



MSSSA: a multi-strategy enhanced sparrow search algorithm for global optimization^{*#}

Kai MENG^{1,2}, Chen CHEN^{†1,2}, Bin XIN^{1,2}

¹School of Automation, Beijing Institute of Technology, Beijing 100081, China

²State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing 100081, China

E-mail: 3120195451@bit.edu.cn; xiaofan@bit.edu.cn; brucebin@bit.edu.cn

Received May 30, 2022; Revision accepted Sept. 13, 2022; Crosschecked Oct. 9, 2022

Abstract: The sparrow search algorithm (SSA) is a recent meta-heuristic optimization approach with the advantages of simplicity and flexibility. However, SSA still faces challenges of premature convergence and imbalance between exploration and exploitation, especially when tackling multimodal optimization problems. Aiming to deal with the above problems, we propose an enhanced variant of SSA called the multi-strategy enhanced sparrow search algorithm (MSSSA) in this paper. First, a chaotic map is introduced to obtain a high-quality initial population for SSA, and the opposition-based learning strategy is employed to increase the population diversity. Then, an adaptive parameter control strategy is designed to accommodate an adequate balance between exploration and exploitation. Finally, a hybrid disturbance mechanism is embedded in the individual update stage to avoid falling into local optima. To validate the effectiveness of the proposed MSSSA, a large number of experiments are implemented, including 40 complex functions from the IEEE CEC2014 and IEEE CEC2019 test suites and 10 classical functions with different dimensions. Experimental results show that the MSSSA achieves competitive performance compared with several state-of-the-art optimization algorithms. The proposed MSSSA is also successfully applied to solve two engineering optimization problems. The results demonstrate the superiority of the MSSSA in addressing practical problems.

Key words: Swarm intelligence; Sparrow search algorithm; Adaptive parameter control strategy; Hybrid disturbance mechanism; Optimization problems

<https://doi.org/10.1631/FITEE.2200237>

CLC number: TP301

1 Introduction

Optimization problems exist in many fields, such as industrial production, artificial intelligence, and medical applications (Deng and Wang, 2017; Ding et al., 2018; Wang MJ and Chen, 2020). Because these problems are multi-peak, highly dimen-

sional, and highly nonlinear, it is difficult to reach the best solution in a limited time period using traditional optimization methods (Gao and Xin, 2019). In addition, traditional methods tend to generate inaccurate solutions. Hence, it is still a great challenge to develop desirable and efficient optimization algorithms to cope with increasingly diverse and complicated problems.

In recent years, meta-heuristic algorithms that imitate natural phenomena, physical laws, and animal's social and life behaviors have attracted widespread attention due to their strong exploration and exploitation abilities in addressing complex engineering and scientific research problems (Xin et al., 2010; Heidari et al., 2020; Ruan and Duan, 2020). Although the optimal solution cannot be guaranteed,

[†] Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 62022015 and 62088101), the Shanghai Municipal Science and Technology Major Project, China (No. 2021SHZDZX0100), and the Shanghai Municipal Commission of Science and Technology Project, China (No. 19511132101)

Electronic supplementary materials: The online version of this article (<https://doi.org/10.1631/FITEE.2200237>) contains supplementary materials, which are available to authorized users

ORCID: Kai MENG, <https://orcid.org/0000-0002-6659-8530>; Chen CHEN, <https://orcid.org/0000-0001-9354-6974>

© Zhejiang University Press 2022

meta-heuristic algorithms can still yield an approximate answer to the question in a relatively short time period because they are stochastic and indeterminate. In general, meta-heuristic algorithms can be classified roughly into four categories (Fister et al., 2013). The first one is swarm intelligence based algorithms such as particle swarm optimization (PSO) (Poli et al., 2007), grey wolf optimization (GWO) (Zhang XQ et al., 2021), the slime mold algorithm (SMA) (Li SM et al., 2020), hunger games search (HGS) (Yang et al., 2021), the RUNge Kutta optimizer (RUN) (Ahmadianfar et al., 2021), colony predation algorithm (CPA) (Tu et al., 2021), Harris hawks optimization (HHO) (Heidari et al., 2019), the dolphin swarm algorithm (DSA) (Wu et al., 2016), and the chimp optimization algorithm (ChOA) (Khishe and Mosavi, 2020). These algorithms update the population by integrating the foraging or other social behaviors of creatures into the individuals' mutual collaboration. The second category is bio-inspired algorithms which update the population through biological operators, such as selection, crossover, mutation, and migration operators. The representative algorithms in this category are the genetic algorithm (GA) (Srinivas and Patnaik, 1994), differential evolution (DE) (Storn and Price, 1997), and biogeography-based optimization (BBO) (Simon, 2008). The third category is physical- or chemical-based algorithms, such as gravitational search algorithm (GSA) (Rashedi et al., 2009), Archimedes optimization algorithm (AOA) (Hashim et al., 2021), equilibrium optimizer (EO) (Faramarzi et al., 2020), synergistic fibroblast optimization (SFO) (Dhivyaprabha et al., 2018), Henry gas solubility optimization (HGSO) (Hashim et al., 2019), and water cycle algorithm (WCA) (Eskandar et al., 2012). The mechanism used to update individuals in these algorithms generally adopts physical or chemical laws. The last category is social or human-based algorithms. These algorithms are inspired by the social and cultural interactions seen in human behaviors, including teaching-learning based optimization (TLBO) (Rao et al., 2011), poor and rich optimization (PRO) (Moosavi and Bardsiri, 2019), the political optimizer (PO) (Askari et al., 2020b), and the heap-based optimizer (HBO) (Askari et al., 2020a).

The sparrow search algorithm (SSA) is a swarm intelligence based meta-heuristic algorithm devel-

oped by Xue and Shen (2020). It mimics the foraging and anti-predation behaviors of sparrows in nature. Compared with other meta-heuristic algorithms, SSA possesses the advantages of simple structure, flexibility, and few parameters. Thus, SSA has been applied to solve various real-life application problems. For instance, Gai et al. (2021) proposed to use SSA for parameter optimization of a deep belief network and then applied it to address the fault diagnosis problem. In Abdulhammed (2022), SSA was used to handle a load-balancing problem. Wang X et al. (2021) introduced SSA to optimize the back-propagation neural network and combined it with a bi-directional long short-term memory network to predict potential threats.

The “no free lunch” theorem (Wolpert and Macready, 1997) states that no one optimization technique is suitable for all optimization problems. SSA is no exception, and it still has some drawbacks in dealing with complex optimization problems, such as low accuracy, premature convergence, difficulty in balancing exploration and exploitation, and a tendency to get stuck in local optima. Therefore, scholars have been devoted to designing enhancement strategies to alleviate these drawbacks or tailor the SSA to solve a specific issue. As shown in Table 1, Tian and Chen (2021) proposed a modified SSA combining multiple strategies and applied it to parameter optimization of long short-term memory neural networks. Zhang Z et al. (2022) presented an improved SSA that performs well for mobile robot path planning in a complex environment. An elite opposite strategy SSA was proposed by Fang et al. (2022) to address the fault diagnosis problem. In Chang et al. (2022), a random walk was integrated with SSA to improve the performance of global search and local search. The developed SSA was employed to optimize the deployment of a fifth-generation wireless communication (5G) network base station. In Zhang CL and Ding (2021), aiming at optimizing the stochastic configuration network, SSA was modified to include a logistic mapping, two self-adaptive hyper-parameters, and a mutation operator. Li XJ et al. (2022) combined chaos, student's t -distribution, and Lévy flight with an SSA and introduced it into a study on identifying robot parameters. Zhu and Yousefi (2021) developed a new technique for optimal parameter identification based on adaptive SSA. They

Table 1 A summary of key works on sparrow search algorithm (SSA)

Reference	Method	Description	Application
Tian and Chen (2021)	ISSA	Chaos, Cauchy variation, and reverse learning are embedded in SSA	Hyper-parameter optimization of the long short-term memory model
Zhang Z et al. (2022)	ISSA	SSA with a neighborhood search strategy	Mobile robot path planning
Fang et al. (2022)	EOSSA	Elite opposite learning strategy and orifice imaging opposite learning strategy are embedded in SSA	Fault diagnosis
Chang et al. (2022)	RWSSA	SSA with random walk	5G base station deployment
Zhang CL and Ding (2021)	CSSA	Chaos, self-adaptive parameters, and mutation are embedded in SSA	Numerical optimization and stochastic configuration network
Li XJ et al. (2022)	ICSSA	Chaos, student's t -distribution, and Lévy flight are embedded in SSA	Parameter identification of robot manipulator
Zhu and Yousefi (2021)	ASSA	Adaptive learning factor, DE/best/1, and crossover operation are embedded in SSA	Parameter identification of proton exchange membrane fuel cell stacks

introduced the adaptive learning factor, DE/best/1, and crossover operation to enrich the search power of the original SSA.

Nevertheless, the standard SSA also suffers from the same shortcomings as meta-heuristic algorithms, including poor initial solution quality, low convergence speed, and ease of falling into local optima when addressing complex and multimodal problems. First, because the initial SSA populations are initialized at random without rules, the initial population cannot be appropriately assigned in the searching space of practical problems, which limits the accuracy of initial solutions. Second, a fixed ratio of producers to scroungers does not accurately reflect the actual optimization searching process and strikes a limited balance between exploration and exploitation. Third, insufficient population diversity causes the SSA to fall into local optima. Therefore, in this study we propose a multi-strategy enhanced sparrow search algorithm (MSSSA) to deal with these problems. First, a modified sparrow generation mechanism, which incorporates chaotic maps and opposition-based learning (OBL), is proposed to overcome the weakness of the poor initial solution quality of the standard SSA. It allows the SSA to rapidly enlarge its searching range in the early stage of the algorithm iteration and generate the initial solution with high quality. Second, adopting the idea of adaptive transformation, a novel evolution operator based on the nonlinear function is designed to replace the fixed ratio of producers to scroungers in the update process, and a better balance between global search and local search is achieved. Third, a hybrid disturbance mechanism, combined with Gaus-

sian mutation (GM) and a firefly algorithm, is incorporated into SSA to enhance the local searching capability by a sudden random jump to choose a more proper solution in the neighboring regions and avoid getting trapped in the local optima.

The main contributions of this paper are as follows:

1. A modified sparrow generation mechanism based on a chaotic map and OBL is developed to generate a high-quality population while increasing the population diversity.
2. An adaptive parameter control strategy, which can dynamically adjust the ratio of producers to scroungers, is adopted to achieve a better balance between intensification and diversification. In addition, the hybrid disturbance mechanism is incorporated into the SSA to improve the ability to jump out of local optima and to improve the convergence speed.
3. The performance of MSSSA is investigated on 50 benchmark functions and two real-world engineering optimization problems. Experimental results demonstrate the effectiveness and superiority of the proposed MSSSA.

2 Sparrow search algorithm

SSA (Xue and Shen, 2020) is a new swarm intelligence optimization algorithm that mimics the foraging and anti-predation behaviors of sparrows. In the SSA, sparrow population is generally separated into two classes: producers and scroungers. Sparrows with better fitness values are called producers. They are responsible for providing foraging areas

or directions for the scroungers due to their wide searching range. The others belong to the scrounger group and follow the producer to forage. In addition, reconnaissance and early-warning mechanisms will be adopted by sparrows when suffering attacks from predators. The SSA implementation process can be described as follows:

Step 1: initialize some important parameters, including the size of the sparrow population n , the maximum iteration number T_{\max} , the dimension of searching space d , the number of producers N_p^* , and the number of scouts n_2 . The initial position of each sparrow is expressed as follows:

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d} \end{bmatrix}. \quad (1)$$

At the end of the initialization phase, sparrows are evaluated based on the fitness functions, and the entire group is divided into producers and scroungers.

Step 2: update the producer's position according to

$$\mathbf{x}_i^{t+1} = \begin{cases} \mathbf{x}_i^t \exp\left(\frac{-i}{\beta T_{\max}}\right), & q < st, \\ \mathbf{x}_i^t + R\mathbf{L}, & q \geq st, \end{cases} \quad (2)$$

where $i = 1, 2, \dots, n$, t is the current iteration number, β is a uniformly distributed random number in the range $(0, 1]$, R is a random number obeying normal distribution, \mathbf{L} denotes a matrix of size $1 \times d$ and each element is equal to 1, and $q \in [0, 1]$ and $st \in [0.5, 1]$ are the alarm value and safety threshold, respectively.

Step 3: update the scrounger's position in accordance with the following formula:

$$\mathbf{x}_i^{t+1} = \begin{cases} R \exp\left(\frac{\mathbf{G}_{\text{worst}} - \mathbf{x}_i^t}{i^2}\right), & i > n/2, \\ \mathbf{S}_{\text{best}} + |\mathbf{x}_i^t - \mathbf{S}_{\text{best}}| \mathbf{C}^+ \mathbf{L}, & \text{otherwise,} \end{cases} \quad (3)$$

where $\mathbf{G}_{\text{worst}}$ is the worst position of sparrows, \mathbf{S}_{best} denotes the current optimal position of producers, and \mathbf{C} is a $1 \times d$ random vector with element values defined as -1 or 1 , $\mathbf{C}^+ = \mathbf{C}^T (\mathbf{C}\mathbf{C}^T)^{-1}$.

Step 4: update the position of scouts by

$$\mathbf{x}_i^{t+1} = \begin{cases} \mathbf{G}_{\text{best}} + \tau |\mathbf{x}_i^t - \mathbf{G}_{\text{best}}|, & f(\mathbf{x}_i^t) > f(\mathbf{G}_{\text{best}}), \\ \mathbf{x}_i^t + S \frac{|\mathbf{x}_i^t - \mathbf{G}_{\text{worst}}|}{f(\mathbf{x}_i^t) - f(\mathbf{G}_{\text{worst}}) + \delta}, & f(\mathbf{x}_i^t) = f(\mathbf{G}_{\text{best}}), \end{cases} \quad (4)$$

where \mathbf{G}_{best} represents the current global optimal position, τ is a random number that obeys normal distribution with a mean value of 0 and a variance of 1, $f(\mathbf{x}_i^t)$ denotes the fitness value of sparrow i , S is a random number in $[-1, 1]$, and δ denotes the minimal constant. When $f(\mathbf{x}_i^t) > f(\mathbf{G}_{\text{best}})$, sparrows are at the group's edge and are highly vulnerable to predators. When $f(\mathbf{x}_i^t) = f(\mathbf{G}_{\text{best}})$, sparrows are in the middle of the population and need to approach the other individuals to reduce the predation risk.

Step 5: compare the new sparrows with the parent sparrows. If the offspring performs better, the parent sparrows will be replaced. Through this approach, the quality of the current sparrow population is gradually improved.

Step 6: output the optimal population and its fitness value if the maximum iterative number T_{\max} is satisfied; otherwise, repeat steps 2-5.

3 The proposed MSSSA

This section describes the proposed MSSSA, which combines three effective strategies, as illustrated in Fig. 1. In the MSSSA, we first introduce a modified sparrow generation mechanism based on a tent map (TM) and OBL (TMOM), which is beneficial for building a high-quality initial population. Then, an adaptive parameter control strategy is adopted to balance the exploration and exploitation trends. Finally, the hybrid disturbance mechanism helps the SSA avoid premature convergence by providing a random jump during the searching process.

3.1 Modified sparrow generation mechanism based on TMOM

The quality of the initial population is of crucial significance in meta-heuristic algorithms because it can affect the accuracy and convergence speed of the algorithm. However, the standard SSA randomly generates the initial population, leading to the problem that the ergodicity and diversity of the swarm

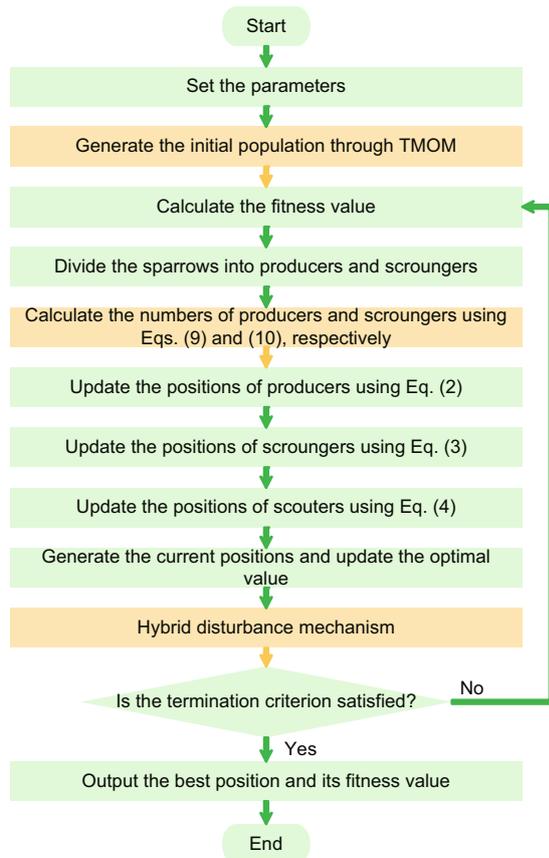


Fig. 1 Flowchart of the proposed multi-strategy enhanced sparrow search algorithm (MSSSA)

are not guaranteed. Considering the shortcoming of random initialization in the SSA, we propose a modified sparrow generation mechanism based on a chaotic map and OBL.

Chaotic maps (Li CH et al., 2017), which possess properties of non-repetition, stochasticity, and ergodicity, are a powerful and fruitful paradigm developed as a way to enhance searching capability and prevent premature convergence. Therefore, it is feasible to replace such randomness by employing a chaotic map. There are various chaotic maps, such as TMs, cubic maps, and logistic maps. Considering that a TM offers higher computational efficiency and better dynamic behavior than the other maps (Fan et al., 2021), it is employed to initialize the sparrow population. The TM is defined as follows:

$$ch_{i,j}(s+1) = \begin{cases} \frac{ch_{i,j}(s)}{0.7}, & ch_{i,j}(s) < 0.7, \\ \frac{10}{3}(1 - ch_{i,j}(s)), & ch_{i,j}(s) \geq 0.7, \end{cases} \quad (5)$$

where $i = 1, 2, \dots, n$ is the sparrow index, $j =$

$1, 2, \dots, d$ is the dimension index, and s represents the iteration counter of the TM. $ch_{i,j}(s) \in (0, 1)$ with an initial value $ch_{i,j}(0) = 0.152$. Then, the chaotic variable $ch_{i,j}(s+1)$ is combined with the optimization variable to form the initial value, which is expressed as

$$x_{i,j} = x_{lb,j} + ch_{i,j}(s+1)(x_{ub,j} - x_{lb,j}), \quad (6)$$

where $x_{lb,j}$ and $x_{ub,j}$ represent the lower and upper bounds of $x_{i,j}$, respectively.

Fig. 2 shows the comparison curve of the TM and random map initialization strategies for 200 iterations in the range $[0, 1]$. It is noticeable that the TM initialization based line distribution is more uniform in extensive ranges than the one generated by the random initialization, which signifies the higher initial population quality of the TM initialization strategy.

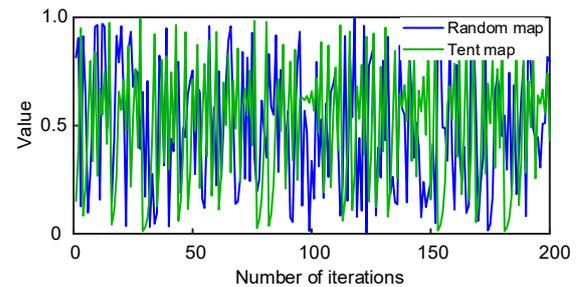


Fig. 2 Comparison of tent map and random map

OBL (Tizhoosh, 2005) is a new technique that chooses the fittest solutions between estimates and counter-estimates. Here, aiming to diversify the sparrow population, OBL is applied to increase the exploration of the searching space on the basis of the high-quality population obtained from TM. According to $x_{i,j}$ obtained from Eq. (6), the related matching opposite solution $ox_{i,j}$ is calculated as follows:

$$ox_{i,j} = x_{ub,j} + x_{lb,j} - x_{i,j}. \quad (7)$$

In the end, the best n sparrows are selected from $2n$ sparrows generated by Eqs. (6) and (7) as the initial population. The mechanism of population initialization using TMOM is shown in Algorithm 1.

3.2 Adaptive parameter control strategy

In the standard SSA, the ratio of producers to scroungers is an essential parameter that influences

the optimization of the objective function. With many producers, the chances of finding new areas through strong global searching capability are increased, while with many scroungers, the algorithm would be capable of exploiting promising regions near the optimal solution through powerful local searching capability. However, the standard SSA adopts a fixed value to control the ratio of producers to scroungers, which makes it difficult to ensure a smooth transition from the global search to the local search in some cases. Therefore, to balance the global exploration and local exploitation abilities, a nonlinear control parameter strategy is designed to adaptively adjust the ratio of producers to scroungers according to the number of iterations. It is formulated as follows:

$$g(t) = 0.3 + \frac{1}{1 + e^{(10t - 0.2T_{\max})/T_{\max}}}, \quad (8)$$

where $g(t)$ is an adaptive controlling parameter.

Then, by introducing the adaptive controlling parameter, the numbers of producers and scroungers can be calculated as

$$N_p^* = g(t)n, \quad (9)$$

$$N_s^* = (1 - g(t))n. \quad (10)$$

Fig. 3 shows the curve of the parameter adaptive transformation strategy. From Fig. 3, it is clear to see that with the increase of t , $g(t)$ gradually decreases.

Algorithm 1 Pseudo-code of the TMOM

- 1: Initialize the parameters: population size n , the maximum iteration number of TM s_{\max} , and the lower ($x_{lb,j}$) and upper ($x_{ub,j}$) bounds of $x_{i,j}$
 - 2: **for** $i \leq n$ **do**
 - 3: **for** $j \leq d$ **do**
 - 4: Set the initial variables $ch_{i,j}(0) = 0.152$
 - 5: **for** $s \leq s_{\max}$ **do**
 - 6: Execute the tent map (TM) operation based on Eq. (5)
 - 7: **end for**
 - 8: Generate the initial value $x_{i,j}$ based on Eq. (6)
 - 9: **end for**
 - 10: **end for**
 - 11: **for** $i \leq n$ **do**
 - 12: **for** $j \leq d$ **do**
 - 13: Compute the value of $ox_{i,j}$ based on Eq. (7)
 - 14: **end for**
 - 15: **end for**
 - 16: Merge TM population $x_{i,j}$ and OBL population $ox_{i,j}$
 - 17: Choose the best n sparrows from the produced population as the initial population
-

In addition, at the early stage of evolution, the adaptive control parameter $g(t)$ remains large, and the relationship between N_p^* and N_s^* tends to be $N_p^* > N_s^*$ according to Eqs. (9) and (10), which facilitates the exploration ability of the SSA. Toward the end of the evolution, $g(t)$ is small and decreases slowly, resulting in $N_p^* < N_s^*$, which helps improve the exploitation ability. Therefore, this strategy contributes to a balance between exploration and exploitation.

3.3 Hybrid disturbance mechanism

The interaction between the producers and scroungers determines the performance of the SSA to some extent. Specifically, when getting stuck in local optima, individuals would escape from the local minimum with the assistance of the fittest producers. Nevertheless, if a majority of sparrows fall into the same local optimum, stagnation of the SSA will be unavoidable. Hence, a hybrid disturbance mechanism, which provides the algorithm with a sudden random jump, is introduced to avoid falling into local optima. The GM and the firefly disturbance (FD) cooperate to realize the hybrid disturbance mechanism, as shown in Fig. 4. The new positions of sparrows are calculated by

$$Y_{\text{new}} = \begin{cases} Y_{\text{GM}}, & r < p, \\ Y_{\text{FD}}, & r \geq p, \end{cases} \quad (11)$$

where r is a uniformly distributed random number from 0 to 1, and p denotes the selection probability for the mutation strategy and FD strategy, which is described as follows:

$$p = \frac{f(\mathbf{x}_i)}{\sum_{i=1}^n f(\mathbf{x}_i)}. \quad (12)$$

In Eq. (11), if p is larger than r , then the GM strategy would be chosen; otherwise, the FD strategy

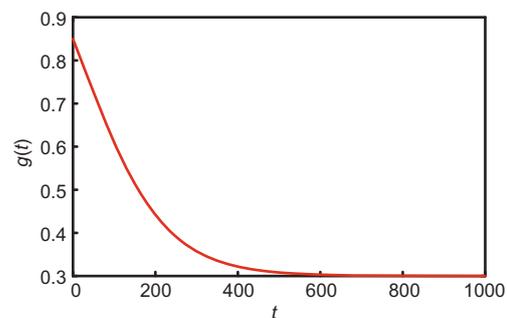


Fig. 3 Fluctuation range of the control parameter $g(t)$

would be adopted. In the following, GM and FD will be separately described in detail.

GM, proposed by Bäck and Schwefel (1993), has been applied to various meta-heuristic algorithms, such as BBO (Zhang XM et al., 2020), EO (Gupta et al., 2020), and the backtracking search algorithm (Nama et al., 2022). Considering that GM can search for valuable information in the vicinity of the current position, it is integrated into SSA to increase the diversity of the population, thereby helping the individuals escape from the local optimal region. The Gaussian density function is defined as follows:

$$f_{\text{Gaussian}(0,\sigma^2)}(z) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(z-\mu)^2}{2\sigma^2}\right), \quad (13)$$

where μ is the mean of the distribution and σ^2 is the variance of each sparrow. Subsequently, μ is set to 0, and the variance σ^2 is set as

$$\sigma^2(t) = a_0 \exp\left(-\frac{t}{T_{\max}}\right), \quad (14)$$

where a_0 is the initial variance with a value of 1.

By Eqs. (13) and (14), a Gaussian distribution operator with variances of varying scales $G(z)$ is obtained. It could be used to interact with two randomly selected locations, thus creating a novel GM operator. Then, each individual in the population mutates according to

$$Y_{\text{GM}} = x_{i,j}^t + G(z) (x_{r1,j}^t - x_{r2,j}^t), \quad (15)$$

where $x_{r1,j}^t$ and $x_{r2,j}^t$ are the j^{th} -dimensional locations of two randomly selected sparrows.

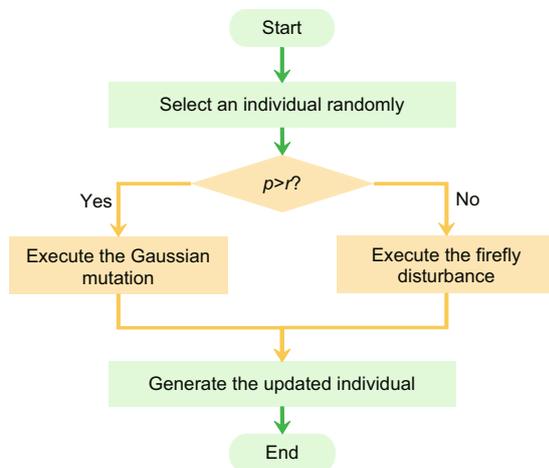


Fig. 4 Flowchart of the hybrid disturbance mechanism

From Eq. (15), the difference of two random individuals is added to the current sparrow's position to obtain the new position. Generally, these differences enable the information of individuals within the population to be shared and increase the information exchange among sparrows. In addition, the developed GM operator dynamically alters the variances with an increasing number of iterations, which allows for a global solution space to be explored in the early searching phase and a local solution space to be explored in the late searching phase. Therefore, the developed GM operators combined with the difference vector can increase the likelihood of sparrows jumping out of the local optima.

FDs (Yelghi and Köse, 2018) are primarily inspired by fireflies' flashing and attraction behaviors. In general, all fireflies approach the firefly that is brighter than themselves. In the proposed FD strategy, the current optimal individual is used to exchange information with other individuals, which is formulated as follows:

$$Y_{\text{FD}} = x_{i,j}^t + \theta_0 \exp(-\gamma r^2) (x_{\text{best},j}^t - x_{i,j}^t) + \alpha \varepsilon_0, \quad (16)$$

$$r = \|\mathbf{x}_i - \mathbf{x}_{\text{best}}\| = \sqrt{\sum_{j=1}^d (x_{i,j} - x_{\text{best},j})^2}, \quad (17)$$

where " $\|\cdot\|$ " stands for the L_2 norm, $x_{\text{best},j}$ is the j^{th} -dimensional position of the best individual, \mathbf{x}_{best} represents the optimal individual, $\theta_0 = 1$ is the attractiveness at $r = 0$, γ represents a light absorption coefficient, α is a random value in the range $[0, 1]$, and ε_0 means a random value obeying a uniform distribution. The cooperation of GM and FD strategies contributes to the improvement of the searching efficiency and effectiveness of the proposed MSSSA.

To sum up, the pseudo-code of the MSSSA is illustrated in Algorithm 2.

3.4 Computational complexity

The computational complexity of an algorithm is a vital criterion for evaluating its performance. In this work, the computational complexity of MSSSA is determined mainly by four parts: initialization, individual position update, hybrid disturbance mechanism, and the best location selection. Suppose that the population size is n , the number of iterations is T , and the dimension of the problem is d . The computational complexity of the initialization is $O(nd + nd)$, counting the chaotic map and

OBL. The computational complexity of the individual position update is $O(ndT)$. The hybrid disturbance mechanism includes GM and FD, so the complexity is $O(ndT + ndT)$. The time required to select sparrows with the best position is $O(nT)$. Hence, the final computational complexity of MSSSA is $O(\text{MSSSA}) = O(nd + nd) + O(ndT) + O(ndT + ndT) + O(nT) = O(n(2d + 3dT + T))$. The complexity of the SSA is $O(\text{SSA}) = O(nd) + O(ndT) + O(nT) = O(n(d + dT + T))$. While the complexity of the MSSSA is slightly larger than that of the SSA, both are in the same order of magnitude. Considering the performance improvement, the increase in computational complexity is acceptable and negligible.

Algorithm 2 Pseudo-code of the proposed MSSSA

Input: the number of sparrows n , the maximum number of iterations T_{\max} , the number of producers N_p^* , the number of scouters n_2 , and the alarm value q

Output: the global best position and its optimal fitness value

```

1: Generate a high-quality initialized population using Algorithm 1
2: Calculate the fitness value for each sparrow
3: Sort all the sparrows according to their fitness values and select the current best and worst individuals
4: while  $t < T_{\max}$  do
5:    $q = \text{rand}(1)$ 
6:   Calculate the numbers of producers and scroungers by Eqs. (9) and (10), respectively
7:   for  $i = 1 : N_p^*$  do
8:     Update the positions of producers by Eq. (2)
9:   end for
10:  for  $i = N_p^* + 1 : n$  do
11:    Update the positions of scroungers by Eq. (3)
12:  end for
13:  for  $i = 1 : n_2$  do
14:    Update the positions of scouters by Eq. (4)
15:  end for
16:  for  $i = 1 : n$  do
17:    Record the new position of the sparrow generated at this time as  $x_{i,j}^{\text{new}}$ 
18:    if  $f(x_{i,j}^{\text{new}}) \leq f(x_{i,j}^t)$  then
19:      Replace the position  $x_{i,j}^t$  by  $x_{i,j}^{\text{new}}$ 
20:    else
21:      Calculate the new position  $Y_{\text{new}}$  through the hybrid disturbance mechanism by Eqs. (11)–(17)
22:      if  $f(Y_{\text{new}}) \leq f(x_{i,j}^t)$  then
23:        Replace the position  $x_{i,j}^t$  by  $Y_{\text{new}}$ 
24:      end if
25:    end if
26:  end for
27:   $t = t + 1$ 
28: end while
29: return the global best position and its fitness value

```

4 Numerical results and comparison

4.1 Test functions and evaluation criteria

To investigate the performance of the MSSSA, we conduct extensive experiments on 40 complex functions from the IEEE CEC2014 and IEEE CEC2019 test suites, as well as 10 classical functions with different dimensions. The specific descriptions of these functions are shown in Tables S1–S3 in the supplementary materials.

The mean value (Mean) and the standard deviation (Std) are employed to compare the potential performance of the MSSSA with other algorithms. In addition, a non-parametric statistical test of the Wilcoxon signed-rank test is carried out at a 5% significance level to check if the MSSSA outperforms the competitive algorithms. Furthermore, the symbols “+/=/-” are used to denote these statistical results, meaning that the proposed MSSSA is better than, similar to, or worse than the competitive optimization algorithms, respectively. Finally, the Friedman test is used to rank the multiple algorithms.

4.2 Comparison algorithms and parameter settings

To test the performance of the developed method, several state-of-the-art algorithms are introduced for comparison, including the SSA, HFPSO (Aydilek, 2018), ALCPSO (Chen WN et al., 2013), CLPSO (Liang et al., 2006), SADE (Qin et al., 2009), RDWOA (Chen HL et al., 2020), I-GWO (Nadimi-Shahraki et al., 2021), SOGWO (Dhargupta et al., 2020), AOSMA (Naik et al., 2021), CKGSA (Mittal et al., 2016), and MSBSO (Liu JN et al., 2020). HFPSO is an enhanced PSO that embeds firefly algorithm mechanisms into PSO to effectively keep a better balance between exploration and exploitation. I-GWO and SOGWO are GWO variants; SOGWO uses an OBL mechanism that is similar to that in the MSSSA. CKGSA is an improved GSA using a chaotic map mechanism. ALCPSO and CLPSO are two canonical PSO variants applied to the CEC2014 test set in related papers and have competitive optimization performances. SADE is an enhanced DE that incorporates an adaptive learning strategy to reduce the computational costs of the standard DE. In the proposed MSSSA, the parameter adaptation strategy is inspired by RDWOA, AOSMA, and MSBSO.

Therefore, these three algorithms are employed for comparison. The parameter settings of these algorithms are listed in Table S4 in the supplementary materials. For fair comparison, the population size and the maximum number of iterations of all algorithms are set to 30 and 1000, respectively. For each benchmark test function, each algorithm is run 30 times to mitigate randomness effects. All algorithms are run in MATLAB R2018b on a server with 8.00 GB of RAM and a 2.60 GHz CPU.

4.3 Ablation study of the MSSSA

As mentioned earlier, three modifications, namely, the TMOM, the adaptive parameter control strategy, and the hybrid disturbance mechanism, are integrated to improve the performance of the standard SSA in the MSSSA. The goal of this subsection is to see if there is synergy between the different modifications. For this purpose, we compare the MSSSA with its three kinds of degradations and the SSA on CEC2014 benchmark functions. Specifically, MSSSAwt means MSSSA without TMOM, MSSSAwa denotes MSSSA without the adaptive parameter control strategy, and MSSSAwh is MSSSA without the hybrid disturbance mechanism. The incomplete variants use the same parameters as the MSSSA. The comparison results provided by the Wilcoxon signed-rank test are shown in Table S5 in the supplementary materials. Moreover, for intuitive comparison, the average ranking bar of the MSSSA and its comparison algorithms is plotted in Fig. 5.

From Table S5, the numbers of functions whose MSSSA optimization effects are better than those of MSSSAwt, MSSSAwa, MSSSAwh, and SSA are 12, 17, 23, and 25 out of 30 functions, respectively.

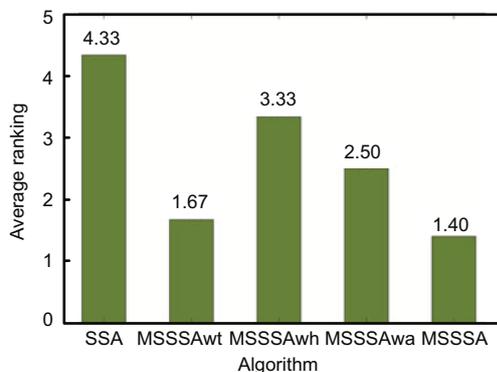


Fig. 5 Ranking statistical bar of MSSSA and its degradative variants

Although the MSSSA generates a worse result than MSSSAwt on F2, it is superior to the other comparison methods on all unimodal functions. It indicates that the adaptive parameter control strategy and the hybrid disturbance mechanism contribute significantly to the exploitation ability of the MSSSA. The MSSSA performs worse than the MSSSAwt, MSSSAwa, MSSSAwh, and SSA on 3, 1, 0, and 0 out of 13 simple multimodal functions, respectively, which shows that the hybrid disturbance mechanism contributes most to the improvement of exploration ability. On the remaining 14 complex test functions, the performance of the MSSSA is worse than those of the MSSSAwt, MSSSAwa, MSSSAwh, and SSA on 4, 2, 1, and 0 test functions, respectively. This proves that the hybrid disturbance mechanism is more conducive to improving the ability to address the complex problems of the MSSSA, followed by the adaptive parameter control strategy and TMOM.

From Fig. 5, the average ranking of the SSA is inferior to that of any of the MSSSA and its degradative variants, indicating that each enhancement strategy is effective and boosts the performance of the SSA to some degree. Among the three degradative variants, MSSSAwh achieves the worst ranking, which means that the hybrid disturbance mechanism is the most responsible for the MSSSA. In addition, the rank from the best to the worst is MSSSA, MSSSAwt, MSSSAwa, MSSSAwh, and SSA, which implies that the MSSSA provides the best optimization performance. It also means that the absence of any of these strategies will negatively affect the algorithm's optimization capability. In conclusion, all enhancement strategies presented in this work are effective and indispensable.

4.4 Performance comparisons on the CEC2014 benchmark suite

4.4.1 Accuracy analysis

Table S6 in the supplementary materials presents the comparison test results of the proposed MSSSA with other algorithms on the 30-dimensional CEC2014 benchmark suite. All the best results in the current test problem are in bold. From the perspective of the mean value, the MSSSA performs optimally on unimodal functions F1 and F3, far exceeding the performances of the other algorithms. On multimodal functions F4–F16, MSSSA provides

the best solutions for eight test problems out of this set. Even though the results obtained by the MSSSA are not optimal on F5, F6, F9, F14, and F15, they still remain at the forefront. On hybrid functions F17–F22, MSSSA significantly outperforms its rivals on F19–F22. MSBSO shows good performance on F17, and SADE and CKGSA perform well on F18. Compared with the first three classes of functions, the composition function is more complicated. Except for ranking second on F24, MSSSA still has the best performance for the remaining test problems. In addition, we evaluate the stability of these algorithms in all the above cases and record the Std values of the objective functions, also shown in Table S6. From the perspective of the Std value, MSSSA is significantly superior to the other counterparts for

most of the examples except MSBSO on F9, F17, and F26, CLPSO on F8 and F11, SADE on F14 and F18, CKGSA on F5, and AOSMA on F24, which verifies the strong stability of the MSSSA.

4.4.2 Statistical analysis

To understand the statistical difference in the results of the MSSSA and the comparison algorithms, a Friedman test and a Wilcoxon signed-rank test are performed. Detailed results are listed in Table S6. Fig. 6 shows the ranking statistics based on Table S6. It can be observed that the MSSSA ranks first with 21 times, second with 7 times, and third with 2 times. As a result, the MSSSA ranks first among the compared algorithms with a mean ranking of 1.3700. The average ranking of MSBSO

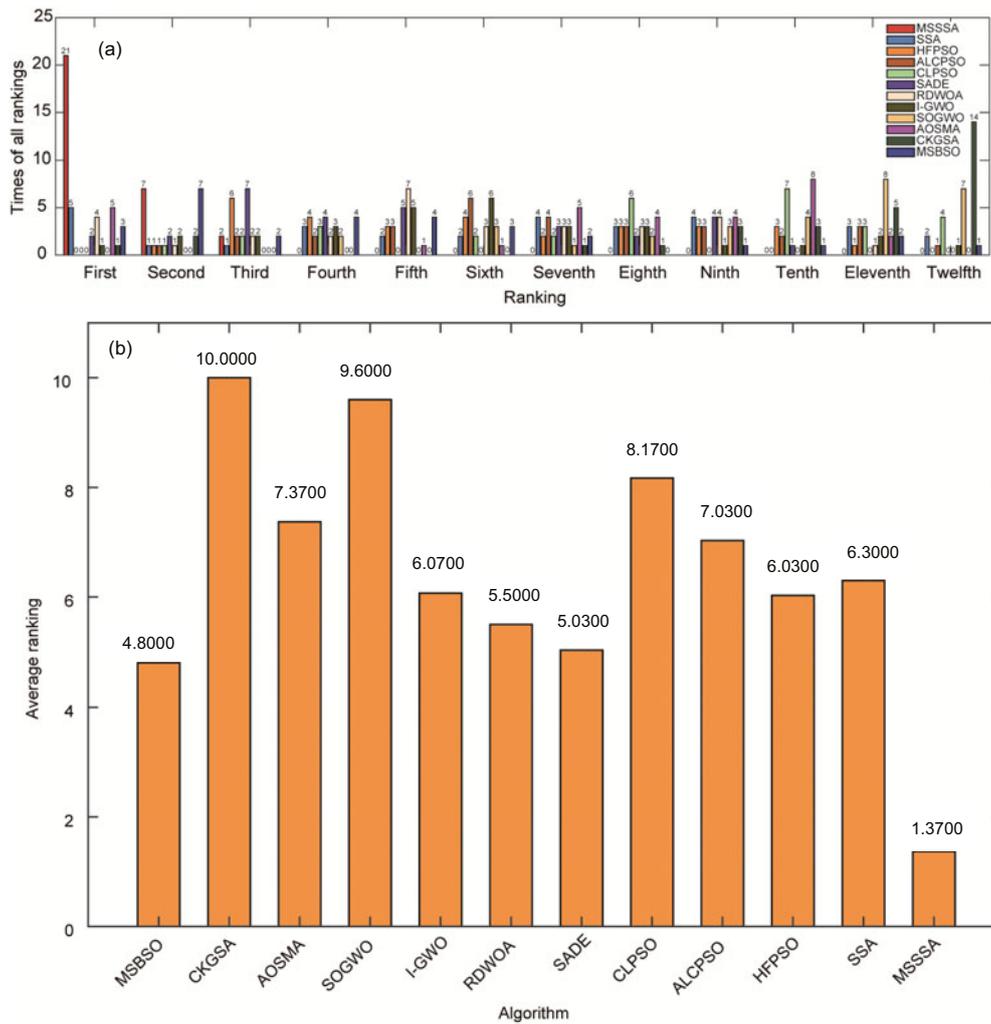


Fig. 6 Ranking statistics on the CEC2014 benchmark functions: (a) ranking statistical bars of 12 algorithms; (b) average ranking bar of 12 algorithms

is 4.8000, second only to MSSSA. SADE, RDWOA, HFPSO, I-GWO, SSA, ALCPSO, AOSMA, CLPSO, SOGWO, and CKGSA rank from third to twelfth, in terms of the total ranking.

According to the data in the column “+/=/-” in Table S6, the outcomes of the MSSSA in solving all test functions are significantly superior to those generated by the HFPSO, ALCPSO, and CLPSO. In addition, the numbers of test functions on which the MSSSA outperforms SSA, SADE, RDWOA, I-GWO, SOGWO, AOSMA, CKGSA, and MSBSO are 24, 28, 26, 28, 29, 25, 28, and 26 out of 30, respectively. Therefore, these results strongly indicate that the best performance achieved by the MSSSA is statistically significant.

4.4.3 Convergence analysis

To intuitively investigate the convergence speed and optimization accuracy of the proposed algorithm, Fig. 7 displays the convergence graphs of the MSSSA, SSA, HFPSO, ALCPSO, CLPSO, SADE, RDWOA, I-GWO, SOGWO, AOSMA, CKGSA, and MSBSO for some representative test functions. For the unimodal functions F2 and F3, MSSSA exhibits significant superiority in terms of the convergence speed and optimization accuracy compared with its opponents. For the multimodal functions F8 and F10, MSSSA can rapidly find the optimum solution. On F11, MSSSA converges slower than RDWOA, CKGSA, and SSA at the first 100 iterations, but after that, both the convergence speed and optimization accuracy are superior to those of these three algorithms. On F16, although the convergence of the MSSSA slows down at some moments, its final optimization accuracy is the best. At the beginning iterations of the hybrid function F20, the convergence speed of the MSSSA is superior to those of all the other algorithms, and it stays in first place until the end. On F29, MSSSA achieves significantly higher convergence speed than any of its rivals. This is owing to the collaboration of the parameter adaptive transformation strategy and the hybrid disturbance mechanism, where the former is conducive to accelerating the convergence and the latter is capable of escaping from the local optima. In general, the convergence performance of the MSSSA is far superior to those of its comparison methods.

4.5 Performance comparisons on the CEC2019 benchmark suite

To further assess the capability of the MSSSA in dealing with more complex problems, a series of experiments are conducted on 10 functions from the CEC2019 test set in this part. The experimental results are shown in Table S7 in the supplementary materials. From Table S7, we can observe that the MSSSA achieves almost the best overall performance. In detail, MSSSA outperforms all the opponents on F31–F33, F36, and F40, and has the second-best position on F34, F37, and F39. Although slightly underperformed on F35 and F38, within the error range, the searching accuracy of the MSSSA is still satisfactory. In terms of the Std, MSSSA is significantly superior to the other counterparts for most examples except CKGSA on F35, CLPSO on F37, and HFPSO on F40, which indicates the strong stability of the MSSSA. Moreover, according to the statistical results in Table S7, the proposed MSSSA achieves the best ranking and performs better than its opponents in most test functions. In conclusion, MSSSA has outstanding capability in dealing with the highly challenging CEC2019 test functions.

4.6 Balance and diversity analysis

To analyze the balance between exploration and exploitation and the diversity of the MSSSA in the entire iteration process compared to the SSA, we conduct balance and diversity tests on CEC2014 and CEC2019 test functions. Due to space constraints, not all the test results are displayed. The results are analyzed on four representative test functions, including the unimodal function F1, the simple multimodal function F12, the hybrid function F18, and the expanded Schaffer’s F6 function (F38). The study results are plotted in Figs. 8 and 9.

Fig. 8 shows the balance analysis of the two algorithms. The incremental-decremental curve exhibits an upward trend when the exploration percentage is bigger than the exploitation percentage at some point during the searching process; otherwise, a downward trend emerges. When the incremental-decremental curve reaches its peak, it indicates that exploration and exploitation achieve a state of balance.

In Fig. 8, the exploration behavior of the two algorithms dominates in the initial stage, but the

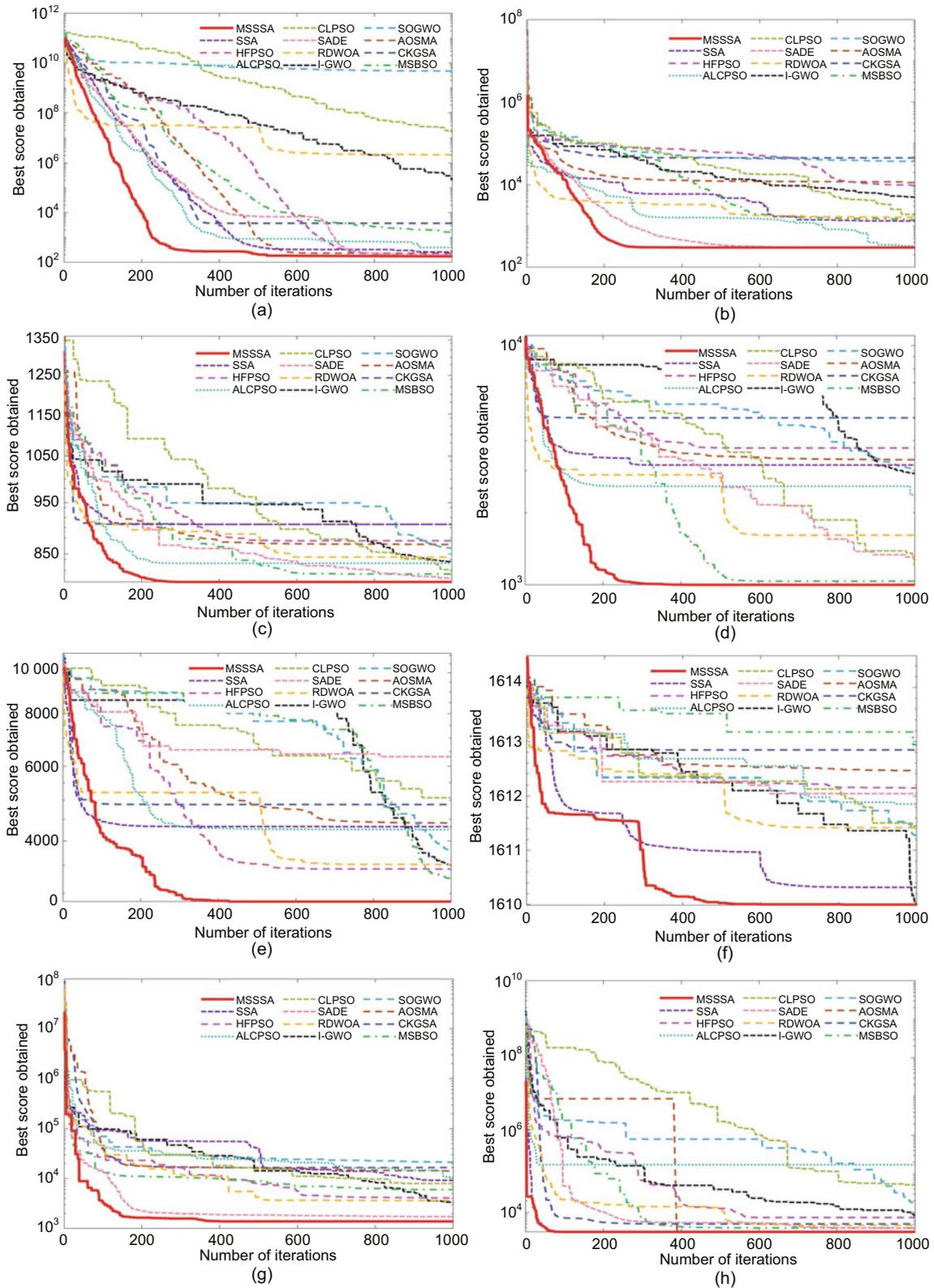


Fig. 7 Convergence curves of the MSSSA and the other algorithms on some selected CEC2014 functions: (a) F2; (b) F3; (c) F8; (d) F10; (e) F11; (f) F16; (g) F20; (h) F29

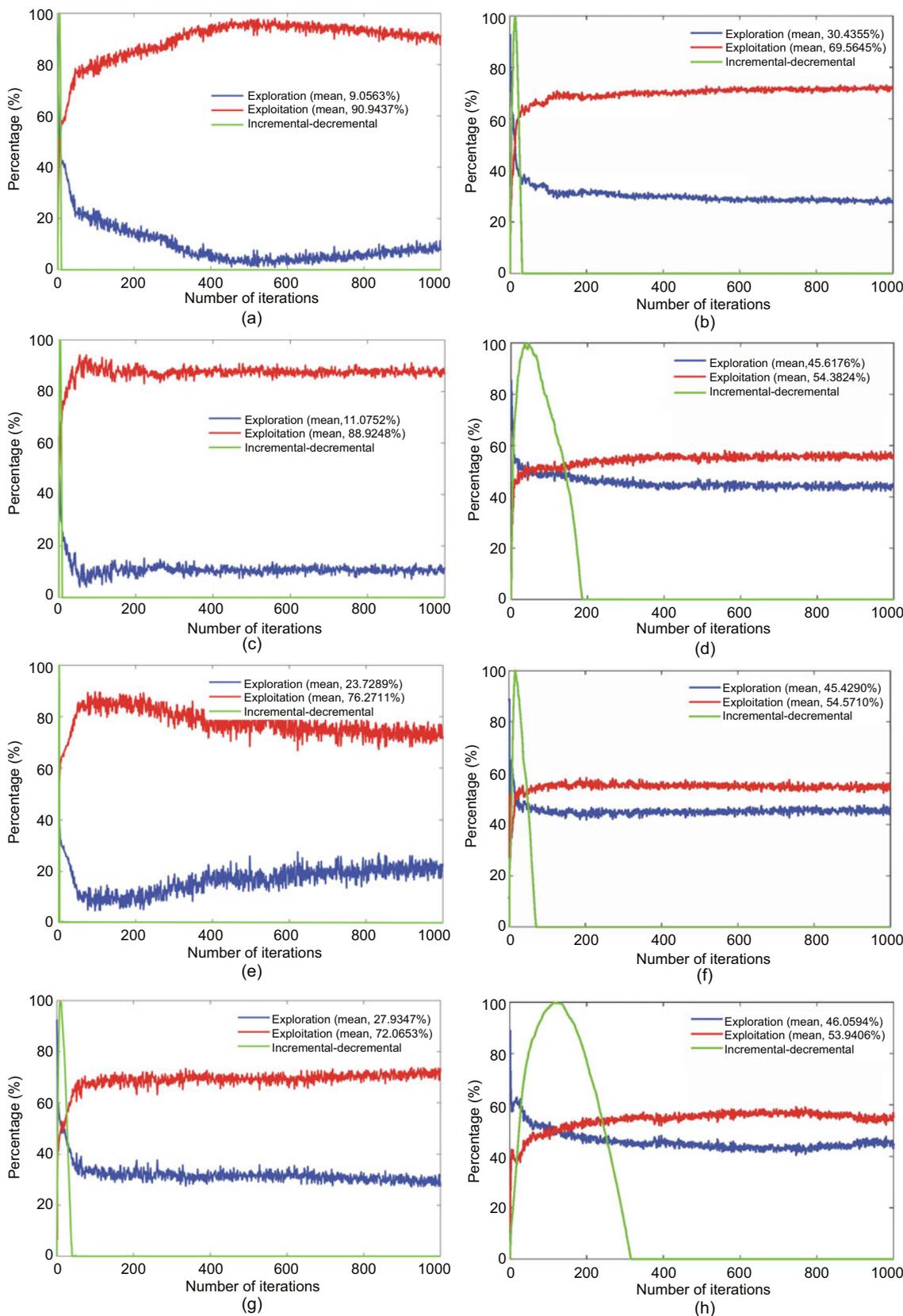


Fig. 8 Balance analysis of the MSSSA and SSA on F1, F12, F18, and F38: (a) MSSSA on F1; (b) SSA on F1; (c) MSSSA on F12; (d) SSA on F12; (e) MSSSA on F18; (f) SSA on F18; (g) MSSSA on F38; (h) SSA on F38

exploitation behavior quickly dominates afterward. A reasonable explanation for this phenomenon is that meta-heuristic algorithms usually implement the diversification mechanism before conducting the intensification mechanism. As can be observed from the curves of exploration and exploitation, the percentage of exploration of the standard MSSSA is smaller than that of the SSA on F1, F12, F18, and F38, suggesting that the MSSSA enters the local searching phase more rapidly than the SSA. At this time, the local searching capability of the MSSSA increases quickly, which is conducive to digging deep into a more promising region to find high-quality results. As can be observed from the incremental-decremental curves, the speeds of rising and reaching the peak points of the MSSSA are higher than those of the SSA, indicating that the MSSSA can keep a balance of exploration and exploitation earlier than the SSA. With the combination of the adaptive parameter control strategy and hybrid disturbance mechanism, the local searching capability is strengthened and the convergence speed is improved. Therefore, we can conclude that MSSSA can main-

tain a better balance of exploration and exploitation, so that the convergence speed and optimization accuracy of MSSSA are superior to those of the SSA.

Fig. 9 shows the diversity of the MSSSA and SSA algorithms, where the horizontal axis represents the number of iterations and the ordinate denotes the average Euclidean distance between searching agents. As can be observed from the figure, the diversity value of the MSSSA is lower than that of the SSA in the early searching phase, due to the TMOM strategy used in the initialization. This strategy ensures that the MSSSA can obtain high-quality initialization individuals and enter the local searching phase faster. Then, the diversity curve gradually decreases as the number of iterations increases and stabilizes in the end. The trend of stabilization of the MSSSA appears to be much earlier than the standard SSA, indicating that the combination of three strategies effectively upsurges SSA's convergence.

4.7 Scalability test on the MSSSA

To measure the capability of the proposed algorithm in dealing with different dimensional problems,

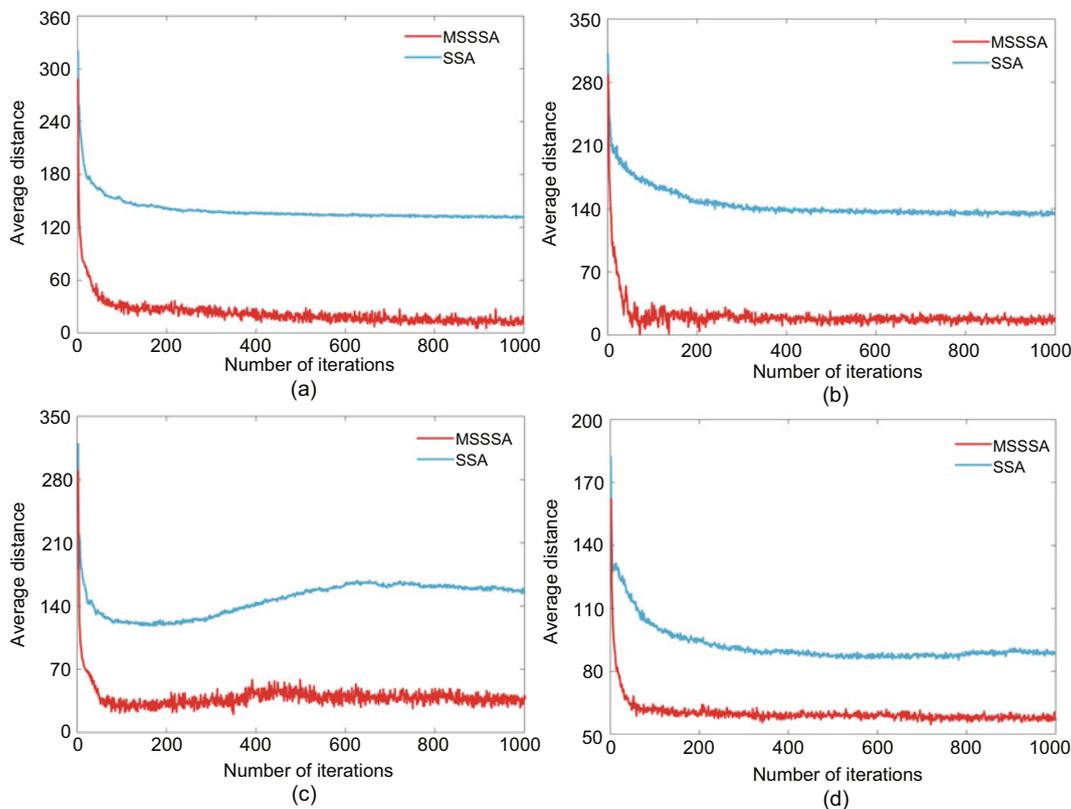


Fig. 9 Diversity analysis of the MSSSA and SSA on F1 (a), F12 (b), F18 (c), and F38 (d)

the MSSSA is compared with the SSA on 30-, 50-, 100-, 200-, 500-, and 1000-dimensional functions. The test cases selected for the experiments are 10 representative classical benchmark functions. The experimental results are shown in Table S8 in the supplementary materials. As can be observed from Table S8, the performance of the SSA decreases significantly with the increase of dimension, while the capability of the MSSSA is steadily enhanced. Compared with the SSA, the MSSSA obtains better accuracy and greater stability on problems of varying complexity, indicating the strength and potential of the MSSSA in dealing with high-dimensional optimization problems. Therefore, it can be concluded that the proposed MSSSA has better scalability.

5 MSSSA for solving constrained engineering optimization problems

In this section, two classical constrained engineering problems, including pressure vessel design (PVD) and speed reducer design (SRD), are employed to investigate the effectiveness of the MSSSA in practical applications. The death penalty is adopted to deal with constraints in this study. The parameter settings of all the experiments are the same as in Section 4.2.

5.1 PVD problem

The first engineering problem, depicted in Fig. 10, is to obtain the minimum cost of the pressure vessel by optimizing four related variables, including the thickness of the shell T_s , thickness of the head T_h , inner radius R , and length of the cylindrical section of the vessel L . The mathematical model is described as follows:

$$\mathbf{x} = [x_1, x_2, x_3, x_4] = [T_s, T_h, R, L], \quad (18)$$

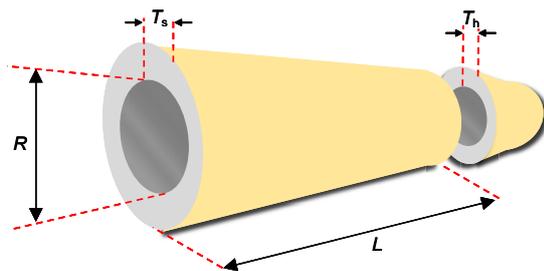


Fig. 10 Pressure vessel design problem

$$\begin{aligned} \min f(\mathbf{x}) = & 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 \\ & + 3.1661x_1^2x_4 + 19.84x_1^2x_3, \end{aligned} \quad (19)$$

subject to

$$\begin{aligned} g_1(\mathbf{x}) &= -x_1 + 0.0193x_3 \leq 0, \\ g_2(\mathbf{x}) &= -x_2 + 0.00954x_3 \leq 0, \\ g_3(\mathbf{x}) &= -\pi x_3^2 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0, \\ g_4(\mathbf{x}) &= x_4 - 240 \leq 0, \end{aligned}$$

$$\begin{aligned} 0 \leq x_1 \leq 99, \quad 0 \leq x_2 \leq 99, \\ 10 \leq x_3 \leq 200, \quad 10 \leq x_4 \leq 200. \end{aligned}$$

For the PVD problem, the best solutions and statistical results of comparing the MSSSA with the other algorithms are recorded in Tables 2 and 3, respectively, where “Best” represents the optimal value and “Worst” is the worst result. As can be observed in Table 2, the MSSSA is superior to the other algorithms and acquires an optimal

Table 2 Best solutions of all the methods on the pressure vessel design problem

Algorithm	T_s	T_h	R	L	Best cost
MSSSA	0.7782	0.3846	40.3196	200.0000	5885.3328
SSA	0.7816	0.3864	40.4980	197.5312	5891.2446
HFPSSO	0.7782	0.3847	40.3196	200.0000	5891.5859
ALCPSO	0.7782	0.3847	40.3196	200.0000	5891.5859
CLPSO	0.7824	0.3867	40.5352	197.2093	5903.5077
SADE	0.7782	0.3847	40.3196	200.0000	5891.5859
RDWOA	0.7782	0.3847	40.3198	199.9971	5891.5929
I-GWO	0.7788	0.3850	40.3516	199.5766	5893.5039
SOGWO	0.7793	0.3852	40.3588	199.5329	5897.7459
AOSMA	0.7782	0.3847	40.3531	199.9515	5891.7025
CKGSA	0.8808	0.4354	45.6354	137.2379	6093.9033
MSBSO	0.7782	0.3847	40.3196	200.0000	5891.5859

Best result is in bold. T_s , T_h , R , and L are the optimum variables

Table 3 Statistical results of all the methods on the pressure vessel design problem

Algorithm	Best	Mean	Worst	Std
MSSSA	5885.3328	5890.6000	5949.2423	15.3203
SSA	5891.2446	6432.1386	7318.9684	454.5260
HFPSSO	5891.5859	6280.0335	7345.4732	375.5908
ALCPSO	5891.5859	6256.0094	7345.4732	415.7870
CLPSO	5903.5077	6069.2958	6325.9864	90.5242
SADE	5891.5859	5891.5859	5891.5859	0.0000
RDWOA	5891.5929	6694.9463	7345.4739	678.3317
I-GWO	5893.5039	5894.8587	5898.5681	1.0787
SOGWO	5897.7459	5999.9967	7026.6903	261.6701
AOSMA	5891.7025	6632.4350	7345.4056	635.9922
CKGSA	6093.9033	89198.8909	227939.4854	74928.6694
MSBSO	5891.5859	5891.5859	5891.5859	0.0000

Best results are in bold

value of 5885.3328 corresponding to the optimal solution [0.7782, 0.3846, 40.3196, 200.0000]. In addition, SADE and MSBSO obtain the best stability, with I-GWO ranking the third and MSSSA ranking the fourth, as shown in Table 3. The “no free lunch” theorem states that the MSSSA cannot solve all problems successfully. Even though the stability of the MSSSA is not the best, it is still very competitive. Therefore, the MSSSA has superior performance in addressing the PVD optimization problem.

5.2 SRD problem

Another engineering problem optimized in this study is the SRD problem, as shown in Fig. 11. This problem aims to minimize the total weight determined by decision variables, i.e., the number of teeth on a pinion z , face width b , module of teeth m , length of the first shaft l_1 , diameter of the first shaft d_1 , length of the second shaft l_2 , and diameter of the second shaft d_2 . The mathematical model is described as follows:

$$\mathbf{x} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7] = [b, m, z, l_1, l_2, d_1, d_2], \tag{20}$$

$$\begin{aligned} \min f(\mathbf{x}) = & 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 \\ & - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777 \\ & \cdot (x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2), \end{aligned} \tag{21}$$

subject to

$$\begin{aligned} g_1(\mathbf{x}) &= \frac{27}{x_1x_2^2x_3} - 1 \leq 0, \\ g_2(\mathbf{x}) &= \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0, \\ g_3(\mathbf{x}) &= \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \leq 0, \\ g_4(\mathbf{x}) &= \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \leq 0, \end{aligned}$$

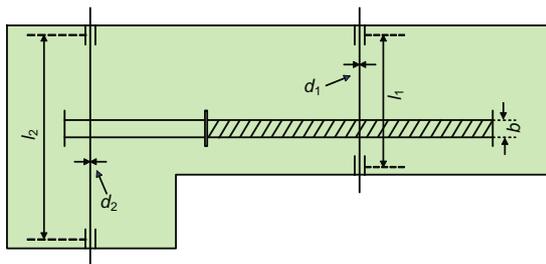


Fig. 11 Speed reducer design problem

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

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

$$g_7(\mathbf{x}) = \frac{x_2x_3}{40} - 1 \leq 0,$$

$$g_8(\mathbf{x}) = \frac{5x_2}{x_1} - 1 \leq 0,$$

$$g_9(\mathbf{x}) = \frac{x_1}{12x_2} - 1 \leq 0,$$

$$g_{10}(\mathbf{x}) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0,$$

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

$$\begin{aligned} 2.6 \leq x_1 \leq 3.6, \quad 0.7 \leq x_2 \leq 0.8, \quad 17 \leq x_3 \leq 28, \\ 7.3 \leq x_4 \leq 8.3, \quad 7.3 \leq x_5 \leq 8.3, \quad 2.9 \leq x_6 \leq 3.9, \\ 5.0 \leq x_7 \leq 5.5. \end{aligned}$$

For the SRD problem, the optimal results and statistical indices obtained by the involved methods are recorded in Tables 4 and 5, respectively. From these tables, we can observe that the best and mean values of the MSSSA are superior to those of its opponents, with the Std slightly larger than those of SADE, MSBSO, RDWOA, CLPSO, and I-GWO, indicating that the MSSSA has an advantage on the SRD problem.

From these two engineering cases, we can conclude that the MSSSA is capable of dealing with constrained engineering design problems well.

6 Discussions

In this paper, we propose an enhanced SSA by integrating TMOM, an adaptive parameter control strategy, and a hybrid disturbance mechanism. Specifically, TMOM combines the TM and OBL to generate high-quality initial sparrow populations. The adaptive parameter control strategy is designed to adjust the ratio of producers to scroungers with an increased number of iterations, which contributes to establishing a balance between exploration and exploitation. The hybrid disturbance mechanism aims to facilitate the MSSSA, strengthening local exploration capability, and effectively jumping out of the local optimal region. The ablation experiments are conducted to investigate the promotion effect of the three strategies on the SSA. Experimental results verify the effectiveness of integrating TMOM, an

Table 4 Best solutions of all the methods on the speed reducer design problem

Algorithm	Optimal values for variables							Best cost
	b	m	z	l_1	l_2	d_1	d_2	
MSSSA	3.5000	0.7000	17.0000	7.3000	7.7153	3.3502	5.2866	2994.4711
SSA	3.5000	0.7000	17.0000	7.3000	7.7159	3.3502	5.2872	2994.7097
HFPSO	3.5000	0.7000	17.0000	7.3000	7.7159	3.3502	5.2872	2994.7097
ALCPSO	3.5000	0.7000	17.0000	7.3000	7.7159	3.3502	5.2872	2994.7097
CLPSO	3.5000	0.7000	17.0000	7.3000	7.7159	3.3502	5.2872	2994.7097
SADE	3.5000	0.7000	17.0000	7.3000	7.7159	3.3502	5.2872	2994.7097
RDWOA	3.5000	0.7000	17.0000	7.3000	7.7159	3.3502	5.2872	2994.7097
I-GWO	3.5000	0.7000	17.0000	7.3000	7.7159	3.3502	5.2872	2994.7097
SOGWO	3.5028	0.7002	17.0000	7.4308	7.7667	3.3567	5.2875	3000.7795
AOSMA	3.5000	0.7000	17.0000	7.3002	7.7164	3.3502	5.2872	2994.7221
CKGSA	3.5926	0.7134	17.1221	7.7464	8.1030	3.4464	5.3013	3163.2207
MSBSO	3.5000	0.7000	17.0000	7.3000	7.7159	3.3502	5.2872	2994.7097

Best result is in bold

Table 5 Statistical results of all the methods on the speed reducer design problem

Algorithm	Best	Mean	Worst	Std
MSSSA	2994.4711	2994.6003	2995.1814	0.2369
SSA	2994.7097	2995.0208	3005.0435	1.7041
HFPSO	2994.7097	3001.0011	3033.9863	13.5509
ALCPSO	2994.7097	2994.8501	2998.8993	0.7648
CLPSO	2994.7097	2994.7097	2994.7098	0.0000
SADE	2994.7097	2994.7097	2994.7097	0.0000
RDWOA	2994.7097	2994.7097	2994.7098	0.0000
I-GWO	2994.7097	2994.7097	2994.7100	0.0001
SOGWO	3000.7795	3006.2565	3014.5684	3.6924
AOSMA	2994.7221	2999.0901	3013.5460	5.0783
CKGSA	3163.2207	3635.4198	4666.9643	378.3087
MSBSO	2994.7097	2994.7097	2994.7097	0.0000

Best results are in bold. The actual variances of the CLPSO and RDWOA are 0.000 01, so they are not marked in bold

adaptive parameter control strategy, and a hybrid disturbance mechanism to improve the SSA. In addition, compared with some advanced algorithms in two benchmark function sets, the MSSSA obtains the expected results and shows superior performance. The optimization accuracy and convergence speed are more prominent regardless of the IEEE CEC2014 and IEEE CEC2019. The balanced diversity experimental results show that the MSSSA has an excellent capacity to avoid getting stuck at the local optima, which is conducive to obtaining a high-quality solution. Furthermore, the MSSSA obtains satisfactory results on two engineering optimization problems, indicating that the proposed algorithm is of great practical value.

Nevertheless, the MSSSA still has some deficiencies in particular aspects. First of all, the convergence speed of the MSSSA still has space for im-

provement when addressing multimodal functions. Also, the computation time of the MSSSA is slightly higher than that of the conventional SSA. Hence, when handling some practical problems, there must be a good balance between accuracy and computational cost. In the future, the MSSSA can also be applied to other real-world problems, such as fault detection (Long et al., 2022), scheduling (Zhang GH et al., 2021), and medical diagnosis (Liu JC et al., 2022).

7 Conclusions and future work

In this study, a novel MSSSA is proposed to solve global optimization problems. In the MSSSA, the TM and OBL are first combined to generate a high-quality initial population. Then, an adaptive parameter control strategy is applied to balance exploration and exploitation. Finally, a hybrid disturbance mechanism is embedded in the SSA to increase the probability of jumping out of local optima. Two benchmark function sets (IEEE CEC2014 and IEEE CEC2019) and 10 classical functions are used to evaluate the performance of the proposed MSSSA. Compared with the SSA and 10 state-of-the-art algorithms, the MSSSA has superior performance in terms of solution accuracy, convergence speed, scalability, and stability. Moreover, the proposed MSSSA is applied to deal with two real-world optimization problems. Experimental results indicate that the MSSSA is a practicable and efficient method for complex engineering optimization problems. Although the MSSSA has been demonstrated

to be a highly effective single-objective algorithm for continuous optimization problems, the potential for multi-objective optimization is still waiting to be developed as a future attempt. In addition, the MSSSA could be hybridized with other meta-heuristic algorithms to enhance its overall performance.

Contributors

Kai MENG designed the research and drafted the paper. Chen CHEN guided the research. Bin XIN revised and finalized the paper.

Compliance with ethics guidelines

Kai MENG, Chen CHEN, and Bin XIN declare that they have no conflict of interest.

References

- Abdulhammed OY, 2022. Load balancing of IoT tasks in the cloud computing by using sparrow search algorithm. *J Supercomput*, 78(3):3266-3287. <https://doi.org/10.1007/s11227-021-03989-w>
- Ahmadianfar I, Heidari AA, Gandomi AH, et al., 2021. RUN beyond the metaphor: an efficient optimization algorithm based on Runge Kutta method. *Expert Syst Appl*, 181:115079. <https://doi.org/10.1016/j.eswa.2021.115079>
- Askari Q, Saeed M, Younas I, 2020a. Heap-based optimizer inspired by corporate rank hierarchy for global optimization. *Expert Syst Appl*, 161:113702. <https://doi.org/10.1016/j.eswa.2020.113702>
- Askari Q, Younas I, Saeed M, 2020b. Political optimizer: a novel socio-inspired meta-heuristic for global optimization. *Knowl-Based Syst*, 195:105709. <https://doi.org/10.1016/j.knsys.2020.105709>
- Aydilek İB, 2018. A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems. *Appl Soft Comput*, 66:232-249. <https://doi.org/10.1016/j.asoc.2018.02.025>
- Bäck T, Schwefel HP, 1993. An overview of evolutionary algorithms for parameter optimization. *Evol Comput*, 1(1):1-23. <https://doi.org/10.1162/evco.1993.1.1.1>
- Chang ZZ, Gu QH, Lu CW, et al., 2022. 5G private network deployment optimization based on RWSSA in open-pit mine. *IEEE Trans Ind Inform*, 18(8):5466-5476. <https://doi.org/10.1109/TII.2021.3132041>
- Chen HL, Yang CJ, Heidari AA, et al., 2020. An efficient double adaptive random spare reinforced whale optimization algorithm. *Expert Syst Appl*, 154:113018. <https://doi.org/10.1016/j.eswa.2019.113018>
- Chen WN, Zhang J, Lin Y, et al., 2013. Particle swarm optimization with an aging leader and challengers. *IEEE Trans Evol Comput*, 17(2):241-258. <https://doi.org/10.1109/TEVC.2011.2173577>
- Deng J, Wang L, 2017. A competitive memetic algorithm for multi-objective distributed permutation flow shop scheduling problem. *Swarm Evol Comput*, 32:121-131. <https://doi.org/10.1016/j.swevo.2016.06.002>
- Dhargupta S, Ghosh M, Mirjalili S, et al., 2020. Selective opposition based grey wolf optimization. *Expert Syst Appl*, 151:113389. <https://doi.org/10.1016/j.eswa.2020.113389>
- Dhivyaprabha TT, Subashini P, Krishnaveni M, 2018. Synergistic fibroblast optimization: a novel nature-inspired computing algorithm. *Front Inform Technol Electron Eng*, 19(7):815-833. <https://doi.org/10.1631/FITEE.1601553>
- Ding SX, Chen C, Xin B, et al., 2018. A bi-objective load balancing model in a distributed simulation system using NSGA-II and MOPSO approaches. *Appl Soft Comput*, 63:249-267. <https://doi.org/10.1016/j.asoc.2017.09.012>
- Eskandar H, Sadollah A, Bahreininejad A, et al., 2012. Water cycle algorithm—a novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput Struct*, 110-111:151-166. <https://doi.org/10.1016/j.compstruc.2012.07.010>
- Fan Q, Chen ZJ, Li Z, et al., 2021. A new improved whale optimization algorithm with joint search mechanisms for high-dimensional global optimization problems. *Eng Comput*, 37(3):1851-1878. <https://doi.org/10.1007/s00366-019-00917-8>
- Fang QC, Shen B, Xue JK, 2022. A new elite opposite sparrow search algorithm-based optimized LightGBM approach for fault diagnosis. *J Amb Intell Human Comput*, early access. <https://doi.org/10.1007/s12652-022-03703-5>
- Faramarzi A, Heidarinejad M, Stephens B, et al., 2020. Equilibrium optimizer: a novel optimization algorithm. *Knowl-Based Syst*, 191:105190. <https://doi.org/10.1016/j.knsys.2019.105190>
- Fister IJr, Yang XS, Fister I, et al., 2013. A brief review of nature-inspired algorithms for optimization. <https://arxiv.org/abs/1307.4186>
- Gai JB, Zhong KY, Du XJ, et al., 2021. Detection of gear fault severity based on parameter-optimized deep belief network using sparrow search algorithm. *Measurement*, 185:110079. <https://doi.org/10.1016/j.measurement.2021.110079>
- Gao GQ, Xin B, 2019. A-STC: auction-based spanning tree coverage algorithm formation planning of cooperative robots. *Front Inform Technol Electron Eng*, 20(1):18-31. <https://doi.org/10.1631/FITEE.1800551>
- Gupta S, Deep K, Mirjalili S, 2020. An efficient equilibrium optimizer with mutation strategy for numerical optimization. *Appl Soft Comput*, 96:106542. <https://doi.org/10.1016/j.asoc.2020.106542>
- Hashim FA, Houssein EH, Mabrouk MS, et al., 2019. Henry gas solubility optimization: a novel physics-based algorithm. *Fut Gener Comput Syst*, 101:646-667. <https://doi.org/10.1016/j.future.2019.07.015>
- Hashim FA, Hussain K, Houssein EH, et al., 2021. Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. *Appl Intell*, 51(3):1531-1551. <https://doi.org/10.1007/s10489-020-01893-z>
- Heidari AA, Mirjalili S, Faris H, et al., 2019. Harris hawks optimization: algorithm and applications. *Fut Gener Comput Syst*, 97:849-872. <https://doi.org/10.1016/j.future.2019.02.028>

- Heidari AA, Aljarah I, Faris H, et al., 2020. An enhanced associative learning-based exploratory whale optimizer for global optimization. *Neur Comput Appl*, 32(9):5185-5211. <https://doi.org/10.1007/s00521-019-04015-0>
- Khishe M, Mosavi MR, 2020. Chimp optimization algorithm. *Expert Syst Appl*, 149:113338. <https://doi.org/10.1016/j.eswa.2020.113338>
- Li CH, Song Y, Wang FY, et al., 2017. A chaotic coverage path planner for the mobile robot based on the Chebyshev map for special missions. *Front Inform Technol Electron Eng*, 18(9):1305-1319. <https://doi.org/10.1631/FITEE.1601253>
- Li SM, Chen HL, Wang MJ, et al., 2020. Slime mould algorithm: a new method for stochastic optimization. *Fut Gener Comput Syst*, 111:300-323. <https://doi.org/10.1016/j.future.2020.03.055>
- Li XJ, Gu JN, Sun XH, et al., 2022. Parameter identification of robot manipulators with unknown payloads using an improved chaotic sparrow search algorithm. *Appl Intell*, 52(9):10341-10351. <https://doi.org/10.1007/s10489-021-02972-5>
- Liang JJ, Qin AK, Suganthan PN, et al., 2006. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans Evol Comput*, 10(3):281-295. <https://doi.org/10.1109/TEVC.2005.857610>
- Liu JC, Wei JH, Heidari AA, et al., 2022. Chaotic simulated annealing multi-verse optimization enhanced kernel extreme learning machine for medical diagnosis. *Comput Biol Med*, 144:105356. <https://doi.org/10.1016/j.compbimed.2022.105356>
- Liu JN, Peng H, Wu ZJ, et al., 2020. Multi-strategy brain storm optimization algorithm with dynamic parameters adjustment. *Appl Intell*, 50(4):1289-1315. <https://doi.org/10.1007/s10489-019-01600-7>
- Long W, Jiao JJ, Liang XM, et al., 2022. A velocity-guided Harris hawks optimizer for function optimization and fault diagnosis of wind turbine. *Artif Intell Rev*, early access. <https://doi.org/10.1007/s10462-022-10233-1>
- Mittal H, Pal R, Kulhari A, et al., 2016. Chaotic Kbest gravitational search algorithm (CKGSA). *Proc 9th Int Conf on Contemporary Computing*, p.1-6. <https://doi.org/10.1109/IC3.2016.7880252>
- Moosavi SHS, Bardsiri VK, 2019. Poor and rich optimization algorithm: a new human-based and multi populations algorithm. *Eng Appl Artif Intell*, 86:165-181. <https://doi.org/10.1016/j.engappai.2019.08.025>
- Nadimi-Shahraki MH, Taghian S, Mirjalili S, 2021. An improved grey wolf optimizer for solving engineering problems. *Expert Syst Appl*, 166:113917. <https://doi.org/10.1016/j.eswa.2020.113917>
- Naik MK, Panda R, Abraham A, 2021. Adaptive opposition slime mould algorithm. *Soft Comput*, 25(22):14297-14313. <https://doi.org/10.1007/s00500-021-06140-2>
- Nama S, Sharma S, Saha AK, et al., 2022. A quantum mutation-based backtracking search algorithm. *Artif Intell Rev*, 55(4):3019-3073. <https://doi.org/10.1007/s10462-021-10078-0>
- Poli R, Kennedy J, Blackwell T, 2007. Particle swarm optimization. *Swarm Intell*, 1(1):33-57. <https://doi.org/10.1007/s11721-007-0002-0>
- Qin AK, Huang VL, Suganthan PN, 2009. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans Evol Comput*, 13(2):398-417. <https://doi.org/10.1109/TEVC.2008.927706>
- Rao RV, Savsani VJ, Vakharia DP, 2011. Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. *Comput-Aided Des*, 43(3):303-315. <https://doi.org/10.1016/j.cad.2010.12.015>
- Rashedi E, Nezamabadi-Pour H, Saryazdi S, 2009. GSA: a gravitational search algorithm. *Inform Sci*, 179(13):2232-2248. <https://doi.org/10.1016/j.ins.2009.03.004>
- Ruan WY, Duan HB, 2020. Multi-UAV obstacle avoidance control via multi-objective social learning pigeon-inspired optimization. *Front Inform Technol Electron Eng*, 21(5):740-748. <https://doi.org/10.1631/FITEE.2000066>
- Simon D, 2008. Biogeography-based optimization. *IEEE Trans Evol Comput*, 12(6):702-713. <https://doi.org/10.1109/TEVC.2008.919004>
- Srinivas M, Patnaik LM, 1994. Genetic algorithms: a survey. *Computer*, 27(6):17-26. <https://doi.org/10.1109/2.294849>
- Storn R, Price K, 1997. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim*, 11(4):341-359. <https://doi.org/10.1023/A:1008202821328>
- Tian ZD, Chen H, 2021. A novel decomposition-ensemble prediction model for ultra-short-term wind speed. *Energy Conv Manag*, 248:114775. <https://doi.org/10.1016/j.enconman.2021.114775>
- Tizhoosh HR, 2005. Opposition-based learning: a new scheme for machine intelligence. *Proc Int Conf on Computational Intelligence for Modelling, Control and Automation and Int Conf on Intelligent Agents, Web Technologies and Internet Commerce*, p.695-701. <https://doi.org/10.1109/CIMCA.2005.1631345>
- Tu JZ, Chen HL, Wang MJ, et al., 2021. The colony predation algorithm. *J Bion Eng*, 18(3):674-710. <https://doi.org/10.1007/s42235-021-0050-y>
- Wang MJ, Chen HL, 2020. Chaotic multi-swarm whale optimizer boosted support vector machine for medical diagnosis. *Appl Soft Comput*, 88:105946. <https://doi.org/10.1016/j.asoc.2019.105946>
- Wang X, Liu J, Hou T, et al., 2021. The SSA-BP-based potential threat prediction for aerial target considering commander emotion. *Def Technol*, 18(11):2097-2106. <https://doi.org/10.1016/j.dt.2021.05.017>
- Wolpert DH, Macready WG, 1997. No free lunch theorems for optimization. *IEEE Trans Evol Comput*, 1(1):67-82. <https://doi.org/10.1109/4235.585893>
- Wu TQ, Yao M, Yang JH, 2016. Dolphin swarm algorithm. *Front Inform Technol Electron Eng*, 17(8):717-729. <https://doi.org/10.1631/FITEE.1500287>
- Xin B, Chen J, Peng ZH, et al., 2010. An adaptive hybrid optimizer based on particle swarm and differential evolution for global optimization. *Sci China Inform Sci*, 53(5):980-989. <https://doi.org/10.1007/s11432-010-0114-9>

- Xue JK, Shen B, 2020. A novel swarm intelligence optimization approach: sparrow search algorithm. *Syst Sci Contr Eng*, 8(1):22-34. <https://doi.org/10.1080/21642583.2019.1708830>
- Yang YT, Chen HL, Heidari AA, et al., 2021. Hunger games search: visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. *Expert Syst Appl*, 177:114864. <https://doi.org/10.1016/j.eswa.2021.114864>
- Yelghi A, Köse C, 2018. A modified firefly algorithm for global minimum optimization. *Appl Soft Comput*, 62:29-44. <https://doi.org/10.1016/j.asoc.2017.10.032>
- Zhang CL, Ding SF, 2021. A stochastic configuration network based on chaotic sparrow search algorithm. *Knowl-Based Syst*, 220:106924. <https://doi.org/10.1016/j.knsys.2021.106924>
- Zhang GH, Wang L, Xing KY, 2021. Dual-space co-evolutionary memetic algorithm for scheduling hybrid differentiation flowshop with limited buffer constraints. *IEEE Trans Syst Man Cybern Syst*, 52(11):6822-6836. <https://doi.org/10.1109/TSMC.2021.3102658>
- Zhang XM, Wang DD, Fu ZH, et al., 2020. Novel biogeography-based optimization algorithm with hybrid migration and global-best Gaussian mutation. *Appl Math Model*, 86:74-91. <https://doi.org/10.1016/j.apm.2020.05.016>
- Zhang XQ, Zhang YY, Ming ZF, 2021. Improved dynamic grey wolf optimizer. *Front Inform Technol Electron Eng*, 22(6):877-890. <https://doi.org/10.1631/FITEE.2000191>
- Zhang Z, He R, Yang K, 2022. A bioinspired path planning approach for mobile robots based on improved sparrow search algorithm. *Adv Manuf*, 10(1):114-130. <https://doi.org/10.1007/s40436-021-00366-x>
- Zhu YL, Yousefi N, 2021. Optimal parameter identification of PEMFC stacks using adaptive sparrow search algorithm. *Int J Hydrogen Energy*, 46(14):9541-9552. <https://doi.org/10.1016/j.ijhydene.2020.12.107>

List of supplementary materials

- Table S1 Description of the CEC2014 test functions
 Table S2 Description of the CEC2019 test functions
 Table S3 Description of the classical benchmark functions
 Table S4 Parameter settings
 Table S5 Results generated by Wilcoxon signed-rank tests on the CEC2014 test functions
 Table S6 Statistical results of the MSSSA and the other algorithms on 30-dimensional CEC2014 benchmark functions
 Table S7 Statistical results of the MSSSA and the other algorithms on 10-dimensional CEC2019 benchmark functions
 Table S8 Scalability results of the MSSSA and SSA in 10 classical functions