12	3.00E-11	1.21E-12	3.02E-11	1.36E-07	2.60E-08
		<i>h</i> -value	1	1	1	1	1	1	1	1
		<i>z</i> -value	-7.104E	-6.646	-7.104	-6.646	-7.104	-6.645	-5.270	-5.566
	MFO	<i>p</i> -value	1.21E-12	3.02E-11	1.21E-12	3.00E-11	1.21E-12	2.35E-05	3.02E-11	2.28E-01
		<i>h</i> -value	1	1	1	1	1	1	1	0
		<i>z</i> -value	-7.104	-6.645	-7.104	-6.646	-7.104	-4.228	-6.645	-1.205
	HHO	<i>p</i> -value	NaN	9.76E-10	8.86E-07	3.00E-11	1.21E-12	3.02E-11	9.88E-03	5.57E-10
		<i>h</i> -value	0	1	1	1	1	1	1	1
		<i>z</i> -value	NaN	-6.113	-4.915	-6.646	-7.104	-6.645	2.579	6.202
	MVO	<i>p</i> -value	1.21E-12	3.02E-11	1.21E-12	3.00E-11	1.21E-12	3.02E-11	3.02E-11	3.56E-04
		<i>h</i> -value	1	1	1	1	1	1	1	1
		<i>z</i> -value	-7.104	-6.645	-7.104	-6.646	-7.104	-6.645	-6.645	-3.570
SMA	<i>p</i> -value	NaN	2.52E-11	NaN	5.06E-01	1.21E-12	3.02E-11	3.63E-01	5.56E-10	
	<i>h</i> -value	0	1	0	0	1	1	0	1	
	<i>z</i> -value	NaN	6.672	NaN	0.664	-7.104	-6.645	0.909	6.202	
SCA	<i>p</i> -value	1.21E-12	3.01E-11	1.21E-12	3.00E-11	1.21E-12	3.02E-11	3.02E-11	1.61E-10	
	<i>h</i> -value	1	1	1	1	1	1	1	1	
	<i>z</i> -value	-7.104	-6.645	-7.104	-6.646	-7.104	-6.645	-6.645	-6.394	
SOA	<i>p</i> -value	NaN	1.21E-12	1.21E-12	3.00E-11	1.21E-12	3.02E-11	2.64E-01	9.69E-11	
	<i>h</i> -value	0	1	1	1	1	1	0	1	
	<i>z</i> -value	NaN	7.104	-7.104	-6.646	-7.104	-6.645	1.116	6.471	
WOA	<i>p</i> -value	1.65E-11	2.01E-04	1.21E-12	3.00E-11	1.21E-12	3.02E-11	1.74E-05	1.40E-04	
	<i>h</i> -value	1	1	1	1	1	1	1	1	
	<i>z</i> -value	-6.733	3.718	-7.104	-6.646	-7.104	-6.645	-4.294	3.807	
Advanced algorithms	TAPSO	<i>p</i> -value	1.21E-12	3.02E-11	1.21E-12	3.00E-11	NaN	3.04E-01	3.02E-11	4.79E-07
		<i>h</i> -value	1	1	1	1	0	0	1	1
		<i>z</i> -value	-7.104	-6.645	-7.104	-6.646	NaN	1.027	-6.645	-5.034
	MPSO	<i>p</i> -value	1.21E-12	3.02E-11	1.21E-12	3.00E-11	NaN	1.95E-01	3.02E-11	2.37E-07
		<i>h</i> -value	1	1	1	1	0	0	1	1
		<i>z</i> -value	-7.104	-6.645	-7.104	-6.646	NaN	-1.293	-6.645	-5.167
	IPSO	<i>p</i> -value	1.21E-12	3.02E-11	1.21E-12	3.00E-11	NaN	1.95E-01	3.02E-11	1.25E-07
		<i>h</i> -value	1	1	1	1	0	0	1	1
		<i>z</i> -value	-7.104	-6.645	-7.104	-6.646	NaN	-1.293	-6.645	-5.285
	I-GWO	<i>p</i> -value	1.21E-12	3.02E-11	1.21E-12	3.00E-11	1.21E-12	3.02E-11	3.02E-11	3.33E-03
		<i>h</i> -value	1	1	1	1	1	1	1	1
		<i>z</i> -value	-7.104	-6.645	-7.104	-6.646	-7.104	-6.645	-6.645	2.935
	AGPSO1	<i>p</i> -value	1.21E-12	3.02E-11	1.21E-12	3.00E-11	NaN	2.61E-10	3.02E-11	1.06E-07
		<i>h</i> -value	1	1	1	1	0	1	1	1
		<i>z</i> -value	-7.104	-6.645	-7.104	-6.646	NaN	-6.320	-6.645	-5.315
	AGPSO2	<i>p</i> -value	1.21E-12	3.02E-11	1.21E-12	3.00E-11	NaN	4.40E-10	3.02E-11	1.06E-07
		<i>h</i> -value	1	1	1	1	0	1	1	1
		<i>z</i> -value	-7.104	-6.645	-7.104	-6.646	NaN	-6.239	-6.645	-5.315
	AGPSO3	<i>p</i> -value	1.21E-12	3.02E-11	1.21E-12	3.00E-11	NaN	1.08E-01	3.02E-11	9.82E-08
		<i>h</i> -value	1	1	1	1	0	0	1	1
		<i>z</i> -value	-7.104	-6.645	-7.104	-6.646	NaN	1.604	-6.645	-5.329
GWCS	<i>p</i> -value	1.21E-12	3.02E-11	1.21E-12	3.00E-11	1.21E-12	2.98E-11	8.88E-10	3.14E-02	
	<i>h</i> -value	1	1	1	1	1	1	1	1	
	<i>z</i> -value	-7.104	-6.6457	-7.104	-6.646	-7.104	-6.647	-6.128	2.151	
CMSRAS Vs.	Wilcoxon’s rank sum test	F9	F10	F11	F12	F13	F14	F15	F16	
Traditional algorithms	PSO	<i>p</i> -value	1.21E-12	1.21E-12	1.21E-12	3.02E-11	3.02E-11	1.20E-07	2.44E-10	NaN
		<i>h</i> -value	1	1	1	1	1	1	1	0
		<i>z</i> -value	-7.104	-7.104	-7.104	-6.645	-6.645	-5.292	-6.330	NaN
	CS	<i>p</i> -value	1.21E-12	1.21E-12	1.21E-12	3.02E-11	3.02E-11	NaN	2.78E-03	NaN
		<i>h</i> -value	1	1	1	1	1	0	1	0
		<i>z</i> -value	-7.104	-7.104	-7.104	-6.645	-6.645	NaN	2.990	NaN
DA	<i>p</i> -value	1.21E-12	1.21E-12	1.21E-12	3.02E-11	3.02E-11	3.33E-01	3.26E-11	4.57E-12	

TABLE 23. (Continued.) Results of wilcoxon’s rank-sum test on 32 benchmark functions.

GWO	<i>h</i> -value	1	1	1	1	1	0	1	1
	<i>z</i> -value	-7.104	-7.104	-7.104	-6.645	-6.645	-0.966	-6.633	-6.918
	<i>p</i> -value	4.19E-02	8.71E-14	NaN	3.02E-11	3.02E-11	7.94E-07	1.44E-05	4.28E-08
MFO	<i>h</i> -value	1	1	0	1	1	1	1	1
	<i>z</i> -value	-2.034	-7.459	NaN	-6.645	-6.645	-4.936	-4.337	-5.478
	<i>p</i> -value	1.21E-12	1.21E-12	1.20E-12	3.32E-11	6.08E-08	1.23E-05	1.47E-10	NaN
HHO	<i>h</i> -value	1	1	1	1	1	1	1	0
	<i>z</i> -value	-7.104	-7.104	-7.104	-6.631	-5.416	-4.371	-6.407	NaN
	<i>p</i> -value	NaN	NaN	NaN	3.02E-11	3.02E-11	NaN	1.62E-08	NaN
MVO	<i>h</i> -value	0	0	0	1	1	0	1	0
	<i>z</i> -value	NaN	NaN	NaN	-6.645	-6.645	NaN	-5.647	NaN
	<i>p</i> -value	1.21E-12	1.21E-12	1.21E-12	3.02E-11	3.02E-11	NaN	4.93E-11	4.54E-12
SMA	<i>h</i> -value	1	1	1	1	1	0	1	1
	<i>z</i> -value	-7.104	-7.104	-7.104	-6.645	-6.645	NaN	-6.572	-6.919
	<i>p</i> -value	NaN	NaN	NaN	3.02E-11	3.02E-11	NaN	8.71E-09	NaN
SCA	<i>h</i> -value	0	0	0	1	1	0	1	0
	<i>z</i> -value	NaN	NaN	NaN	-6.645	-6.645	NaN	-5.754	NaN
	<i>p</i> -value	1.21E-12	1.21E-12	1.21E-12	3.02E-11	3.02E-11	1.15E-12	1.22E-10	1.21E-12
SOA	<i>h</i> -value	1	1	1	1	1	1	1	1
	<i>z</i> -value	-7.104	-7.104	-7.104	-6.645	-6.645	-7.110	-6.436	-7.104
	<i>p</i> -value	NaN	NaN	NaN	3.02E-11	3.02E-11	1.61E-11	5.46E-11	1.21E-12
WOA	<i>h</i> -value	0	0	0	1	1	1	1	1
	<i>z</i> -value	NaN	NaN	NaN	-6.645	-6.645	-6.737	-6.557	-7.104
	<i>p</i> -value	NaN	8.81E-10	1.60E-01	3.02E-11	3.02E-11	5.56E-03	2.21E-10	NaN
TAPSO	<i>h</i> -value	0	1	0	1	1	1	1	0
	<i>z</i> -value	NaN	-6.129	-1.402	-6.645	-6.645	-2.772	-6.345	NaN
	<i>p</i> -value	1.21E-12	1.21E-12	1.21E-12	8.77E-02	4.03E+01	3.08E-09	2.21E-04	5.35E-09
MPSO	<i>h</i> -value	1	1	1	1	0	1	1	1
	<i>z</i> -value	-7.1042	-7.104	-7.104	-1.707	-0.836	-5.927	-3.693	5.835
	<i>p</i> -value	1.21E-12	1.21E-12	1.19E-12	3.87E+01	3.78E+01	5.30E-13	9.05E-08	1.16E-13
IPSO	<i>h</i> -value	1	1	1	0	0	1	1	1
	<i>z</i> -value	-7.104	-7.104	-7.105	-0.864	-0.8807	-7.217	-5.344	7.420
	<i>p</i> -value	1.21E-12	1.21E-12	1.20E-12	1.15E+01	1.10E+01	2.66E-07	2.10E-04	5.18E-07
I-GWO	<i>h</i> -value	1	1	1	0	0	1	1	1
	<i>z</i> -value	-7.104	-7.104	1.20E-12	0.115	1.596	-5.145	-3.706	5.019
	<i>p</i> -value	1.20E-12	4.16E-14	4.19E-02	3.01E-11	1.09E-10	NaN	2.15E-02	NaN
AGPSO1	<i>h</i> -value	1	1	1	1	1	0	1	0
	<i>z</i> -value	-7.104	-7.555	-2.034	-6.645	-6.453	NaN	2.297	NaN
	<i>p</i> -value	1.21E-12	1.21E-12	1.19E-12	8.98E-11	2.00E-08	2.52E-12	7.96E-07	1.16E-13
AGPSO2	<i>h</i> -value	1	1	1	1	1	1	1	1
	<i>z</i> -value	-7.104	-7.104	-7.106	-6.483	-5.611	-7.001	-4.936	7.420
	<i>p</i> -value	1.21E-12	1.21E-12	1.19E-12	1.47E-07	4.28E-04	8.64E-14	4.97E-07	1.68E-14
AGPSO3	<i>h</i> -value	1	1	1	1	1	1	1	1
	<i>z</i> -value	-7.104	-7.104	-7.106	-5.255	-3.521	-7.460	-5.027	7.672
	<i>p</i> -value	1.21E-12	1.20E-12	1.20E-12	1.76E-01	4.28E-04	7.04E-12	6.18E-06	7.15E-13
GWOCs	<i>h</i> -value	1	1	1	0	1	1	1	1
	<i>z</i> -value	-7.104	-7.104	-7.104	0.176	-3.521	-6.856	-4.520	7.176
	<i>p</i> -value	3.33E-01	2.53E-13	8.15E-02	3.00E-11	4.28E-04	7.87E-07	2.43E-01	NaN
CMSRAS Vs.	<i>h</i> -value	0	1	0	1	1	1	0	0
	<i>z</i> -value	-0.966	-7.317	-1.741	-6.646	-3.521	-4.938	1.166	NaN
	<i>p</i> -value	1.21E-12	1.21E-12	1.19E-12	1.47E-07	4.28E-04	8.64E-14	4.97E-07	1.68E-14
Traditional algorithms	<i>h</i> -value	1	1	1	1	1	1	1	1
	<i>z</i> -value	-7.104	-7.104	-7.106	-5.255	-3.521	-7.460	-5.027	7.672
	<i>p</i> -value	1.21E-12	1.20E-12	1.20E-12	1.76E-01	4.28E-04	7.04E-12	6.18E-06	7.15E-13
PSO	<i>h</i> -value	1	1	1	0	1	1	1	1
	<i>z</i> -value	-7.104	-7.104	-7.104	0.176	-3.521	-6.856	-4.520	7.176
	<i>p</i> -value	3.33E-01	2.53E-13	8.15E-02	3.00E-11	4.28E-04	7.87E-07	2.43E-01	NaN
CS	<i>h</i> -value	0	0	0	1	0	0	0	0
	<i>z</i> -value	NaN	NaN	NaN	7.420	NaN	NaN	NaN	NaN
	<i>p</i> -value	1.21E-12	1.21E-12	1.21E-12	8.32E-01	1.21E-12	1.21E-12	1.21E-12	1.60E-01
DA	<i>h</i> -value	1	1	1	0	1	1	1	0
	<i>z</i> -value	-7.104	-7.104	-7.104	0.211	-7.104	-7.104	-7.104	-1.402
	<i>p</i> -value	1.20E-12	1.21E-12	1.21E-12	2.61E-01	1.21E-12	1.21E-12	1.21E-12	1.21E-12
GWO	<i>h</i> -value	1	1	1	0	1	1	1	1
	<i>z</i> -value	-7.104	-7.104	-7.104	1.122	-7.104	-7.104	-7.104	-7.104
	<i>p</i> -value	NaN	NaN	NaN	1.06E-02	2.87E-05	2.88E-05	6.55E-05	1.21E-12
MFO	<i>h</i> -value	1	1	1	1	1	1	1	1
	<i>z</i> -value	-7.104	-7.104	-7.104	-6.645	-6.645	-4.936	-4.337	-5.478
	<i>p</i> -value	4.19E-02	8.71E-14	NaN	3.02E-11	3.02E-11	7.94E-07	1.44E-05	4.28E-08

TABLE 23. (Continued.) Results of wilcoxon’s rank-sum test on 32 benchmark functions.

	<i>h</i> -value	0	0	0	1	1	1	1	1	
	<i>z</i> -value	NaN	NaN	NaN	2.554	-4.182	-4.182	-3.992	-7.104	
	<i>p</i> -value	6.605E-04	1.20E-12	1.21E-12	1.09E-04	1.21E-12	1.21E-12	1.21E-12	NaN	
HHO	<i>h</i> -value	1	1	1	1	1	1	1	0	
	<i>z</i> -value	-3.4057	-7.104	-7.104	-3.869	-7.104	-7.104	-7.104	NaN	
	<i>p</i> -value	4.55E-12	1.21E-12	1.21E-12	5.04E-01	1.21E-12	1.21E-12	1.21E-12	1.21E-12	
MVO	<i>h</i> -value	1	1	1	0	1	1	1	1	
	<i>z</i> -value	-6.918	-7.104	-7.104	0.667	-7.104	-7.104	-7.104	-7.104	
	<i>p</i> -value	2.74E-03	1.17E-12	1.21E-12	3.08E-03	1.21E-12	1.21E-12	1.21E-12	1.21E-12	
SMA	<i>h</i> -value	1	1	1	1	1	1	1	1	
	<i>z</i> -value	-2.994	-7.108	-7.104	-2.958	-7.104	-7.104	-7.104	-7.104	
	<i>p</i> -value	1.21E-12	1.21E-12	1.21E-12	1.72E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	
SCA	<i>h</i> -value	1	1	1	1	1	1	1	1	
	<i>z</i> -value	-7.104	-7.104	-7.104	-7.055	-7.104	-7.104	-7.104	-7.104	
	<i>p</i> -value	1.21E-12	1.21E-12	1.21E-12	1.72E-12	1.21E-12	1.21E-12	1.21E-12	1.20E-12	
SOA	<i>h</i> -value	1	1	1	1	1	1	1	1	
	<i>z</i> -value	-7.104	-7.104	-7.104	-7.055	-7.104	-7.104	-7.104	-7.104	
	<i>p</i> -value	1.30E-07	1.21E-12	1.21E-12	8.32E-01	1.21E-12	1.21E-12	1.21E-12	1.21E-12	
WOA	<i>h</i> -value	1	1	1	0	1	1	1	1	
	<i>z</i> -value	-5.278	-7.104	-7.104	0.2118	-7.104	-7.104	-7.104	-7.104	
	<i>p</i> -value	5.35E-09	NaN	5.35E-09	2.92E-08	5.10E-11	8.14E-02	1.06E-04	NaN	
TAPSO	<i>h</i> -value	1	0	1	1	1	0	1	0	
	<i>z</i> -value	5.835	NaN	5.835	5.545	-6.567	-1.742	3.874	NaN	
	<i>p</i> -value	1.16E-13	NaN	1.16E-13	3.11E-02	1.56E-12	2.14E-02	7.03E-12	NaN	
MPSO	<i>h</i> -value	1	0	1	1	1	1	1	0	
	<i>z</i> -value	7.420	NaN	7.420	2.155	-7.068	-2.300	6.856	NaN	
	<i>p</i> -value	5.18E-07	NaN	5.18E-07	3.43E-09	4.93E-10	2.15E-02	1.74E-04	NaN	
IPSO	<i>h</i> -value	1	0	1	1	1	1	1	0	
	<i>z</i> -value	5.019	NaN	5.019	5.909	-6.221	-2.298	3.753	NaN	
	<i>p</i> -value	NaN	NaN	NaN	1.47E-09	1.61E-01	3.33E-01	NaN	NaN	
I-GWO	<i>h</i> -value	0	0	0	1	0	0	0	0	
	<i>z</i> -value	NaN	NaN	NaN	6.047	-1.402	-0.966	NaN	NaN	
	<i>p</i> -value	1.16E-13	NaN	1.16E-13	5.02E-06	3.59E-13	NaN	7.03E-12	NaN	
AGPSO1	<i>h</i> -value	1	0	1	1	1	0	1	0	
	<i>z</i> -value	7.420	NaN	7.420	4.563	-7.269	NaN	6.856	NaN	
	<i>p</i> -value	1.68E-14	NaN	1.68E-14	1.48E-11	3.61E-13	1.10E-02	1.68E-14	NaN	
AGPSO2	<i>h</i> -value	1	0	1	1	1	1	1	0	
	<i>z</i> -value	7.6727	NaN	7.672	6.749	-7.269	-2.542	7.672	NaN	
	<i>p</i> -value	7.15E-13	NaN	7.15E-13	6.12E-12	6.45E-12	8.14E-02	5.74E-07	NaN	
AGPSO3	<i>h</i> -value	1	0	1	1	1	0	1	0	
	<i>z</i> -value	7.176	NaN	7.176	6.876	-6.869	-1.742	4.999	NaN	
	<i>p</i> -value	NaN	NaN	NaN	1.68E-14	5.51E-03	1.21E-12	1.61E-01	1.21E-12	
GWOCs	<i>h</i> -value	0	0	0	1	1	1	0	1	
	<i>z</i> -value	NaN	NaN	NaN	7.672	-2.775	-7.104	-1.402	-7.104	
	<i>p</i> -value	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Advanced algorithms	CMSRAS Vs. Wilcoxon’s rank sum test	F25	F26	F27	F28	F29	F30	F31	F32	
	PSO	<i>p</i> -value	NaN	1.21E-12	NaN	NaN	NaN	1.21E-12	1.68E-14	1.21E-12
		<i>h</i> -value	0	1	0	0	0	1	1	1
		<i>z</i> -value	NaN	-7.104	NaN	NaN	NaN	-7.104	-7.672	-7.104
	CS	<i>p</i> -value	NaN	1.21E-12	NaN	NaN	NaN	1.21E-12	1.68E-14	NaN
		<i>h</i> -value	0	1	0	0	0	1	1	0
		<i>z</i> -value	NaN	-7.104	NaN	NaN	NaN	-7.104	-7.672	NaN
	DA	<i>p</i> -value	6.242E-10	1.10E-02	1.10E-02	1.21E-12	1.21E-12	1.21E-12	1.68E-14	1.13E-12
		<i>h</i> -value	1	1	1	1	1	1	1	1
		<i>z</i> -value	-6.184	-2.541	-2.541	-7.104	-7.104	-7.104	-7.672	-7.113
	GWO	<i>p</i> -value	1.20E-12	NaN	NaN	1.21E-12	1.21E-12	NaN	1.68E-14	1.21E-12
		<i>h</i> -value	1	0	0	1	1	0	1	1
		<i>z</i> -value	-7.1043	NaN	NaN	-7.1041	-7.1041	NaN	-7.6726	-7.1041
	MFO	<i>p</i> -value	NaN	NaN	NaN	1.21E-12	1.21E-12	1.21E-12	1.68E-14	1.21E-12
		<i>h</i> -value	0	0	0	1	1	1	1	1
		<i>z</i> -value	NaN	NaN	NaN	-7.104	-7.1041	-7.104	-7.672	-7.104
	HHO	<i>p</i> -value	1.64E-11	NaN	NaN	1.65E-11	1.21E-12	NaN	NaN	1.21E-12
		<i>h</i> -value	1	0	0	1	1	0	0	1
		<i>z</i> -value	-6.734	NaN	NaN	-6.733	-7.104	NaN	NaN	-7.104
	MVO	<i>p</i> -value	1.20E-12	1.21E-12	1.21E-12	1.20E-12	1.21E-12	1.21E-12	1.68E-14	1.21E-12
		<i>h</i> -value	1	1	1	1	1	1	1	1
		<i>z</i> -value	-7.1044	-7.104	-7.1041	-7.1044	-7.1041	-7.1041	-7.672	-7.1040
	SMA	<i>p</i> -value	6.55E-05	NaN	NaN	1.21E-12	1.21E-12	NaN	NaN	1.21E-12

TABLE 23. (Continued.) Results of wilcoxon’s rank-sum test on 32 benchmark functions.

	<i>h</i> -value	1	0	0	1	1	0	0	1
	<i>z</i> -value	-3.992	NaN	NaN	-7.104	-7.104	NaN	NaN	-7.104
	<i>p</i> -value	1.21E-12	1.21E-12	NaN	1.21E-12	1.21E-12	1.21E-12	1.68E-14	1.21E12
SCA	<i>h</i> -value	1	1	0	1	1	1	1	1
	<i>z</i> -value	-7.104	-7.104	NaN	-7.104	-7.104	-7.104	-7.672	-7.104
	<i>p</i> -value	1.65E-11	NaN	NaN	1.21E-12	1.21E-12	NaN	4.57E-12	1.21E-12
SOA	<i>h</i> -value	1	0	0	1	1	0	1	1
	<i>z</i> -value	-6.733	NaN	NaN	-7.104	-7.104	NaN	-6.918	-7.104
	<i>p</i> -value	1.25E-07	NaN	NaN	1.21E-12	1.21E-12	NaN	3.33E-01	1.21E-12
WOA	<i>h</i> -value	1	0	0	1	1	0	0	1
	<i>z</i> -value	-5.284	NaN	NaN	-7.104	-7.104	NaN	-0.9666	-7.104
	<i>p</i> -value	NaN	1.21E-12	NaN	NaN	NaN	1.21E-12	5.35E-09	NaN
TAPSO	<i>h</i> -value	0	1	0	0	0	1	1	0
	<i>z</i> -value	NaN	-7.104	NaN	NaN	NaN	-7.104	5.835	NaN
	<i>p</i> -value	NaN	1.21E-12	NaN	NaN	NaN	1.21E-12	1.16E-13	NaN
MPSO	<i>h</i> -value	0	1	0	0	0	1	1	0
	<i>z</i> -value	NaN	-7.104	NaN	NaN	NaN	-7.104	7.420	NaN
	<i>p</i> -value	NaN	1.21E-12	NaN	NaN	NaN	1.21E-12	5.18E-07	NaN
IPSO	<i>h</i> -value	0	1	0	0	0	1	1	0
	<i>z</i> -value	NaN	-7.104	NaN	NaN	NaN	-7.104	5.019	NaN
	<i>p</i> -value	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1.21E-12
I-GWO	<i>h</i> -value	0	0	0	0	0	0	0	1
	<i>z</i> -value	NaN	NaN	NaN	NaN	NaN	NaN	NaN	-7.104
	<i>p</i> -value	NaN	1.21E-12	NaN	NaN	NaN	1.21E-12	1.16E-13	NaN
Advanced algorithms	AGPSO1	<i>h</i> -value	0	1	0	0	1	1	0
		<i>z</i> -value	NaN	-7.104	NaN	NaN	-7.1041	7.420	NaN
		<i>p</i> -value	NaN	1.21E-12	NaN	NaN	1.21E-12	1.68E-14	NaN
	AGPSO2	<i>h</i> -value	0	1	0	0	1	1	0
		<i>z</i> -value	NaN	-7.104	NaN	NaN	-7.104	7.672	NaN
		<i>p</i> -value	NaN	1.21E-12	NaN	NaN	1.21E-12	7.15E-13	NaN
	AGPSO3	<i>h</i> -value	0	1	0	0	1	1	0
		<i>z</i> -value	NaN	-7.104	NaN	NaN	-7.104	7.176	NaN
		<i>p</i> -value	NaN	NaN	NaN	1.21E-12	1.21E-12	NaN	1.21E-12
	GWOCs	<i>h</i> -value	0	0	0	1	1	0	1
		<i>z</i> -value	NaN	NaN	NaN	-7.104	-7.104	NaN	-7.104

TABLE 24. Results of friedman test of 32 benchmark functions.

Algorithms	CMSRAS	PSO	CS	DA	GWO	MFO	HHO	MVO	SMA	SCA
Avg.	4.45	11.09	7.86	17.16	11.50	13.56	8.92	15.25	7.36	16.11
Rank	1	13	4	19	14	15	6	17	3	18
Algorithms	SOA	WOA	TAPSO	MPSO	IPSO	IGWO	AGPSO1	AGPSO2	AGPSO3	GWOCs
Avg.	13.59	10.95	8.92	9.81	9.02	6.19	9.69	9.48	8.45	10.63
Rank	16	12	6	10	7	2	9	8	5	11

(2) Stiffness constraint

The mid-span deflection of the main beam shall be less than its allowable deflection, as shown in (18).

$$g_2(X) = \frac{F_1 S^3}{1.68 \times 10^6 (3x_1^2 x_2 x_4 + x_1^3 x_3)} - \frac{L}{700} \leq 0 \quad (18)$$

(3) Stability constraint

In order to ensure the local stability of the flange plate of the main girder without the need for stiffening plate and reduce the manufacturing cost and avoid the stress concentration caused by too many welds during the processing of the main girder, a longitudinal stiffened plate should be added to the web. Therefore, local stability conditions are satisfied as

shown in (19) and (20).

$$g_3(X) = \frac{x_2}{x_4} - 60 \leq 0 \quad (19)$$

$$g_4(X) = \frac{x_1}{x_3} - 160 \leq 0 \quad (20)$$

(4) Geometric constraint

In order to reduce the complexity of the welding process, the thickness of the plate should be less than 5 mm. Therefore, the geometric size should meet the geometric constraints, as shown in (21) and (22).

$$g_5(X) = 5 - x_3 \leq 0 \quad (21)$$

$$g_6(X) = 5 - x_4 \leq 0 \quad (22)$$

TABLE 25. The average time-consuming of algorithms on 32 benchmark functions.

ID	CMSRAS	SRA	PSO	CS	DA	GWO	MFO	HHO	MVO	SMA	WOA	IPSO	IGWO
F1	1.902	0.0369	0.295	1.24	81.864	0.63	1.924	0.599	0.778	3.872	0.363	0.323	3.151
F2	1.939	0.037	0.307	1.267	90.076	0.56	1.947	0.479	0.752	3.868	0.387	0.345	3.118
F3	3.13	0.0559	0.89	2.57	105.01	1.181	2.545	2.121	1.344	4.604	1.067	0.937	4.441
F4	1.913	0.0369	0.304	1.246	85.747	0.669	1.945	0.559	0.808	3.92	0.381	0.359	3.092
F5	1.953	0.0329	0.312	1.39	105.35	0.622	1.92	0.646	0.681	3.949	0.356	0.338	3.008
F6	1.94	0.0379	0.302	1.24	80.469	0.651	1.905	0.649	0.778	3.95	0.374	0.332	3.054
F7	3.109	0.049	0.619	1.978	81.574	0.873	2.267	1.324	1.111	4.295	0.689	0.658	3.996
F8	2.441	0.0439	0.396	1.577	104.74	0.699	2.021	0.89	0.706	4.254	0.477	0.413	3.326
F9	2.047	0.036	0.36	1.379	89.061	0.687	1.982	0.738	0.848	3.883	0.377	0.41	3.173
F10	2.037	0.039	0.359	1.418	80.562	0.621	1.99	0.787	0.883	3.976	0.39	0.388	3.178
F11	2.175	0.041	0.404	1.438	84.422	0.678	2.027	0.893	0.926	4.077	0.453	0.425	3.297
F12	4.116	0.076	1.277	3.26	80.703	1.544	2.956	3.238	1.916	5.524	1.439	1.391	5.742
F13	4.087	0.0739	1.261	3.259	80.289	1.553	2.885	3.016	1.749	4.962	1.349	1.293	5.26
F14	4.284	0.0889	1.948	4.292	45.031	1.935	2.151	4.928	2.186	2.976	2.176	1.948	6.172
F15	1.023	0.025	0.195	1.007	76.443	0.276	0.559	0.664	0.458	1.52	0.34	0.231	2.754
F16	0.802	0.024	0.17	0.903	43.375	0.216	0.429	0.583	0.381	1.257	0.302	0.196	2.424
F17	0.872	0.0229	0.139	0.792	47.187	0.195	0.366	0.529	0.399	1.25	0.289	0.182	2.414
F18	0.781	0.022	0.144	0.791	43.492	0.183	0.379	0.501	0.359	1.224	0.277	0.157	2.363
F19	0.935	0.028	0.205	0.932	51.043	0.248	0.492	0.655	0.413	1.399	0.338	0.242	2.565
F20	1.056	0.0289	0.226	0.995	74.03	0.291	0.654	0.661	0.427	1.649	0.341	0.257	2.653
F21	1.457	0.0333	0.307	1.182	59.16	0.359	0.667	0.886	0.537	1.757	0.448	0.319	3.136
F22	1.55	0.03	0.361	1.271	57.379	0.405	0.702	1.026	0.584	1.794	0.504	0.384	3.291
F23	1.62	0.034	0.419	1.415	55.009	0.475	0.777	1.184	0.645	1.858	0.576	0.454	3.463
F24	0.852	0.023	0.147	0.801	44.794	0.192	0.355	0.515	0.366	1.211	0.278	0.173	2.405
F25	0.924	0.025	0.145	0.781	45.347	0.19	0.36	0.52	0.364	1.621	0.274	0.191	2.556
F26	0.842	0.023	0.163	0.815	41.862	0.207	0.374	0.515	0.388	1.235	0.309	0.181	2.611
F27	0.888	0.024	0.149	0.781	12.969	0.187	0.342	0.519	0.375	1.194	0.282	0.182	2.436
F28	0.799	0.024	0.151	0.799	47.232	0.197	0.389	0.538	0.369	1.231	0.284	0.191	2.415
F29	0.76	0.021	0.143	0.774	46.649	0.186	0.349	0.524	0.364	1.203	0.269	0.182	2.36
F30	0.786	0.024	0.139	0.774	45.734	0.193	0.352	0.483	0.352	1.208	0.284	0.174	2.647
F31	0.747	0.024	0.137	0.776	45.943	0.187	0.359	0.504	0.355	1.219	0.277	0.184	2.436
F32	0.792	0.024	0.152	0.789	76.308	0.194	0.371	0.571	0.357	1.208	0.281	0.184	2.413

TABLE 26. Design parameters of the crane box girder.

Parameters	Definition	Value range
F_1	Wheel pressure on the crane box girder	$1.2 \times 10^5 N$
F_2	uniform load on the crane box girder	$1.2 \times 10^4 N$
S	Span of the crane box girder	10.5m
x_1	Height of the crane box girder	760mm
x_2	Width of the crane box girder	340mm
x_3	Thickness of the Web plate	6mm
x_4	Thickness of the flange plate	10mm

where $700 \text{ mm} \leq x_1 \leq 800 \text{ mm}$, $350 \text{ mm} \leq x_2 \leq 400 \text{ mm}$, $5 \text{ mm} \leq x_3 \leq 10 \text{ mm}$, $5 \text{ mm} \leq x_4 \leq 10 \text{ mm}$. CMSRAS algorithm code is compiled by the MATLAB R2019a, and set the population number as 10, the number of iterations as 500, and independently calculation times as 20, respectively. The statistical optimization results of the crane box girder are obtained, as shown in Table 27. The results show that the optimal cross-section area, average cross-section area and worst cross-section area of the crane box girder is 5903.30929 mm^2 , 5903.3094 mm^2 , 5903.3435 mm^2 , respectively. In addition, from the standard deviation and the average iteration time, it shows that the CMSRAS algorithm can efficiently and stably obtain the reasonable optimal design parameters of the crane box girder.

Compared with other methods in literatures, the results of before and after optimization is described as shown

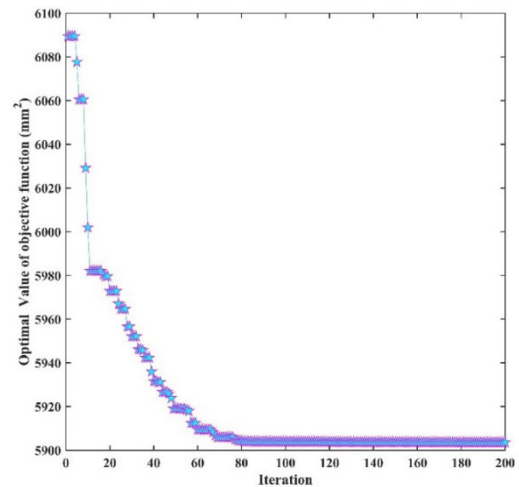


FIGURE 26. Optimal convergence curve of the crane box girder.

in Table 28, and optimal convergence curve of the crane box girder is shown in Fig. 26. These results indicate that CMSRAS algorithm can obtain the best solutions in this engineering problem, reflecting the applicability of CMSRAS to engineering problems.

B. CASE II

The case II is an optimization design problem for a cylindrical pressure vessel with mixed variables (discrete and continuous

TABLE 27. Statistical optimization results of the crane box girder.

Best value	Mean value	Worst value	Standard deviation	Number of mirrors	Iteration number	Average time (s)
5903.30929	5903.3094	5903.3435	0.0062	10	500	0.088

TABLE 28. Comparison and analysis of the best solutions with other references.

Name	Value of design variable (unit: mm)				Value of objective function $f(X)$ (unit: mm ²)
	x_1	x_2	x_3	x_4	
Before optimization	760	340	6	10	7960
CMSRAS	772.3285	350	5.000	5.8333	5903.30929
Error	-1.62%	-2.94%	16.67%	41.67%	25.84%
CGA [69]	799.0784	350.2652	5.021	5.8796	6071.6127
Error	-5.14%	-3.02%	1.63%	41.20%	23.72%
GA-AN2 [70]	791.968	320.440	5.000	6.627	6113.396
Error	-4.21%	-5.75%	16.67%	33.73%	23.20%
Normal way [71]	790	310	5.0	8	6430
Error	-3.95%	-8.82%	16.67%	20%	19.22%

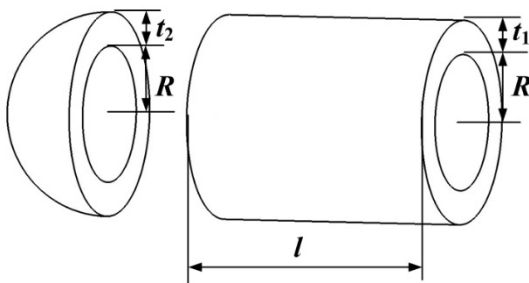


FIGURE 27. Simplified model of cylindrical pressure vessel.

variables), The simplified model of cylindrical pressure vessel is shown in Fig. 27 and Table 29, respectively. It is made of rolled steel plate to form a cylinder. Both ends of the cylinder are sealed by welding two forged hemispherical heads. The design requirements for cylindrical pressure vessels with operating pressure of 3000 psi and minimum volume of 750 ft³ must be in accordance with ASME specifications for boilers and pressure vessels.

The optimization design goal is to minimize the manufacturing cost of cylindrical pressure vessels. The objective function and relevant constraints are determined in Ref. [72]. The objective function of the optimal design of cylindrical pressure vessels is shown in (23) and the constraints are shown in (24) ~ (27).

$$\min f(t_1, t_2, R, l) = 0.6224t_1Rl + 1.7781t_2R^2 + 3.1661t_1^2l + 19.84t_2^2R \quad (23)$$

s.t.

$$g_1 = -t_1 + 0.0193R \leq 0 \quad (24)$$

$$g_2 = -t_2 + 0.00954R \leq 0 \quad (25)$$

$$g_3 = -\pi R^2l - \frac{4}{3}\pi R^3 + 1.296 \times 10^6 \leq 0 \quad (26)$$

$$g_4 = l - 240 \leq 0 \quad (27)$$

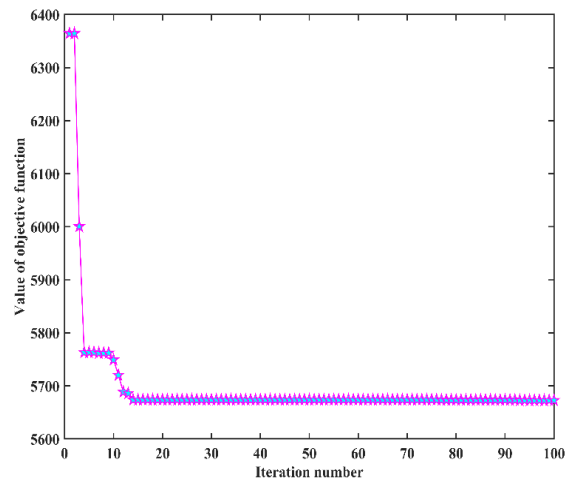


FIGURE 28. Optimal convergence curve of cylindrical pressure vessel.

TABLE 29. Definition and value range of design parameters.

Parameters	Definition	Value range (unit: mm)
t_1	Thickness of spherical shell	$1 \times 0.0625 \leq t_1 \leq 99 \times 0.0625$
t_2	Thickness of ball head	$1 \times 0.0625 \leq t_2 \leq 99 \times 0.0625$
R	Radius of spherical shell	$10 \leq R \leq 200$
l	Length of spherical shell	$10 \leq l \leq 240$

Note: t_1 and t_2 can only take discrete values which are integer multiples of 0.0625 in, respectively; R and l are continuous variables, respectively.

CMSRAS algorithm code is compiled by the MATLAB R2019a, and set the population number as 50, the number of iterations as 1000, and independently calculation times as 20, respectively. The statistical analysis of optimization results is shown in Table 30 and the iterative convergence curve of the objective function is shown in Fig. 28. Comparison and analysis of the best solutions with other references is

TABLE 30. Statistical results of the MSRAS optimization runs executed for cylindrical pressure vessel.

Best value	Mean value	Worst value	Standard deviation	Number of mirrors	Iteration number	Average time (s)
5671.48635	5792.2	6097.82401	124.5258	50	1000	1.93266897

TABLE 31. Comparison and analysis of the best solutions with other references.

Methods	t_1	t_2	R	l	Best value	Mean value	Worst value	Standard deviation	Iteration number
Ref. [73]	0.8125	0.4375	42.09844611	176.63658942	6059.71427196	6090.52614259	6064.33605261	11.28785324	50000
Ref. [74]	0.8125	0.4375	40.3239	200.0000	6288.7445	-	-	-	900000
Ref. [75]	0.8125	0.4375	41.9768	182.2845	6171.0	-	-	-	20000
Ref. [76]	0.8125	0.4375	42.09754674	176.64838674	6059.83905683	6823.60245024	6149.72760669	210.77	20000
Present work	0.8125	0.4375	42.09844147	176.777242	5671.486345	5792.2	6097.82401	124.5258	1000

shown in Table 31. The results show that the best solution of CMSRAS algorithm is better than other references and CMSRAS algorithm has high efficiency and stability to solve the optimization problem of cylindrical pressure vessel.

Therefore, through the above two cases, the results show that the CMSRAS algorithm can better deal with the continuous and discrete nonlinear constrained optimization problems of complex structures. It is an efficient and stable swarm intelligence optimization algorithm, and has a broad application prospects in the optimization design of complex mechanical structures.

VII. CONCLUSION

In this paper, in order to improve the performance of basic SRA algorithm, a chaotic multi-specular reflection optimization algorithm considering shared nodes (CMSRAS) is proposed on the combination of population strategy with shared nodes, improved Tent chaos strategy and Gaussian mutation strategy.

Initially, the influence rule and sensitivity analysis of the performance of CMSRAS algorithm are obtained by the combination of the Sobol's method and RSM method. These results indicated that c_1 , D and n are more sensitive to the performance of CMSRAS algorithm. In addition, to comprehensively evaluate the performance of CMSRAS algorithm, the qualitative analysis of CMSRAS was implemented. Then, 32 benchmark functions were used to evaluate the performance of CMSRAS algorithm, and Wilcoxon sign-rank test and Friedman test were applied to estimate the effectiveness of CMSRAS algorithm more scientifically and reasonably. The experimental results indicate that CMSRAS can maintain a superior balance between exploitation and exploration. From the statistical analysis results, CMSRAS is superior to other optimization algorithms. Two mechanical constrained engineering problems were applied to demonstrate the ability to solve real-world problems. The results show that the CMSRAS algorithm can effectively solve real-world optimization problems.

The proposed CMSRAS algorithm can have better performance attributed to the following viewpoints:

(a) The sharing mechanism was introduced into population strategy to enrich the population diversity and improve the searching efficiency.

(b) The search mechanism can be made more stable by introducing shared nodes into the mirror reflection ring.

(c) Both the improved Tent chaos strategy and Gaussian mutation strategy can alleviate the stagnation problems of the basic SRA to improve the performance of basic SRA.

In future works, on the one hand, CMSRAS algorithm can be extended to multi-objective version to solve multi-objective optimization problems. On the other hand, it also can hybridize with other intelligent optimization algorithms.

REFERENCES

- [1] A. Adha and F. Agus, "Shape optimization of shell structure by using genetic algorithm (GA) method," *IOP Conf. Ser., Earth Environ. Sci.*, vol. 419, Feb. 2020, Art. no. 012034.
- [2] L. Zhang, X. Li, J. Fang, and Z. Long, "Multi-objective positioning design optimization of flexure hinge mechanism considering thermal-mechanical coupling deformation and natural frequency," *Adv. Mech. Eng.*, Vol. 9, no. 1, pp. 1–7, 2017.
- [3] C. Lamini, S. Benhlama, and A. Elbekri, "Genetic algorithm based approach for autonomous mobile robot path planning," *Procedia Comput. Sci.*, vol. 127, pp. 180–189, Jan. 2018.
- [4] G. Shen and X. Huang, Eds., "Advanced research on computer science and information engineering," in *CSIE Communications in Computer and Information Science*, vol. 152. Berlin, Germany: Springer, 2011.
- [5] Y. Cui and X. Hong, "Design and implementation of the air conditioning fuzzy controller based on the genetic algorithms," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2020, pp. 4642–4646.
- [6] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf., Neural Netw.*, Perth, WA, Australia, Nov. 1995, pp. 1942–1948.
- [7] Y. Song, "A fractional PID controller based on particle swarm optimization algorithm," *J. Auton. Intell.*, Vol. 3, no. 1, pp. 1–8, 2020.
- [8] C. Ren, N. An, J. Wang, L. Li, B. Hu, and D. Shang, "Optimal parameters selection for BP neural network based on particle swarm optimization: A case study of wind speed forecasting," *Knowl.-Based Syst.*, vol. 56, pp. 226–239, Jan. 2014.
- [9] J. Robinson and Y. Rahmat-Samii, "Particle swarm optimization in electromagnetics," *IEEE Trans. Antennas Propag.*, vol. 52, no. 2, pp. 397–407, Feb. 2004.

- [10] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, "A new feature selection method to improve the document clustering using particle swarm optimization algorithm," *J. Comput. Sci.*, vol. 25, pp. 456–466, Mar. 2018.
- [11] M. Nouri, A. Bekrar, and A. Jemai, "An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem," *J. Intell. Manuf.*, vol. 29, no. 3, pp. 603–615, Mar. 2018.
- [12] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
- [13] L. Qiu, X. Tian, J. Zhang, C. Gu, and S. Sai, "LIDDE: A differential evolution algorithm based on local-influence-descending search strategy for influence maximization in social networks," *J. Netw. Comput. Appl.*, vol. 178, Mar. 2021, Art. no. 102973.
- [14] D. Zou, S. Li, X. Kong, H. Ouyang, and Z. Li, "Solving the dynamic economic dispatch by a memory-based global differential evolution and a repair technique of constraint handling," *Energy*, vol. 147, pp. 59–80, Mar. 2018.
- [15] Q. Zhang, D. Zou, N. Duan, and X. Shen, "An adaptive differential evolutionary algorithm incorporating multiple mutation strategies for the economic load dispatch problem," *Appl. Soft Comput.*, vol. 78, pp. 641–669, May 2019.
- [16] X. Shen, D. Zou, N. Duan, and Q. Zhang, "An efficient fitness-based differential evolution algorithm and a constraint handling technique for dynamic economic emission dispatch," *Energy*, vol. 186, Nov. 2019, Art. no. 115801.
- [17] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Mixed variable structural optimization using firefly algorithm," *Comput. Struct.*, vol. 89, nos. 23–24, pp. 2325–2336, Dec. 2011.
- [18] M. Gao, X. He, D. Luo, J. Jiang, and Q. Teng, "Object tracking using firefly algorithm," *IET Comput. Vis.*, vol. 7, no. 4, pp. 227–237, Aug. 2013.
- [19] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: A Metaheuristic approach to solve structural optimization problems," *Eng. Comput.*, vol. 29, no. 1, pp. 17–35, Jan. 2013.
- [20] A. Kaveh, T. Bakhshpoori, and M. Barkhori, "Optimum design of multi-span composite box girder bridges using cuckoo search algorithm," *Steel Compos. Struct.*, vol. 17, no. 5, pp. 705–719, Nov. 2014.
- [21] X. Yang and A. Hossein Gandomi, "Bat algorithm: A novel approach for global engineering optimization," *Eng. Comput.*, vol. 29, no. 5, pp. 464–483, Jul. 2012.
- [22] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, Vol. 220, no. 4598, pp. 671–680, 1983.
- [23] M. A. Mohiuddin, S. A. Khan, and A. P. Engelbrecht, "Simulated evolution and simulated annealing algorithms for solving multi-objective open shortest path first weight setting problem," *Int. J. Speech Technol.*, vol. 41, no. 2, pp. 348–365, Sep. 2014.
- [24] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Global Optim.*, vol. 39, no. 3, pp. 459–471, Oct. 2007.
- [25] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009.
- [26] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
- [27] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 495–513, Feb. 2016.
- [28] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.
- [29] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, Vol. 95, pp. 51–67, May 2016.
- [30] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016.
- [31] S. Mirjalili, "Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Comput. Appl.*, vol. 27, no. 4, pp. 1053–1073, May 2016.
- [32] A. Askarzadeh, "A novel Metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm," *Comput. Struct.*, vol. 169, pp. 1–12, Jun. 2016.
- [33] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017.
- [34] G. Dhiman and V. Kumar, "Spotted hyena optimizer: A novel bio-inspired based Metaheuristic technique for engineering applications," *Adv. Eng. Softw.*, vol. 114, pp. 48–70, Dec. 2017.
- [35] G. Dhiman and V. Kumar, "Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems," *Knowl.-Based Syst.*, vol. 165, pp. 169–196, Feb. 2019.
- [36] S. Arora and S. Singh, "Butterfly optimization algorithm: A novel approach for global optimization," *Soft Comput.*, vol. 23, no. 3, pp. 715–734, Feb. 2019.
- [37] 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.
- [38] 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.
- [39] I. Ahmadianfar, O. Bozorg-Haddad, and X. Chu, "Gradient-based optimizer: A new Metaheuristic optimization algorithm," *Inf. Sci.*, vol. 540, pp. 131–159, Nov. 2020.
- [40] Q. Askari, M. Saeed, and I. Younas, "Heap-based optimizer inspired by corporate rank hierarchy for global optimization," *Expert Syst. Appl.*, vol. 161, Dec. 2020, Art. no. 113702.
- [41] A. M. Fathollahi-Fard, M. Hajiaghahi-Keshteli, and R. Tavakkoli-Moghaddam, "The social engineering optimizer (SEO)," *Eng. Appl. Artif. Intell.*, vol. 72, pp. 267–293, Jun. 2018.
- [42] M. Ghasemi, I. F. Davoudkhani, E. Akbari, A. Rahimnejad, S. Ghavidel, and L. Li, "A novel and effective optimization algorithm for global optimization and its engineering applications: Turbulent flow of water-based optimization (TFWO)," *Eng. Appl. Artif. Intell.*, vol. 92, Jun. 2020, Art. no. 103666.
- [43] M. H. Sulaiman, Z. Mustafa, M. M. Saari, and H. Daniyal, "Barnacles mating optimizer: A new bio-inspired algorithm for solving engineering optimization problems," *Eng. Appl. Artif. Intell.*, vol. 87, Jan. 2020, Art. no. 103330.
- [44] O. Olorunda and A. P. Engelbrecht, "Measuring exploration/exploitation in particle swarms using swarm diversity," in *Proc. IEEE Congr. Evol. Comput. (IEEE World Congr. Comput. Intell.)*, Jun. 2008, pp. 1128–1134.
- [45] L. Lin and M. Gen, "Auto-tuning strategy for evolutionary algorithms: Balancing between exploration and exploitation," *Soft Comput.*, vol. 13, pp. 157–168, Jan. 2009.
- [46] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.
- [47] Q. Qi, G. Xu, X. Fan, and J. Wang, "A new specular reflection algorithm," *Adv. Mech. Eng.*, vol. 7, no. 10, pp. 1–10, 2015.
- [48] Q. Qi, J. Wang, and G. Xu, "Reliability-based robust optimization design based on specular reflection algorithm," *Acta Automatica Sinica*, vol. 43, no. 8, pp. 1457–1464, 2017.
- [49] Q. Qi, Q. Dong, and Y. Xin, "Robust optimization design of structures based on the specular reflection algorithm," *Adv. Mech. Eng.*, vol. 11, no. 3, pp. 1–11, 2019.
- [50] W. Hui, "Artificial bee colony algorithm with sharing factor," *Comput. Eng.*, vol. 37, no. 22, pp. 139–142, 2011.
- [51] X. Ziyun, Z. Damin, and C. Zhongyun, "Shared crow algorithm using multi-segment perturbation," *Comput. Eng. Appl.*, vol. 56, no. 2, pp. 55–61, 2020.
- [52] C. Zhong-Yun, Z. Da-Min, and X. Zi-Yun, "Multi-subpopulation based symbiosis and non-uniform Gaussian mutation salp swarm algorithm," *Acta Automatica Sinica*, to be published. Accessed: May 7, 2020. [Online]. Available: <https://doi.org/10.16383/j.aas.c190684>
- [53] F. J. Kuang, W. H. Xu, and Z. Jin, "Artificial bee colony algorithm based on self-adaptive Tent chaos search," *Control Theory Appl.*, vol. 31, no. 11, pp. 1502–1509, 2014.
- [54] L. Shan, H. Qiang, J. Li, and Z. Q. Wang, "Chaos optimization algorithm based Tent map," *Control Decis.*, vol. 2, no. 2, pp. 179–182, 2005.
- [55] N. ZHANG, Z. D. ZHAO, X. A. BAO, and E. al. "Gravitational search algorithm based on improved Tent chaos," *Control Decis.*, vol. 2020, Vol. 35, no. 4, pp. 893–900.
- [56] X. H. Wang, X. Y. Liu, and M. H. Bai, "Population migration algorithm with Gaussian mutation and the steepest descent operator," *Comput. Eng. Applications*, vol. 45, no. 20, pp. 57–60, 2009.
- [57] M. A. Bezerra, R. E. Santelli, E. P. Oliveira, L. S. Villar, and L. A. Escalera, "Response surface methodology (RSM) as a tool for optimization in analytical chemistry," *Talanta*, vol. 76, no. 5, pp. 965–977, Sep. 2008.
- [58] L. Zhang, Z. Long, J. Cai, F. Luo, J. Fang, and M. Y. Wang, "Multi-objective optimization design of a connection frame in macro-micro motion platform," *Appl. Soft Comput.*, vol. 32, pp. 369–382, Jul. 2015.
- [59] N. Bilal, "Implementation of Sobol's method of global sensitivity analysis to a compressor simulation model," in *Proc. 22th Int. Compressor Eng. Conf.*, 2014, pp. 1–17.

- [60] S. Mirjalili, A. Lewis, and A. S. Sadiq, "Autonomous particles groups for particle swarm optimization," *Arabian J. Sci. Eng.*, vol. 39, no. 6, pp. 4683–4697, Jun. 2014.
- [61] T. Ziyu and Z. Dingxue, "A modified particle swarm optimization with an adaptive acceleration coefficients," in *Proc. Asia-Pacific Conf. Inf. Process.*, Jul. 2009, pp. 330–332.
- [62] G. Q. Bao and K. F. Mao, "Particle swarm optimization algorithm with asymmetric time varying acceleration coefficients," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2009, pp. 2134–2139.
- [63] Z. Cui, J. Zeng, and Y. Yin, "An improved PSO with time-varying accelerator coefficients," in *Proc. 8th Int. Conf. Intell. Syst. Design Appl.*, Nov. 2008, pp. 638–643.
- [64] M. H. Nadimi-Shahraki, S. Taghian, and S. Mirjalili, "An improved grey wolf optimizer for solving engineering problems," *Expert Syst. Appl.*, vol. 166, Mar. 2021, Art. no. 113917.
- [65] H. Xu, X. Liu, and J. Su, "An improved grey wolf optimizer algorithm integrated with cuckoo search," in *Proc. 9th IEEE Int. Conf. Intell. Data Acquisition Adv. Comput. Syst., Technol. Appl. (IDAACS)*, Sep. 2017, pp. 490–493.
- [66] F. Vandenbergh and A. Engelbrecht, "A study of particle swarm optimization particle trajectories," *Inf. Sci.*, vol. 176, no. 8, pp. 937–971, Apr. 2006.
- [67] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*. London, U.K.: Chapman & Hall, 2007.
- [68] 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.
- [69] H. Guo, X. Che, and W. Xiao, "Chaos-genetic optimal algorithm and application in mechanical optimal design," *J. Mech. Design*, vol. 20, no. 10, pp. 23–25, 2003.
- [70] Z. Zhengjia, H. Hongzhong, and C. Xin, "A genetic-neural network algorithm in optimum design," *J. Southwest Jiaotong Univ.*, vol. 35, no. 1, pp. 67–70, 2000.
- [71] S. Guozheng, *Optimization Design and Application*. Beijing, China: People's Transportation Press, 1992.
- [72] E. Sandgren, "Nonlinear integer and discrete programming in mechanical design optimization," *J. Mech. Des.*, vol. 112, no. 2, pp. 223–229, Jun. 1990.
- [73] A. Baykasoğlu and F. B. Ozsoydan, "Adaptive firefly algorithm with chaos for mechanical design optimization problems," *Appl. Soft Comput.*, vol. 36, pp. 152–164, Nov. 2015.
- [74] C. A. C. Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Comput. Ind.*, vol. 41, no. 2, pp. 113–127, Mar. 2000.
- [75] S. Akhtar, K. Tai, and T. Ray, "A socio-behavioural simulation model for engineering design optimization," *Eng. Optim.*, vol. 34, no. 4, pp. 341–354, Jan. 2002.
- [76] A. Baykasoğlu, "Design optimization with chaos embedded great deluge algorithm," *Appl. Soft Comput.*, vol. 12, pp. 1055–1067, Mar. 2012.



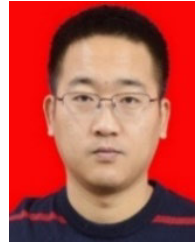
BING MA received the B.S. and M.S. degrees from the Taiyuan University of Science and Technology, Taiyuan, China, in 2012 and 2015, respectively. He is currently pursuing the Ph.D. degree in mechanical engineering with Chang'an University. His research interests include kinetic analysis and simulation, vibration control, optimization design, and reliability evaluation.



PENGMIN LU received the M.S. degree from Lanzhou Jiaotong University, in 1989, and the Ph.D. degree from Southwest Jiaotong University, in 1997. He is currently a Professor with the School of Construction Machinery, Chang'an University. His research interests include dynamic simulation, optimization design, strength analysis, and fatigue life prediction.



LUFAN ZHANG received the Ph.D. degree in mechanical engineering from Xi'an Jiaotong University, Xi'an, China, in 2015. He is currently an Associate Professor with the School of Mechanical and Electrical Engineering, Henan University of Technology, Zhengzhou, China. His research interests include design, simulation, kinetic analysis, motion control, and ultra-precision positioning motion platform.



QISONG QI received the Ph.D. degree in mechanical engineering from the Taiyuan University of Science and Technology, Taiyuan, China, in 2016. He is currently an Associate Professor with the School of Mechanical Engineering, Taiyuan University of Science and Technology. His research interests include modern design theory and design method of metal structure of lifting machinery.



YIXIN CHEN received the B.S., M.S., and Ph.D. degrees from Chang'an University, Xi'an, China, in 2007, 2010, and 2013, respectively. She is currently an Associate Professor with the School of Construction Machinery, Chang'an University. Her research interests include dynamic characteristics of transmission systems, green optimization design, and reliability evaluation.



YONGTAO HU received the Ph.D. degree from Yanshan University, in 2017. He is currently an Instructor with the School of Mechanical Engineering, Henan Institute of Technology, Xinxiang, China. His research interests include fault diagnosis and data analysis.



MENGMENG WANG received the Ph.D. degree in mechanical engineering from China Agricultural University, Beijing, China, in 2017. He is currently an Instructor with the School of Mechanical Engineering, Henan Institute of Technology, Xinxiang, China. His research interests include design, simulation, kinetic analysis, and modern agricultural equipment.



GUOZHU WANG received the Ph.D. degree in control science and engineering from Northeastern University, Shenyang, China, in 2017. He is currently an Instructor with the School of Electrical Engineering and Automation, Henan Institute of Technology, Xinxiang, China. His research interests include industrial process modeling, optimization, and fault diagnosis based on data driven.