

Full length article

MSAO: A multi-strategy boosted snow ablation optimizer for global optimization and real-world engineering applicationsYaning Xiao^{a,*}, Hao Cui^b, Abdelazim G. Hussien^{c,d,e,*}, Fatma A. Hashim^{f,g}^a Center for Control Science and Technology, Southern University of Science and Technology, Shenzhen 518055, China^b College of Mechanical and Electrical Engineering, Northeast Forestry University, Harbin 150040, China^c Department of Computer and Information Science, Linköping University, 58183 Linköping, Sweden^d Faculty of Science, Faiyum University, Faiyum 63514, Egypt^e Applied Science Research Center, Applied Science Private University, Amman 11931, Jordan^f Faculty of Engineering, Helwan University, Egypt^g MEU Research Unit, Middle East University, Amman 11831, Jordan

ARTICLE INFO

Keywords:

Snow Ablation Optimizer
 Good point set initialization
 Greedy selection
 Differential evolution
 Dynamic lens opposition-based learning
 Global optimization

ABSTRACT

Snow Ablation Optimizer (SAO) is a cutting-edge nature-inspired *meta-heuristic* technique that mimics the sublimation and melting processes of snow in its quest for optimal solution to complex problems. While SAO has demonstrated competitive performance in comparison to classical algorithms in early research, it still exhibits certain limitations including low convergence accuracy, a lack of population diversity, and premature convergence, particularly when addressing high-dimensional intricate challenges. To mitigate the above-mentioned adverse factors, this paper introduces a novel variant of SAO with featuring four enhancement strategies collectively referred as MSAO. Firstly, the good point set initialization strategy is employed to generate a uniformly distributed high-quality population, which facilitates the algorithm to enter the appropriate search domain rapidly. Secondly, the greedy selection method is adopted to reserve better candidate solutions for the next iteration, thus striking a robust exploration-exploitation balance. Then, the Differential Evolution (DE) scheme is introduced to expand the search range and enhance the exploitation capability of the algorithm for higher convergence accuracy. Finally, to reduce the risk of falling into local optima, a Dynamic Lens Opposition-Based Learning (DLOBL) strategy is developed to operate on the current optimal solution dimension by dimension. With the blessing of these strategies, the optimization performance of MSAO is comprehensively improved. To comprehensively evaluate the optimization performance of MSAO, a series of numerical optimization experiments are conducted using the IEEE CEC2017 & CEC2022 test sets. In the IEEE CEC2017 experiments, the optimal crossover probability $CR = 0.8$ is determined and the effectiveness of each improvement strategy is ablatively verified. MSAO is compared with the basic SAO, various state-of-the-art optimizers, and CEC2017 champion algorithms in terms of solution accuracy, convergence speed, robustness, and scalability. In the IEEE CEC2022 experiments, MSAO is compared with some recently developed improved algorithms to further validate its superiority. The results demonstrate that MSAO has excellent overall optimization performance, with the smallest Friedman mean rankings of 1.66 and 1.25 on both test suites, respectively. In the majority of test cases, MSAO can provide more accurate and reliable solutions than other competitors. Furthermore, six realistic constrained engineering design challenges and one photovoltaic model parameter estimation issue are employed to demonstrate the practicality of MSAO. Our findings suggest that MSAO has excellent optimization capacity and broad application potential.

1. Introduction

Optimization constitutes a mathematical computing technique aimed at detecting the global optimal solution for a problem under finite

resources or specific constraints [1]. With the advent of the artificial intelligence (AI) era, optimization has gained increasing importance in multiple disciplines and engineering fields, including but not limited to path planning [2], fault diagnosis [3], image segmentation [4], intelligent control [5], pattern recognition [6], chemoinformatics [7], feature

* Corresponding authors at: Department of Computer and Information Science, Linköping University, 58183 Linköping, Sweden (Abdelazim G. Hussien).

E-mail addresses: 12331361@mail.sustech.edu.cn (Y. Xiao), aga08@fayoum.edu.eg (A.G. Hussien).

Nomenclature	
N	Population size
t	Current number of iterations
t_{\max}	Maximum number of iterations
D	Dimension size
\mathbf{X}_i	Position vector of the i -th search agent
\mathbf{Lb}	Lower boundary of the search domain
\mathbf{Ub}	Upper boundary of the search domain
\mathbf{X}_{elite}	A random individual selected from the elite pool
\mathbf{BM}_i	Brownian motion random number set
\mathbf{X}_{best}	Global optimum solution
\mathbf{X}_{mean}	Mean position of the entire population
\mathbf{X}_{second}	Second-best solution in the population
\mathbf{X}_{third}	Third-best solution in the population
\mathbf{X}_c	Centroid location of individuals in the top 50 % of fitness scores
N_1	Total number of leaders
M	Snowmelt ratio
DDF	Degree-day coefficient
T	Average daily temperature
r_1	A random number between 0 and 1
r_2	A random number between 0 and 1
P_a	Sub-population a
P_b	Sub-population b
N_a	Number of individuals in the sub-population P_a
N_b	Number of individuals in the sub-population P_b
\mathbf{V}_i	Position vector of the i -th mutant individual
F	Mutation scale factor
CR	Crossover probability
\tilde{x}	Dynamic lens opposition-based learning point
k	Dynamic learning coefficient
$Mean$	Average fitness value
Std	Standard deviation
$RMSE$	Root mean square error
IEEE	Institute of electrical and electronics engineers
CEC	Congress on evolutionary computation
SAO	Snow Ablation Optimizer
MSAO	Multi-strategy boosted Snow Ablation Optimizer
PV	Photovoltaic
SDM	Single diode model
DE	Differential evolution
DLOBL	Dynamic lens opposition-based learning
MAs	Meta-heuristic algorithms
AOA	Arithmetic Optimization Algorithm
SMA	Slime Mould Algorithm
AO	Aquila Optimizer
ARO	Artificial Rabbits Optimization
GJO	Golden Jackal Optimization
TSA	Tunicate Swarm Algorithm
DO	Dandelion Optimizer
SO	Snake Optimizer
CMA-ES	Covariance Matrix Adaptation Evolution Strategy
LSHADE-SPACMA	LSHADE with Semi-Parameter Adaptation Hybrid with CMA-ES
LSHADE-cnEpSin	Ensemble Sinusoidal Differential Covariance Matrix Adaptation with Euclidean Neighborhood
CSOOAO	Enhanced hybrid Arithmetic Optimization Algorithm
IYDSE	Ameliorated Young's double-slit experiment optimizer
IHAOAVOA	Improved hybrid Aquila Optimizer and African Vultures Optimization Algorithm
ESO	Enhanced Snake Optimizer
EESHHO	Memetic Harris Hawks Optimization
RCLSMAOA	Hybrid Slime Mold and Arithmetic Optimization Algorithm with random center learning and restart mutation

selection [8], task reallocation [9], and parameter identification [10]. Real-world optimization problems typically manifest inherent complexity, characterized by large-scale dimension, non-linearity, and non-convexity. These problems can be conceptually framed as follows within the scope of single-objective optimization [11]:

$$\begin{aligned} & \text{Minimize } f(\mathbf{x}), \mathbf{x} = (x_1, x_2, \dots, x_D) \in \mathbb{R}^n \\ & \text{Subject to } g_i(\mathbf{x}) \geq 0, i = 1, 2, \dots, l \\ & \quad h_i(\mathbf{x}) = 0, i = 1, 2, \dots, m \\ & \quad Lb_i \leq x_i \leq Ub_i, i = 1, 2, \dots, D \end{aligned} \quad (1)$$

where $f(\bullet)$ denotes an objective function on domain Ω which is a subset of the Euclidean space \mathbb{R}^n , indicating the search space. $x_i (i = 1, 2, \dots, D)$ denotes the problem variables. D denotes the quantity of variables. l and m represent the total number of inequality constraints and equality constraints, respectively. The variable x_i is subject to lower and upper boundaries, expressed as Lb_i and Ub_i , respectively. Regarding multi-objective optimization problems, Eq. (1) is adapted as:

$$\text{Minimize } \mathbf{F}(\mathbf{x}), \mathbf{x} = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_o(\mathbf{x})] \quad (2)$$

where o signifies the number of targets. When addressing a multi-objective optimization problem, the ultimate goal is to locate a set of non-dominated solutions, namely the Pareto front [12]. It is essential to clarify that this study does not encompass such scenarios and focuses solely on single-objective optimization problems. In general, traditional mathematical methods and *meta-heuristic algorithms* (MAs) are two commonly available tools for real-world optimization issues [13]. Traditional methods relying on derivative information, such as Newton's method, linear programming, and conjugate gradient method can

deliver optimal solutions for some small-scale optimization assignments [14]. However, when they are employed to solve complex problems with a large number of variables and constraints especially real-world engineering challenges, the merits of these methods will be eclipsed due to the high time consumption and prone to converging towards the local optimum. With the promotion of research, a type of stochastic techniques with self-organization, self-adaptation, and self-learning attributes, known as MAs, has emerged. MAs, hailed as problem-independent algorithms, feature a simple structure, do not require gradient information, and exhibit strong local optimum avoidance capability. As a result, they have received growing favor among scholars in recent years and have been used in various optimization problems.

MAs construct mathematical models to solve optimization problems by simulating a series of natural phenomena such as biological behaviors, physical principles & phenomena, chemical reactions, human habits, and so on. In the literature, different classifications of MAs exist. For instance, in reference [15], the categorization divides them into two primary groups: swarm intelligence algorithms and evolutionary algorithms. Similarly, reference [16] classifies them into three categories: swarm intelligence algorithms, evolutionary algorithms, and physical algorithms. Reference [17], on the other hand, classifies them as either population-based solutions or single-based solution. Overall, there isn't a universally accepted standard for classifying metaheuristic algorithms. However, the most commonly used classification criteria are based on their diverse sources of inspiration. In this paper, MAs have been partitioned into five distinct groups: evolutionary, physical, mathematics-based, swarm-based, and human-based [18]. Table 1 provides a concise overview of MAs. Evolutionary algorithms mainly mimic biological strategies, such as mating, genetic variation, and mutational

Table 1

Overview of metaheuristic algorithms (MAs).

Classification	Algorithm	Abbrev.	Inspiration	Ref.	Year
Evolutionary	Genetic Algorithm	GA	Darwin's evolution theory	[19]	1975
	Differential Evolution	DE	Biological evolutionary mechanisms	[20]	1997
	Biogeography-Based Optimization	BBO	Mathematics of biogeography	[21]	2008
	Quantum-inspired Evolutionary Algorithm	QEA	Quantum physics	[22]	2017
	Liver Cancer Algorithm	LCA	Behavior of liver tumors	[23]	2023
Physics-based	Simulated Annealing	SA	Principle of solid annealing	[24]	1983
	Gravitational Search Algorithm	GSA	Law of gravity	[25]	2009
	Multi-Verse Optimizer	MVO	Theory of multi-verse in physics	[26]	2015
	Thermal Exchange Optimization	TEO	Newton's law of cooling	[28]	2017
	Atom Search Optimization	ASO	Basic molecular dynamics	[27]	2019
	Rime Optimization Algorithm	RIME	Physical phenomenon of rime-ice	[29]	2023
	Kepler Optimization Algorithm	KOA	Kepler's laws of planetary motion	[30]	2023
Mathematics-based	Sine Cosine Algorithm	SCA	Sine and Cosine functions	[31]	2016
	Arithmetic Optimization Algorithm	AOA	Distribution behavior of Arithmetic operators	[32]	2021
	RUNge Kutta optimizer	RUN	Foundations of the Runge Kutta method	[33]	2021
	Weighted Mean of Vectors	INFO	Concept of the weighted mean of vectors	[34]	2022
	Exponential Distribution Optimizer	EDO	Exponential probability distribution model	[35]	2023
	Quadratic Interpolation Optimization	QIO	Generalized quadratic interpolation method	[36]	2023
Population-based	Particle Swarm Optimization	PSO	Predation strategies of birds	[37]	1995
	Artificial Bee Colony	ABC	Foraging strategies of bees	[38]	2007
	Grey Wolf Optimizer	GWO	Leadership hierarchy and foraging strategy	[39]	2014
	Whale Optimization Algorithm	WOA	Social behavior of humpback whales	[40]	2016
	Moth Search Algorithm	MSA	Phototaxis and Lévy flights of moths	[41]	2016
	Harris Hawks Optimization	HHO	Harris' hawks collaborative attack behavior	[42]	2019
	Slime Mould Algorithm	SMA	Oscillation mode of slime mould	[43]	2020
	Tunicate Swarm Algorithm	TSA	Jet propulsion and swarm behaviors of tunicates	[53]	2020
	Aquila Optimizer	AO	Predation behavior of Aquila	[44]	2021
	Colony Predation Algorithm	CPA	Corporate predation of animals	[54]	2021
	Remora Optimization Algorithm	ROA	Parasitic behavior of remora	[55]	2021
	Hunger Games Search	HGS	Hunger-driven activities and behavioral choice of animals	[56]	2021
	Artificial Rabbits Optimization	ARO	Survival behavior of rabbits	[45]	2022
	Golden Jackal Optimization	GJO	Predation strategies of golden jackals	[46]	2022
	Dandelion Optimizer	DO	Long-distance flight of dandelion	[57]	2022
	Snake Optimizer	SO	Mating mechanism of snakes	[58]	2022
	Crayfish Optimization Algorithm	COA	Summering, competition and foraging behaviors of crayfish	[59]	2023
Human-based	Harmony Search	HS	Improvisation of musicians	[47]	2001
	Teaching Learning-Based Optimization	TLBO	Teachers' influence on learners	[48]	2011
	Soccer League Competition	SLC	Competition in soccer leagues	[49]	2014
	Collective Decision Optimization	CSO	Features of human decision-making	[50]	2017
	Coronavirus Herd Immunity Optimizer	CHIO	Concept of herd immunity	[51]	2021

evolution. The search procedure commences with a stochastic population and then continuously iterates to achieve multi-generation evolution. Genetic Algorithm (GA) [19], Differential Evolution (DE) [20], Biogeography-Based Optimization (BBO) [21], Quantum-inspired Evolutionary Algorithm (QEA) [22], and Liver Cancer Algorithm (LCA) [23] are included in this category. Physics-based algorithms simulate physical phenomena and their laws in the universe (e.g., gravity, magnetic fields, mass balance, etc.). Simulated Annealing (SA) [24], inspired by the annealing principle in metallurgy, is a representative physics-based optimization paradigm. Some other popular methods in this branch are Gravitational Search Algorithm (GSA) [25], Multi-Verse Optimizer (MVO) [26], Atom Search Optimization (ASO) [27], Thermal Exchange Optimization (TEO) [28], Rime Optimization Algorithm (RIME) [29], and Kepler Optimization Algorithm (KOA) [30]. The third category is a recently burgeoning branch of MAs, i.e., mathematics-based methods. Mathematics-based algorithms stem from mathematical functions, formulas, and theories, which have shown considerable promise in improving the computational efficiency of optimization techniques. Sine Cosine Algorithm (SCA) [31] is a famous optimizer that extracts ideas from trigonometric functions to explore the solution space and converge to the global optimal solution. Other examples include: Arithmetic Optimization Algorithm (AOA) [32], RUNge

Kutta optimizer (RUN) [33], Weighted Mean of Vectors (INFO) [34], Exponential Distribution Optimizer (EDO) [35], and Quadratic Interpolation Optimization (QIO) [36]. The fourth category pertains to swarm-based methods, which derive from the social behaviors of foraging, breeding, and hunting within organism clusters. A wide variety of such algorithms are developed and relatively mature. One classical instance of them is Particle Swarm Optimization (PSO) [37], which models the foraging nature of bird flocks to search for optimal solutions. Other well-regarded optimizers encompass Artificial Bee Colony (ACO) [38], Grey Wolf Optimizer (GWO) [39], Whale Optimization Algorithm (WOA) [40], Moth Search Algorithm (MSA) [41], Harris Hawks Optimization (HHO) [42], Slime Mould Algorithm (SMA) [43], Aquila Optimizer (AO) [44], Artificial Rabbits Optimization (ARO) [45], and Golden Jackal Optimization (GJO) [46]. The last category relates to human cooperation and human behaviors in a community, known as human-based algorithms. Some commonly used techniques are Harmony Search (HS) [47], Teaching Learning-Based Optimization (TLBO) [48], Soccer League Competition (SLC) [49], Collective Decision Optimization (CDO) [50], and Coronavirus Herd Immunity Optimizer (CHIO) [51]. Despite the diverse sources of inspiration for MAs, exploration and exploitation are two indispensable phases in their basic structure [52]. Exploration is designed to explore the entire problem

domain to identify better potential solutions. In contrast, exploitation emphasizes on local search within previously promising regions to steer the algorithm toward the global optimum. MAs need to strike a proper balance between these two stages to ensure that optimal results are robustly obtained during the optimization process [29].

Since MAs are able to provide reliable optimal or near-optimal solutions to diverse complex NP-hard problems within a reasonable timeframe, they have gained widespread popularity for real-world engineering optimization tasks in different fields. Wang et al. [60] proposed an AOA-VMD method by using Archimedes Optimization Algorithm (AOA) to optimize the mode number and penalty factor of Variational Mode Decomposition (VMD). Through simulation and actual cases, it is demonstrated that AOA-VMD owns the edges of rapid convergence speed and effective fault feature extraction. Zhang et al. [61] presented a hybrid IHGS-ELM model for predicting the shrinkage of molded parts by selective laser sintering based on Improved Hunger Games Search Algorithm (IHGS) and Extreme Learning Machine (ELM). Experimental results show that IHGS-ELM can efficiently determine the optimal process parameters for the Polyethersulfone (PES) material. Abdel-Basset et al. [62] introduced a swarm intelligence optimizer called HSMA_WOA, which combines Slime Mould Algorithm (SMA) and Whale Optimization Algorithm (WOA). HSMA_WOA was applied to the multi-threshold image segmentation for X-ray images, and the results show that it outperforms all comparison approaches in terms of fitness, standard deviation, CPU time, etc. Liang et al. [63] proposed an opposition-based learning and parallel strategies Gorilla Troops Optimizer (OPGTO) for solving the localization problem of nodes in wireless sensor networks. The algorithm is effective in minimizing the localization error based on the time difference of arrival. Nassef et al. [64] integrated the dimensional learning hunting strategy into the Honey Badger Algorithm (HBA) to develop an mHBA method, which has been successfully used to extract the global maximum power point of a triple-junction solar photovoltaic system under partial shading conditions. Ayyarao and Kumar [65] proposed a War Strategy Optimization (WSO) algorithm for identifying unknown parameters in a solar photovoltaic model. Jagatheesan et al. [66] employed Grey Wolf Optimizer (GWO) to tune gain factors of a Proportional-Integral-Derivative (PID) regulator to improve its robustness in load frequency control for interconnected power-generating systems.

In response to the ever-growing optimization challenges, more and more new MAs are successively presented to play pivotal roles in intelligent computing. However, in fact, MAs have almost identical average performance for global optimization, and most of them frequently manifest shortcomings related to low convergence accuracy, sluggish convergence speed, and local optimal stagnation [67]. Hence, beyond designing new MAs, it is likewise a meaningful endeavor to improve the optimization capability of existing MAs. Following an extensive review of the literature, it turned out that researchers mainly achieve this requirement through two means [68]: (1) Add one or more search strategies to the original algorithm. To mitigate the deficiencies of Remora Optimization Algorithm (ROA), such as being prone to local optimum and slow convergence, Jia et al. [69] proposed the MSROA algorithm based on random restart mechanism, information entropy evaluation mechanism, and visual perception mechanism. The proposed method was evaluated on the IEEE CEC2005 benchmark suite and CEC2017 benchmark suite, with the results indicating that MSROA has a stronger optimization capability than ROA. Zheng et al. [70] proposed an improved African Vultures Optimization Algorithm (AVOA) called EAVOA. Three improvements were introduced in EAVOA, including rotating flight mechanism, representative vulture selection mechanism, and selecting accumulation operator. EAVOA was used to address three constrained engineering designs and to train multi-layer perceptron, which proved its excellent competitiveness for continuous optimization. Hu et al. [71] proposed a modified Dandelion Optimizer (DETDO) by injecting differential evolution operators, adaptive Tent chaotic map, and t -distribution perturbation. Comprehensive experimental findings

consistently showcase DETDO's substantial superiority over many advanced optimization methods. To enhance the optimization performance of Snake Optimizer (SO) in solving intricate real-life engineering problems, Yao et al. [72] utilized the dynamic update strategy, mirror imaging learning, Tent-chaos mutation, and Sine-cosine composite perturbation factor to design a new ESO algorithm. In competition with thirteen optimizers, ESO ranks first. Pan et al. [73] put forward a golden eagle optimization algorithm with dual learning strategies (GEO-DLS), where personal example learning was used to improve the optimizer's search ability, and mirror reflection learning promoted the convergence accuracy. The effectiveness of GEO-DLS has been validated in resolving path-planning problems for unmanned aerial vehicles (UAVs) during power inspection. Hashim et al. [74] proposed a modified sea horse optimizer (mSHO), which integrates three search operators: neighborhood-based local search, global non-neighbor-based search, and walk around the existing search in return for more robust exploitation trend. The mSHO have been used to solve ten unconstrained CEC2020 benchmark problems as well as nine constrained real-world engineering tasks. The results demonstrate that mSHO is competitive with other state-of-the-art methods. (2) Hybridize different MAs with the potential embedding of one or more search operators. Zhang et al. [75] suggested an OCRUN algorithm by combining Cuckoo Search (CS) and opposition-based learning with Runge-Kutta Optimization (RUN) to reinforce its exploration phase. The performance of OCRUN was measured on 30 CEC2014 functions, and the findings illustrate the excellent robustness and searchability of OCRUN. Wang et al. [76] combined Harris Hawks Optimization (HHO) and Aquila Optimizer (AO) into a novel hybrid optimization technique, referred to as IHAOHHO. This algorithm also introduced random reverse learning and escaping energy factor for balancing exploitation and exploration. Compared with AO and HHO, experimental results reveal that IHAOHHO has superior convergence accuracy for numerical optimization problems. Yuan et al. [77] developed a jellyfish search-boosted hybrid beluga whale optimizer (HBWO-JS). HBWO-JS has stable dimensional scalability and can provide satisfactory solutions for both function optimization problems and topology optimization problems. Chen et al. [69] devised an innovative approach known as RCLSMAOA, which combined Arithmetic Optimization Algorithm (AOA) and Slime Mould Algorithm (SMA), incorporating restart mutation and stochastic center learning. This method has proved to possess superior exploration and exploitation capabilities than its ontologists. Ghith and Tolba [78] blended arithmetic optimization and gorilla troops optimization to boost the dynamic response characteristics of the PID controller for micro-robotics systems. Qian et al. [79] proposed a hybrid SSACO algorithm by fusing the foraging model of salp swarm algorithm into the ant colony optimizer. In SSACO, the salp foraging behavior effectively enhances the original algorithm's local optimum avoidance ability, so that the convergence accuracy is significantly promoted. SSACO is successfully applied to remote sensing image segmentation, and the results based on peak signal-to-noise ratio, structural similarity index, and feature similarity index proved that this method has unique advantages over similar segmentation algorithms.

Snow Ablation Optimizer (SAO) is a novel physics-inspired metaheuristic method proposed by Deng and Liu through simulating the sublimation and melting behaviors of snow [80]. In SAO, depending on the different transformation forms of snow, e.g., liquid water and steam, search agents toggle between exploration and exploitation modes for converging towards the global optimum. The effectiveness of SAO is thoroughly substantiated by the aspects of statistical analysis, convergence analysis, and stability analysis on the IEEE CEC2017 benchmark suite as well as CEC2020 practical optimization problems. Early investigations have underscored that compared with several classical MAs, SAO shares the merits of a flexible framework, fewer parameter configurations, and good precision. Nonetheless, SAO inevitably suffers from certain shortcomings, thus prompting the choice of this paper to advocate for an improved version of SAO. The primary reasons are as

follows: (1) SAO exhibits inadequate solution accuracy and convergence speed in tackling some non-convex and high-dimensional numerical optimization tasks. The global exploration trend of SAO still has room for further improvement; (2) When confronted with modern complex engineering optimization challenges, SAO is prone to local optimum stagnation due to a great number of non-linear constraints. It is crucial to enhance the local optimum avoidance capacity of SAO; (3) The no-free-lunch (NFL) theorem [81] implies that no MA is guaranteed to be feasible for all optimization applications. Thus, appropriate strategies can be leveraged to contribute to a more effective and promising performance of SAO for real-world engineering optimization issues.

Since SAO has just been proposed for a short period of time, research on its application and improvement is still scarce. In the light of the above motivation, this paper proposes a multi-strategy boosted snow ablation optimizer (MSAO) for global optimization. The proposed MSAO incorporates four enhancement techniques to synergistically surmount the shortcomings of slow convergence, poor population diversity, and insufficient balance between exploitation & exploration. First, the good point set strategy is integrated into the initialization phase to generate a uniformly distributed population. Then, the greedy selection technology is added to the exploration and exploitation sessions to retain better individuals for next iterations, which facilitates information interaction among diverse individuals and establishes a good exploration-exploitation tradeoff. Furthermore, three Differential Evolution (DE) operators: crossover, mutation, and selection are implemented to update the candidate positions of all individuals to strengthen the local exploitation and expand the search region in return for higher solution accuracy. Finally, to maximize the likelihood of SAO to escape from local optima, a novel Dynamic Lens Opposition-Based Learning (DLOBL) strategy is proposed to perturb the current optimal solution from the opposite direction at the later stage.

To comprehensively evaluate the optimization performance of MSAO, the sensitivity analysis is first conducted to select the most suitable crossover probability CR . Then, the qualitative analysis, statistical analysis, convergence analysis, stability analysis, and scalability analysis under varying dimensions ($D = 30, 50, 100$) are performed sequentially for MSAO using 29 IEEE CEC2017 benchmark functions. Further, the reliability and competitiveness of MSAO are highlighted on 12 IEEE CEC2022 benchmark mathematical functions. The results of the suggested optimizer are compared with various state-of-the-art MAs, including nine advanced optimizers, three CEC competition regulars, and six modified variant algorithms. In addition, MSAO is applied to address six constrained engineering industrial design cases and one photovoltaic (PV) model parameter estimation issue to verify its practicability in real-world scenarios. The main contributions are summarized as follows:

- A more robust variant of SAO with better overall optimization performance named MSAO is proposed, which is implemented through four improvement steps:
 - (a) Good point set initialization strategy provides a uniformly distributed high-quality population,
 - (b) Greedy selection strategy establishes a good balance between exploration and exploitation,
 - (c) Differential Evolution enhances the local exploitation capability of the original algorithm and expands the search depth, and
 - (d) Dynamic Lens Opposition-Based Learning is designed to escape from local optimal stagnation;
- 29 CEC2017 and 12 CEC2022 test functions are utilized to verify the strength of MSAO, and the obtained results are contrasted with different state-of-the-art methods and high-performance modified variant algorithms.
- The prospect of MSAO in handling complex real-world optimization challenges is validated on six engineering design cases and one model parameter estimation problem.

- The Wilcoxon rank-sum test and Friedman ranking test demonstrate that MSAO can surpass other rival algorithms regarding solution accuracy, convergence rate, and robustness.

The remainder of this study is organized as follows: Section 2 introduces a detailed overview of the concepts and mathematical models underlying SAO, good point set initialization strategy, greedy selection strategy, DE strategy, and DLOBL strategy. In Section 3, a multi-strategy enhanced snow ablation optimization algorithm known as MSAO is developed. In Section 4, the optimization performance of MSAO on the IEEE CEC2017 as well as CEC2022 benchmark suites is evaluated. Section 5 validates the effectiveness of MSAO in practical applications through some constrained engineering design and PV model parameter estimation examples. In Section 6, the experimental results are summarized and discussed. Lastly, Section 7 encapsulates the conclusion along with potential avenues for future research.

2. Preliminaries

This section describes five core themes related to our work. First, the mathematical model of the original SAO is reviewed in subsection 2.1. Then, four modification operators used to enhance the optimization performance of SAO, namely, good point set initialization, greedy selection, differential evolution, and dynamic lens opposition-based learning, are briefly presented sequentially in subsections 2.2 to 2.5.

2.1. Snow ablation optimizer

SAO is a biomimetic optimization algorithm first introduced by Deng and Liu [80] in 2023. As depicted in Fig. 1, within the realm of physics, snow undergoes two processes resulting in its transformation into liquid water and steam. These processes are specifically identified as melting and sublimation. Concurrently, it is noteworthy that the liquid water originating from the snowmelt can further be converted into steam by evaporation. Drawing upon this prior knowledge, SAO constructs four components to implement the search for optimal solutions, namely the initialization phase, exploration phase, exploitation phase, and dual-population mechanism.

2.1.1. Initialization phase

As with other MAs, the iteration procedure of SAO commences with a randomly generated population. Assuming that the dimensional size of the optimization problem is D , the lower and upper boundaries of the search domain are \mathbf{Lb} and \mathbf{Ub} , respectively, and the number of search agents is N , then the initial position of the entire population can be modeled as a matrix with N rows and D columns presented in Eq. (3).

$$\mathbf{X} = \mathbf{Lb} + \text{rand} \times (\mathbf{Ub} - \mathbf{Lb}) \\ = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,D-1} & x_{1,D} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,D-1} & x_{2,D} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N-1,1} & x_{N-1,2} & \cdots & x_{N-1,D-1} & x_{N-1,D} \\ x_{N,1} & x_{N,2} & \cdots & x_{N,D-1} & x_{N,D} \end{bmatrix}_{N \times D} \quad (3)$$

where rand indicates a random number in $[0, 1]$. Usually, the position vector of the i -th search agent is expressed as: $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,j}, \dots, x_{i,D}), i = 1, 2, \dots, N, j = 1, 2, \dots, D$.

2.1.2. Exploration phase

When snow or liquid water transforms into steam, the search agent exhibits a highly decentralized trait as a result of irregular motion. In the exploration phase, Brownian motion is used to characterize this phenomenon. Brownian motion is a random process widely employed in simulating the foraging behavior of animals and the erratic movement of

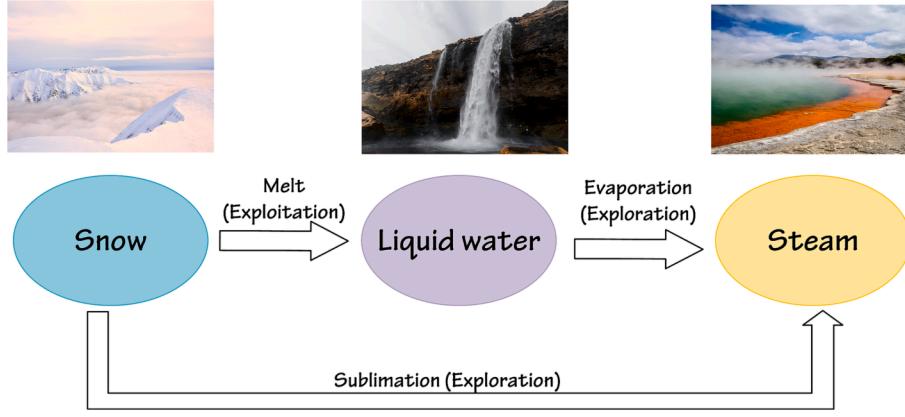


Fig. 1. Inspiration source of SAO.

particles. The step length of standard Brownian motion can be obtained from the probability density function of the normal distribution with mean 0 and variance 1, calculated as follows:

$$f_{BM}(x; 0, 1) = \frac{1}{\sqrt{2\pi}} \times \exp\left(-\frac{x^2}{2}\right) \quad (4)$$

The Brownian motion has a dynamic and homogeneous step size, which assures as many promising areas within the search space as possible to be explored. As a result, it can effectively depict the scenario of steam diffusion. Each search agent \mathbf{X}_i in the exploration iteration can update its current position using the following formula:

$$\begin{aligned} \mathbf{X}_i(t+1) = & \mathbf{X}_{elite}(t) + \mathbf{BM}_i(t) \otimes (r_1 \times (\mathbf{X}_{best}(t) - \mathbf{X}_i(t)) + (1 - r_1) \\ & \times (\mathbf{X}_{mean}(t) - \mathbf{X}_i(t))) \end{aligned} \quad (5)$$

In Eq. (5), $\mathbf{X}_i(t+1)$ represents the location of the i -th individual in the $t+1$ iteration, $\mathbf{BM}_i(t)$ is a set of random numbers symbolizing the Brownian motion, \otimes is the dot product operator, r_1 is a value randomly generated between 0 and 1, and $\mathbf{X}_{best}(t)$ is the optimal solution obtained so far. Besides, $\mathbf{X}_{mean}(t)$ indicates the current mean position of the overall population, and $\mathbf{X}_{elite}(t)$ stands for a random individual selected from the elite pool, which are calculated as follows:

$$\mathbf{X}_{mean}(t) = \frac{1}{N} \sum_{i=1}^N \mathbf{X}_i(t) \quad (6)$$

$$\mathbf{X}_{elite}(t) \in [\mathbf{X}_{best}(t), \mathbf{X}_{second}(t), \mathbf{X}_{third}(t), \mathbf{X}_c(t)] \quad (7)$$

$$\mathbf{X}_c(t) = \frac{1}{N_1} \sum_{i=1}^{N_1} \mathbf{X}_i(t) \quad (8)$$

where $\mathbf{X}_{second}(t)$ and $\mathbf{X}_{third}(t)$ denote the second- and third-best search agents in the current swarm, respectively. $\mathbf{X}_c(t)$ signifies the centroid location of individuals whose fitness scores ranks in the top 50 %, and these individuals are also called leaders. N_1 is the total number of leaders. In this study, $N_1 = 0.5 * N$. The cross terms $r_1 \times (\mathbf{X}_{best}(t) - \mathbf{X}_i(t))$ and $(1 - r_1) \times (\mathbf{X}_{mean}(t) - \mathbf{X}_i(t))$ in the two-dimensional search space are visually presented in Fig. 2. The variable r_1 governs the movements towards the optimal individual obtained so far and leader centroid position. The combination of these two cross terms is primarily designed to capture the interactions among individuals.

2.1.3. Exploitation phase

When snow undergoes conversion into liquid water through the behavior of melting, it is encouraged for search agents to focus on exploiting high-quality solutions around the current optimal solution rather than expanding in the search domain with high-decentralized characteristics. In the exploitation phase, the snowmelt process is modeled by means of a classical degree-day method with the following mathematical expression:

$$M(t) = DDF(t) \times T(t) = \left(0.35 + 0.25 \times \frac{e^{\frac{-t}{T_{max}}} - 1}{e - 1}\right) \times e^{\frac{-t}{T_{max}}} \quad (9)$$

where $M(t)$ indicates the snowmelt ratio, $T(t)$ denotes the average daily

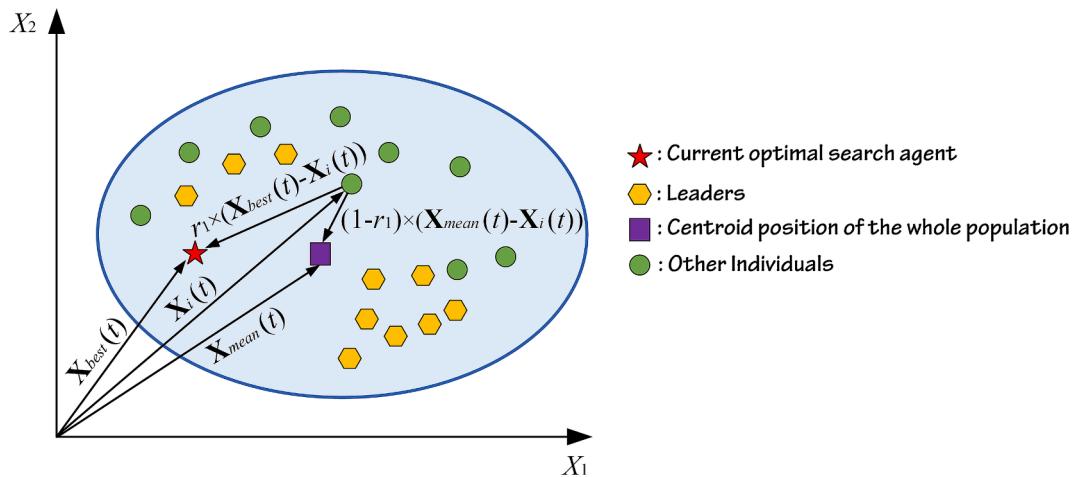


Fig. 2. Schematic diagram of cross terms in two-dimensional space.

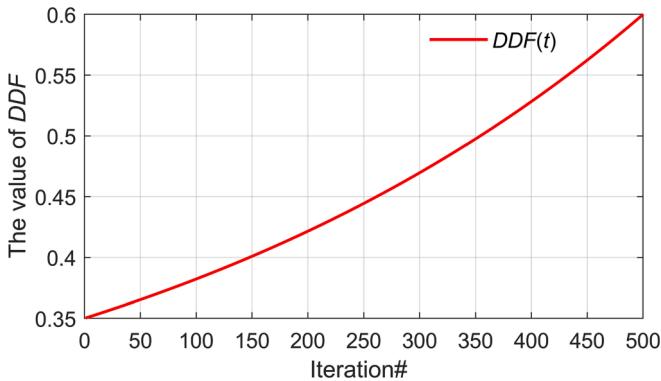


Fig. 3. Trend curve of DDF over iterations.

temperature, t and t_{\max} are the current and maximum iterations, respectively, and $DDF(t)$ refers to the degree-day coefficient with a range from 0.35 to 0.6. Fig. 3 illustrates the trend of DDF over the iterations. Then, the position update equation for this stage is shown below:

$$\begin{aligned} \mathbf{X}_i(t+1) = & M(t) \times \mathbf{X}_{best}(t) + \mathbf{B} \mathbf{M}_i(t) \otimes (r_2 \times (\mathbf{X}_{best}(t) - \mathbf{X}_i(t)) + (1 - r_2) \\ & \times (\mathbf{X}_{mean}(t) - \mathbf{X}_i(t))) \end{aligned} \quad (10)$$

where r_2 is a value randomly generated between 0 and 1.

2.1.4. Dual-population mechanism

To achieve a trade-off between exploration and exploitation in SAO, this part introduces the dual-population mechanism. As described earlier, liquid water derived from snow can also be converted into steam for the exploration session. This implies that as the number of iterations increases, search agents become more inclined to perform irregular movements with high-decentralized features to explore the search space. Thus, in the initial iteration, the entire population P is stochastically split into two equal-sized sub-populations: P_a and P_b , where P_a is responsible for exploration whereas P_b is used for exploitation. The sizes of P , P_a , and P_b correspond to N , N_a , and N_b , respectively. In the succeeding iterations, the amount of N_a gradually decreases and the amount of N_b increases accordingly. The mathematical representation is as follows:

$$N_a = N_a + 1, N_b = N_b - 1, \text{ if } N_a < N \quad (11)$$

Algorithm 1 shows the pseudo-code of SAO.

Algorithm 1. (Snow ablation optimizer (SAO))

Input: Population size N , current iteration t , maximum iterations t_{\max} , dimension size D , $N_a = N_b = \frac{N}{2}$

Output: Global optimal solution \mathbf{X}_{best}

1. Randomly initialize the position of the population $\mathbf{X}_i (i = 1, 2, \dots, N)$ using Eq. (3)
2. Compute the fitness of each individual \mathbf{X}_i
3. Record the current best solution \mathbf{X}_{best}
4. Construct the elite pool using Eqs. (7) and (8)
5. **While** $t \leq t_{\max}$ **do**
6. Calculate the snowmelt rate using Eq. (9)
7. Randomly split the entire population into two sub-populations: P_a and P_b
8. **For** each individual \mathbf{X}_i in P_a ($i = 1, 2, \dots, N_a$) **do**
9. Update the i -th individual's current position using Eq. (5)
10. **End For**
11. **If** $N_a < N$ **then**
12. $N_a = N_a + 1, N_b = N_b - 1$
13. **End If**
14. **For** each individual \mathbf{X}_i in P_b ($i = 1, 2, \dots, N_b$) **do**
15. Update the i -th individual's current position using Eq. (10)
16. **End For**
17. Compute the fitness of each individual \mathbf{X}_i

(continued on next column)

(continued)

-
17. Update the current best solution \mathbf{X}_{best}
 18. Update the elite pool
 19. $t = t + 1$
 20. **End While**
-

2.2. Good point set initialization strategy

The initial distribution positions of the swarm significantly influence both the global convergence rate and the quality of optimal solutions for MAs. A more diverse and uniform population distribution can enhance the algorithm's optimization performance [82]. In SAO, the initial position of each individual is generated by Eq. (3) using random numbers, constituting a random initialization mode. Unfortunately, this method cannot guarantee population diversity and often places the initial point far from the optimal solution, thus affecting the convergence accuracy and search efficiency. As an experimental method aimed at reducing the number of experiments, the good point set theory was first proposed by Hua and Wang [83]. As per reference [84], the point location generated via the good point set method is more uniformly distributed and traverses the whole solution domain than the random initialization. In addition, the deviation order of the good point set sequence is only related to the size of the solution set, but not to the dimensionality of the solution space, which provides theoretical support for solving complex high-dimensional problems. Consequently, this paper adopts the good point set theory to initialize the positions of all search agents, thereby strengthening diversity. The basic principle is elaborated as follows:

Let G_s be the unit cube in s -dimensional Euclidean space and $r \in G_s$. If $\mathbf{P}_n(k) = \left\{ (r_1^{(n)} \times k, \dots, r_i^{(n)} \times k, \dots, r_s^{(n)} \times k), 1 \leq k \leq n \right\}$ and its deviation satisfies $\phi(n) = C(r, \varepsilon)n^{\varepsilon-1}$, where ε denotes an arbitrary positive value and $C(r, \varepsilon)$ indicates a constant associated with r and ε . Then $\mathbf{P}_n(k)$ is defined as the good point set and r denotes a good point.

Take $r_k = \{2\cos(2\pi k/p), 1 \leq k \leq s\}$ where p denotes the smallest prime number meeting $s \leq (p-3)/2$, $\{r_i^{(n)} \times k\}$ represents the decimal part of $r_i^{(n)} \times k$.

If using the good point set technique to generate n points, the deviation is $O(n^{\varepsilon-1})$, which only relates to n and is much smaller than that of the random method. To more visually compare the distributions of random initialization and good point set initialization, let the population size be 100 and search domain be $[-100, 100]$, Fig. 4 illustrates their distributions in two-dimensional space. It can be seen that under the same condition, the distribution of the random population is cluttered with many overlapping individuals, whereas the population initialized by the latter is more uniformly distributed over the search space with no overlapping individuals. Hence, the good point set initialization strategy can effectively promote population diversity as well as improve the global search performance of SAO to some extent.

2.3. Greedy selection strategy

Based on Equations (5) and (10), the basic SAO will directly replace the previous generation solution $\mathbf{X}_i(t)$ with the new candidate solution $\mathbf{X}_i(t+1)$ after performing position updating formulas. However, the fitness value of the updated position may not necessarily surpass that of the original position, potentially hampering the algorithm's convergence to the global optimum. In some cases, this may lead to getting stuck in local optima or even divergence [85]. The greedy selection strategy originates from the idea of "survival of the fittest", which compares the fitness values between the newly generated position and the original position, and then the superior positions in each generation are reserved for the next iteration, meanwhile the worse ones are discarded. This selection operation is formulated as:

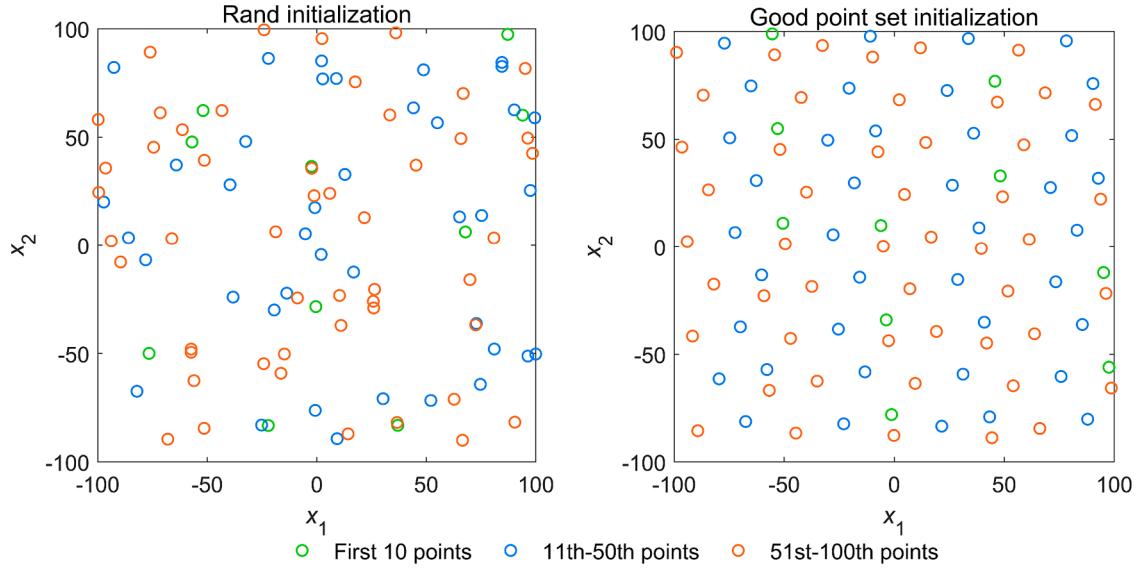


Fig. 4. Comparison of the distribution between random initialization and good point set initialization.

$$\mathbf{X}_i(t+1) = \begin{cases} \mathbf{X}_i(t), & \text{if } f(\mathbf{X}_i(t)) < f(\mathbf{X}_i(t+1)) \\ \mathbf{X}_i(t+1), & \text{otherwise} \end{cases} \quad (12)$$

where $f(\bullet)$ represents the objective function. By integrating the greedy selection strategy into MSAO, its search capability can be more robust because each retained outstanding individual will share the solution space information with other individuals in the subsequent search process. Moreover, it is beneficial for balancing exploration and exploitation trends to drive MSAO to yield higher-quality solutions.

2.4. Differential evolution strategy

Differential Evolution (DE) is a stochastic search method that has shown its effectiveness in addressing a broad spectrum of optimization problems from different practical application tasks [20]. In DE, each search agent is called a target vector. In the initialization period, the algorithm randomly produces all vectors within the search domain. Then, DE runs mutation, crossover, and selection operators to evolve individuals until the termination criterion is satisfied. The mutation operator perturbs a specified set of target vectors to create a mutant vector. The crossover operator combines this mutant vector with its corresponding target vector to form a trial vector. Finally, through evaluating the function values of these trial and target vectors, the selection operator determines which one is superior and retains it in the next generation. [86]. Since mutation, crossover, and selection operators can enforce population diversity to avoid premature convergence, DE has been hybridized with many MAs to improve their exploitation ability and convergence accuracy [87,88]. In this paper, the evolutionary process of DE is embedded into the foundational framework of MSAO to boost the optimization performance. The detailed steps are outlined below:

- (1) Mutation operation (DE/rand/1). Three distinct individuals are randomly chosen from the population, and the gap between two of them is scaled, then combined with the position of the third to create a novel mutant individual. In mathematical terms, this operation is depicted as:

$$\mathbf{V}_i(t) = \mathbf{X}_{R1}(t) + F \times (\mathbf{X}_{R2}(t) - \mathbf{X}_{R3}(t)), R1 \neq R2 \neq R3 \quad (13)$$

where $\mathbf{V}_i(t)$ represents the variant individual, $R1$, $R2$, and $R3$ are three random indexes $\in \{1, 2, \dots, N\}$. F indicates the mutation scale factor, $F =$

$$F_0 \times 2e^{\frac{-t}{t_{\max}}}, \text{ and } F_0 \text{ is equal to 0.5.}$$

- (2) Crossover operation. Once the mutation is complete, the crossover operation creates a trial vector by blending the original target vector and the mutation vector in binary form, using a specified crossover probability.

$$U_{i,j}(t) = \begin{cases} V_{i,j}(t), & \text{if } rand(0, 1) \leq CR \text{ or } j = j_{rand} \\ X_{i,j}(t), & \text{otherwise} \end{cases} \quad (14)$$

where $CR \in [0, 1]$ denotes the crossover probability. In this study, $CR = 0.8$ and how to reach this value will be proven in Section 4. $X_{i,j}(t)$ indicates the i -th search agent's contribution in the j -th dimension. j_{rand} represents a random integer $\in \{1, 2, \dots, D\}$, which assures that $\mathbf{U}_i(t)$ gets at least one parameter from $\mathbf{V}_i(t)$.

- (3) Selection operation. Here as well, the greedy criterion is adopted to assess whether the corresponding target vector $\mathbf{X}_i(t)$ should be substituted with the trial vector $\mathbf{U}_i(t)$. In case of a minimization problem, the selection scheme is defined in Eq. (15).

$$\mathbf{X}_i(t+1) \leftarrow \begin{cases} \mathbf{U}_i(t), & \text{if } f(\mathbf{U}_i(t)) \leq f(\mathbf{X}_i(t)) \\ \mathbf{X}_i(t), & \text{otherwise} \end{cases} \quad (15)$$

If $\mathbf{U}_i(t)$ attains a superior fitness value compared to $\mathbf{X}_i(t)$, it remains as $\mathbf{X}_i(t+1)$ for the subsequent iteration. Otherwise, the existing target vector $\mathbf{X}_i(t)$ is reserved. Fig. 5 illustrates the principle of mutation process and crossover process. By integrating MSAO with the DE strategy, the search scope and population diversity are further increased, thus avoiding local optimal stagnation and enhancing the convergence precision.

2.5. Dynamic lens opposition-based learning strategy

On the basis of Equations (5) and (10) in the conventional SAO, a new candidate solution is generated through guiding the current individual to the global optimal point (\mathbf{X}_{best}). As the optimization process progresses, most of individuals in the colony tend to cluster near the perceived current optimum. Hence, SAO is susceptible to premature convergence. The paramount research priority in the enhancement of SAO is focused on ameliorating its capacity to circumvent local optima. To reinforce the global exploration of MAs, one popular method in the existing literature is Opposition-Based Learning (OBL) [89]. OBL is

rooted in the simultaneous computation of objective values for the current individual and its inverse solution, with the aim of unveiling a more favorable optimal solution for the optimization objective. Nguyen et al. [90] introduced the OBL mechanism into slime mould algorithm to avoid the local optima and provide better optimization performance for optimal operation of terraced hydropower plants. Zhao et al. [91] presented a COLMA method, utilizing OBL to augment the exploration capabilities of the mayfly optimizer.

Lens opposition-based learning (LOBL) is an innovative variation of OBL that simulates the phenomenon of convex lens imaging in optical laws. Specifically, when an object is placed at twice the focal length of a convex lens, an inverted and reduced real image will be shaped on the opposite side of the lens. In the two-dimensional space of Fig. 6, the point O indicates the midpoint of the search interval $[lb, ub]$ and the y -axis is conceptualized as a convex lens. Assumption: the projection of a search individual with height h in the image region on the x -axis is labeled as x (blue point), which lies twice the focal distance away from the lens. After lens imaging, a real image of height \tilde{h} is produced, with its projection on the x -axis represented as \tilde{x} (green point). Following the lens imaging principle, we can derive the geometric equation:

$$\frac{(lb + ub)/2 - x}{\tilde{x} - (lb + ub)/2} = \frac{h}{\tilde{h}} \quad (16)$$

Let $k = h/\tilde{h}$, then Eq. (16) is converted into:

$$\tilde{x} = \frac{lb + ub}{2} + \frac{lb + ub}{2k} - \frac{x}{k} \quad (17)$$

When k equals 1, Eq. (17) can be transformed into the standard form of OBL as follows:

$$\tilde{x} = lb + ub - x \quad (18)$$

This suggests that OBL is merely a particular instance of LOBL, which not only inherits the advantages of OBL but can also promote solution diversity and the likelihood of escaping local optima by tuning the value of k . Extending Eq. (17) to the D -dimensional space is expressed as follows:

$$\tilde{x}_{ij} = \frac{lb_j + ub_j}{2} + \frac{lb_j + ub_j}{2k} - \frac{x_{ij}}{k} \quad (19)$$

where \tilde{x}_{ij} represents the opposite solution of the i -th individual in the j -th dimension. lb_j and ub_j denote the upper and lower boundaries in the j -th dimension, respectively.

In basic LOBL, a constant value is assigned to k , which limits its ability to adapt and create diverse solutions throughout the iterations [92]. Typically, the algorithm emphasizes comprehensive exploration of the search space in the initial iterations to discover hopeful areas with optimal solutions. At this moment, the value of k can be set slightly

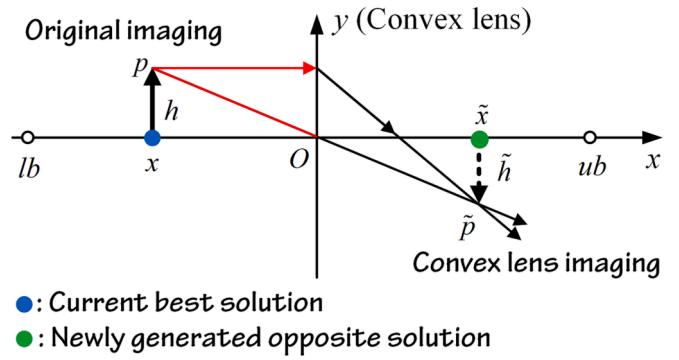


Fig. 6. Lens opposition-based learning.

larger to promote broader search breadth and population diversity. In the latter phases, a smaller k -value can be used to enhance the algorithm's local search effectiveness for a more precise optimal solution. Therefore, this paper proposes a nonlinear dynamic decreasing mechanism for adjusting the value of k as follows:

$$k = 10^4 \times \left[1 - \left(\frac{t}{t_{\max}} \right)^2 \right] + 1 \quad (20)$$

where t denotes the current iteration and t_{\max} indicates the maximum iteration count. Fig. 7 depicts the trend of k . Once all fundamental procedures have been executed, the proposed Dynamic Lens Opposition-Based Learning (DLOBL) mechanism is applied for progressively adjusting the current optimal solution (\mathbf{X}_{best}) dimension by dimension, so as to make it closer to the theoretical optimal solution and accelerate the convergence.

3. Proposed MSAO algorithm

This section presents a detailed framework of the proposed multi-strategy boosted snow ablation optimizer (MSAO) and analyzes its computational complexity.

3.1. Detailed implementation of MSAO

To overcome the limitations of SAO, including poor solution accuracy, sluggish convergence speed, and local optimum stagnation when confronting intricate optimization challenges, this study embeds the good point set initialization strategy, greedy selection strategy, DE strategy, and DLOBL strategy into SAO, thus presenting a multi-strategy boosted variant of SAO named MSAO. First, the good point set

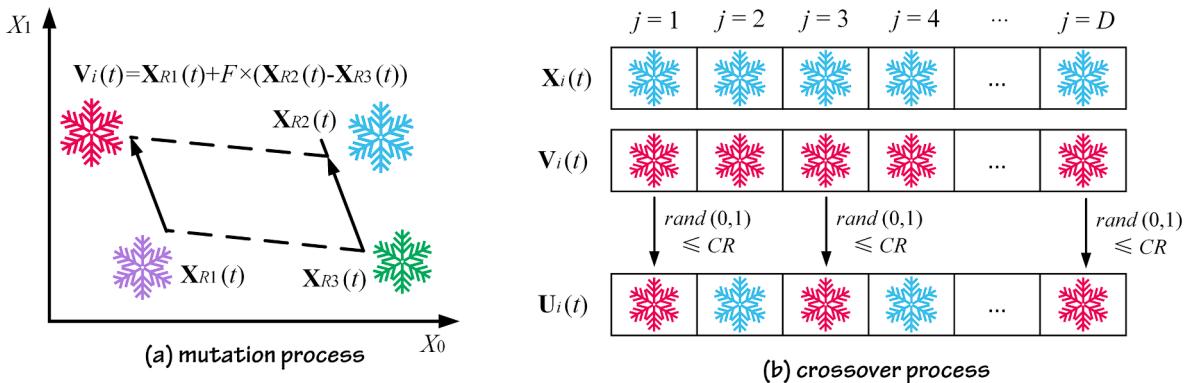


Fig. 5. Schematic diagram of mutation process and crossover process.

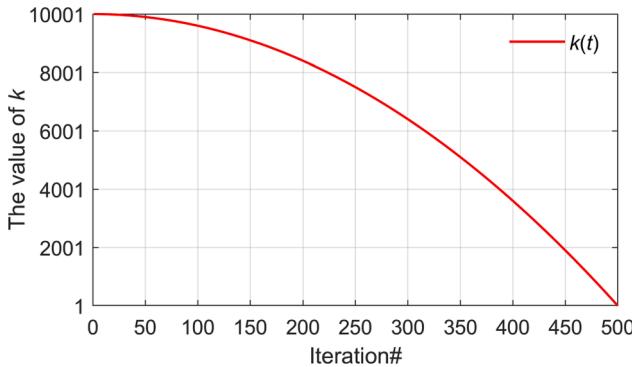


Fig. 7. Trends of the dynamic parameter. k

initialization strategy is introduced to generate a uniformly distributed high-quality population for global search. Then, the greedy selection strategy is implemented during both exploration and exploitation sessions to enhance the robustness and establish an adequate exploration-exploitation balance. Next, the DE mechanism is employed to strengthen the exploitation competence to obtain a superior convergence accuracy. Finally, DLOBL is developed to update the current optimal solution to avoid local stagnation. The detailed steps of MSAO are presented below.

Step 1: Initialize the parameters: population size N , maximum iteration count t_{\max} , variable dimension D , search range $[\mathbf{Lb}, \mathbf{Ub}]$, and crossover probability CR . Set the current iteration $t = 1$;

Step 2: Generate the location of each individual $\mathbf{X}_i (i = 1, 2, \dots, N)$ using the good point set initialization strategy;

Step 3: Assess the fitness value of each search agent, record the optimal solution \mathbf{X}_{best} , and then construct the elite pool using Equations (7) and (8);

Step 4: While $t \leq t_{\max}$, calculate the snowmelt rate M and parameter k using Equation (9) and Equation (20), respectively;

Step 5: Randomly divide the entire population into P_a and P_b . Update the location of each individual $\mathbf{X}_i (i = 1, 2, \dots, N_a)$ in P_a using Eq. (5). Compare the fitness between the newly generated solution and the original solution using greedy selection strategy-Eq. (12) and retain the better one. If N_a is less than N , $N_a = N_a + 1, N_b = N_b - 1$. Update the location of each individual $\mathbf{X}_i (i = 1, 2, \dots, N_b)$ in P_b using Eq. (10). Compare the fitness between the newly generated solution and the original solution using greedy selection strategy-Eq. (12) and retain the better one;

Step 6: Update the location of the population $\mathbf{X}_i (i = 1, 2, \dots, N)$ using the mutation, crossover, and selection operators in turn, i.e., Equations (13)-(15);

Step 7: Re-calculate the fitness of each individual and update the optimal solution \mathbf{X}_{best} ;

Step 8: Perform the DLOBL strategy to update the current optimal solution \mathbf{X}_{best} dimension by dimension using Eq. (19). Then, update the elite pool;

Step 9: Set $t = t + 1$, if $t \leq t_{\max}$, proceed to **Step 4**. if not, output the global optimal solution \mathbf{X}_{best} .

The flowchart of the suggested MSAO is presented in Fig. 8. Algorithm 2 outlines the pseudo-code for MSAO.

Algorithm 2. (Multi-strategy boosted snow ablation optimizer (MSAO))

Input: Population size N , current iteration t , maximum iterations t_{\max} , dimension size D , crossover probability CR , $N_a = N_b = \frac{N}{2}$ **Output:** Global optimal solution \mathbf{X}_{best}

1. Execute good point set initialization to generate the location of each individual $\mathbf{X}_i (i = 1, 2, \dots, N)$
2. Compute the fitness of each individual \mathbf{X}_i
3. Record the current best solution \mathbf{X}_{best}

(continued on next column)

(continued)

4. Construct the elite pool using Eqs. (7) and (8)
5. **While** $t \leq t_{\max}$ **do**
6. Calculate the snowmelt rate using Eq. (9)
7. Calculate the dynamic parameter k using Eq. (20)
8. Randomly split the whole population into two sub-populations: P_a and P_b
9. **For** each individual \mathbf{X}_i in P_a ($i = 1, 2, \dots, N_a$) **do**
10. Update the i -th individual's current position using Eq. (5)
11. Perform the greedy selection to retain the better candidate solution according to Eq. (12)
12. **End For**
13. **If** $N_a < N$ **then**
14. $N_a = N_a + 1, N_b = N_b - 1$
15. **End If**
16. **For** each individual \mathbf{X}_i in P_b ($i = 1, 2, \dots, N_b$) **do**
17. Update the i -th individual's current position using Eq. (10)
18. Perform the greedy selection to retain the better candidate solution according to Eq. (12)
19. **End For**
20. Perform the DE mechanism to update the position of each individual \mathbf{X}_i according to Eqs. (13)-(15)
21. Compute the fitness of each individual \mathbf{X}_i
22. Update the current best solution \mathbf{X}_{best}
23. Perform the DLOBL strategy to update the current optimal solution \mathbf{X}_{best} using Eq. (19)
24. Update the elite pool
25. $t = t + 1$
26. **End While**

3.2. Computational complexity of MSAO

The computational complexity of MSAO hinges on four primary elements: population initialization, position updating, fitness computation, and fitness sorting. In the initialization phase, the computational complexity required to generate the initial position of each individual using the good point set strategy is $O(N \times D)$, where N denotes the number of search agents and D is the dimension. Then, in each iteration, the fitness values of all individuals need to be calculated and sorted. This costs the computational complexity of $O(N \times t_{\max}) + O(N \times \log N \times t_{\max})$, where t_{\max} is the maximum iteration. The computational complexity used to update the locations of all candidate solutions in the exploration and exploitation phases is $O(N \times D \times t_{\max})$. Furthermore, this paper introduces greedy selection strategy, DE strategy, and DLOBL strategy into MSAO, and the computational complexity of these components is: $O(N \times t_{\max}) + O(N \times D \times t_{\max}) + O(D \times t_{\max})$. Thus, the cumulative computational complexity of MSAO is $O(N \times D + T \times (N \times (2 + 2D + \log N) + D))$.

4. Numerical optimization experiments

To comprehensively assess the optimization performance of our suggested MSAO, this section introduces the challenging IEEE CEC2017 and CEC2022 test sets for experimental sequences. MSAO is compared with various well-regarded MAs. For all algorithms, a consistent population size of $N = 30$ and a maximum iteration count of $t_{\max} = 500$ are maintained for fairness in comparisons. Each algorithm independently operates 30 times on each considered benchmark function. The complete experiments are conducted using MATLAB R2020b software on an Intel (R) Core (TM) i5 10300H CPU at 2.50 GHz with 16 GB of RAM and a 64-bit Windows 10 operating system. The best results obtained are emphasized in **boldface**.

4.1. Selected benchmark suites and performance metrics

MSAO undergoes initial testing on 29 benchmark problems spanning four different categories: unimodal functions ($F_1 \sim F_3$), multimodal functions ($F_4 \sim F_{10}$), hybrid functions ($F_{11} \sim F_{20}$), and composition functions ($F_{21} \sim F_{30}$), which together constitute the IEEE CEC2017 test

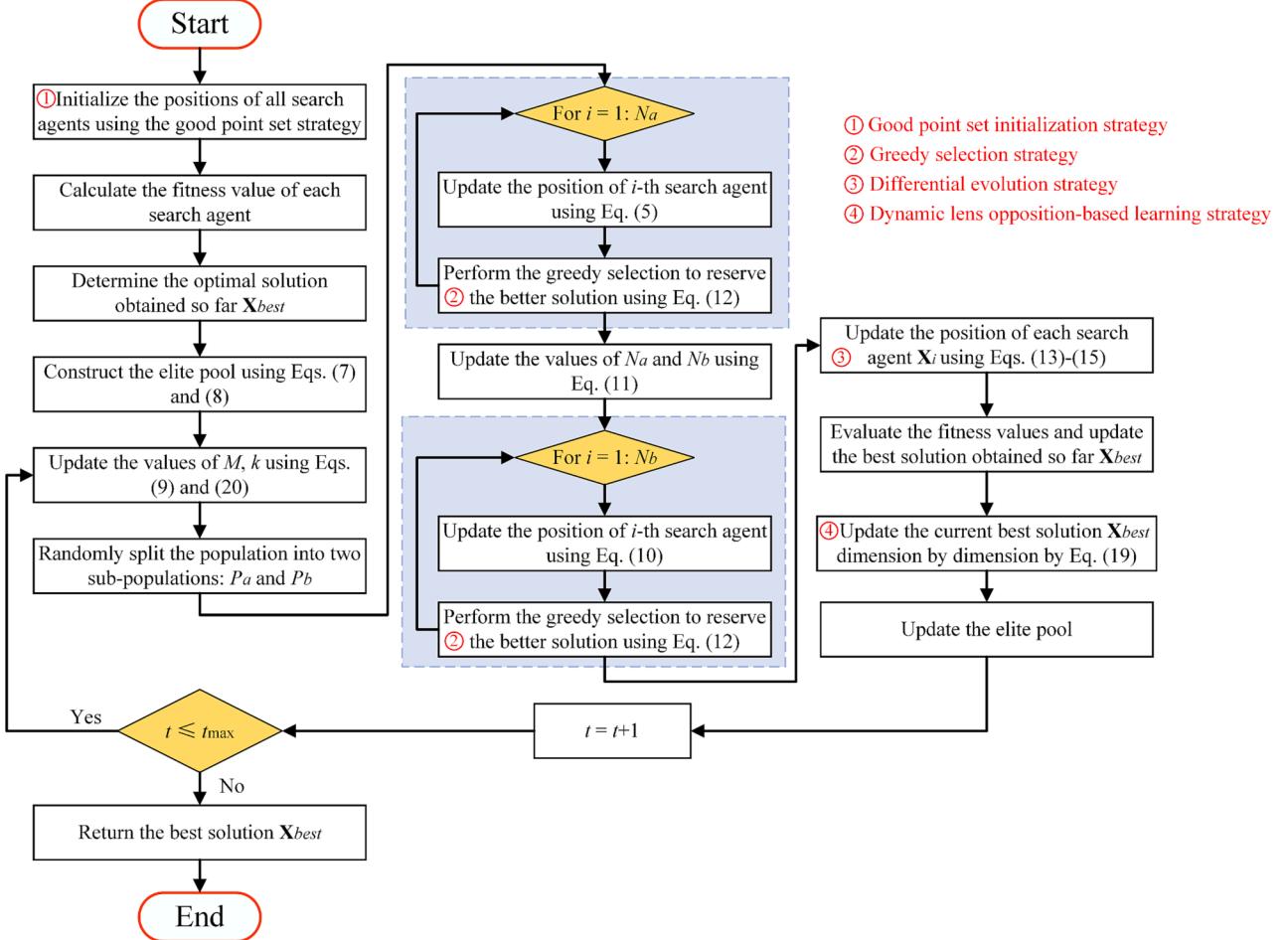


Fig. 8. Flow chart of the proposed MSAO.

set [93]. Unimodal functions have only one global optimum in the search domain, so this category is often used to evaluate an algorithm's exploitation propensity and convergence rate. On the other hand, multimodal functions feature numerous local interference solutions, often adopted to judge the algorithm's exploration potential to reach the global optimum. Hybrid functions merge the properties of fundamental functions, which can well simulate the actual search space to reflect the capability of the algorithm to deal with realistic optimization problems. Composition functions are ensembles of diverse hybrid functions that uphold the contiguity of the global and local optima, and they are commonly utilized to check the overall tracking competence of an algorithm. Table A.1 lists the function names, dimension sizes, search ranges, and theoretical optimal values for these functions.

Moreover, the IEEE CEC2022 test set [94] is further employed to delve deeper into the searchability and robustness of MSAO in tackling more competitive numerical optimization objects. This test suite comprises 12 benchmark functions, which can be similarly categorized into: unimodal function (F_{31}), multimodal functions ($F_{32} \sim F_{35}$), hybrid functions ($F_{36} \sim F_{38}$), and composition functions ($F_{39} \sim F_{42}$). These functions are characterized as nonlinear, nonconvex, non-derivable, and highly complex, for their details refer to Table A.2.

In the experiments, two evaluation metrics are utilized to intuitively compare and analyze the optimization performance of each method.

- ① Good point set initialization strategy
- ② Greedy selection strategy
- ③ Differential evolution strategy
- ④ Dynamic lens opposition-based learning strategy

(1) The average value (Mean)

The mean of the fitness gained by an algorithm over multiple operations, known as the average value, is determined using the following calculation:

$$Mean = \frac{1}{n} \sum_{i=1}^n g_i^* \quad (21)$$

where g_i^* refers to the optimal fitness value achieved in the i -th run, and n is the total count of runs. If the mean value is theoretically closer to the best solution, the algorithm has higher convergence accuracy.

(2) The standard deviation (Std)

The standard deviation quantifies the dispersion of the optimization results. A smaller standard deviation indicates improved computational stability of the algorithm. It is formulated as follows:

$$Std = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (g_i^* - Mean)^2} \quad (22)$$

From a statistical perspective, two non-parametric methods, namely

Wilcoxon rank-sum test [95] and Friedman ranking test [96], are implemented to unveil significant differences between MSAO and other comparison optimizers. The significance level for Wilcoxon rank-sum test is fixed to 0.05. Specifically, if the p -value derived on a test case is less than 0.05, this indicates that MSAO outperforms the comparison algorithm ('+'). Conversely, in case the p -value exceeds 0.05, this signifies that MSAO performs worse than the comparison algorithm ('-'). Moreover, NaN indicates that MSAO and the comparison method exhibit the same performance in statistical aspects ('='). The Friedman ranking test determines the mean rank of an optimizer over all test functions ('Mean rank'), where the smallest value is the best.

4.2. Results comparisons and analysis of CEC2017 benchmark functions

In this subsection, a total of 12 MAs will compete with the proposed MSAO on the IEEE CEC2017 benchmark suite concurrently. Nine of these algorithms were proposed in the last three years, including the original SAO [80], AOA [32], SMA [43], AO [44], ARO [45], GJO [46], TSA [53], DO [57], and SO [58], whereas the remaining four are CEC competition regulars, i.e., Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [97], LSHADE with Semi-Parameter Adaptation Hybrid with CMA-ES (LSHAD-SPACMA) [98], and Ensemble Sinusoidal Differential Covariance Matrix Adaptation with Euclidean Neighborhood (LSHAD-cnEpSin) [99]. The parameter configurations for each method align with the respective references, as indicated in Table 2.

The experiment commences with a parameter sensitivity analysis to select the most suitable crossover probability (CR) for subsequent sessions. Then, the effectiveness of the four improvement strategies is verified. Following that, a qualitative analysis of the optimization performance of MSAO on specific functions is provided. Finally, a detailed comparison is made between MSAO and other selected algorithms based on numerical statistical results, convergence behavior, boxplot, computation time, and multi-dimensional scalability.

Table 2
Parameter configurations for MARO other comparative algorithms.

Algorithm	Reference	Year	Parameter settings
AOA	[32]	2021	$\alpha = 5; \mu = 0.499$
SMA	[43]	2020	$z = 0.03$
AO	[44]	2021	$\omega = 0.005;$ $R = 10;$ $\alpha = 0.1;$ $\delta = 0.1;$ $g_1 \in [-1, 1];$ $g_2 \in [2, 0]$
ARO	[45]	2022	—
GJO	[46]	2022	$E_1 = [2, 0]$
TSA	[53]	2020	$P_{min} = 1; P_{max} = 4$
DO	[57]	2022	$\alpha \in [0, 1]; k \in [0, 1]$
SO	[58]	2022	$c_1 = 0.5; c_2 = 0.05; c_3 = 2$
SAO	[80]	2023	—
CMA-ES	[97]	2003	$\alpha = 2$
LSHAD-SPACMA	[98]	2017	$H = 1.4;$ $P_{best} = 0.11;$ $Arc_rate = 5;$ $Fcp = 0.5;$ $c = 0.8$
LSHAD-cnEpSin	[99]	2017	$H = 5;$ $NP_{min} = 4;$ $P_{best} = 0.11;$ $Arc_rate = 2;$ $pc = 0.4;$ $ps = 0.5$

4.2.1. Sensitivity analysis of crossover probability (CR)

In Subsection 3.3, the DE operators are introduced to enhance the search accuracy of MSAO. The crossover probability (CR) of DE takes a value between 0 and 1. To ensure that MSAO can demonstrate best search capability, a parameter sensitivity analysis is carried out in this part. The crossover probability (CR) is evenly distributed within the interval (0, 1). Table 3 presents the experimental results of MSAO with different CR values, obtained from 30 independent runs on 29 10-dimensional CEC2017 functions.

From this table, MSAO obtains optimal mean and standard deviation values on 13 out of 29 functions when $CR = 0.8(44.83\%)$, higher than when $CR = 0.1(17.24\%)$, when $CR = 0.2(0\%)$, when $CR = 0.3(6.90\%)$, when $CR = 0.4(6.90\%)$, when $CR = 0.5(17.24\%)$, when $CR = 0.6(0\%)$, when $CR = 0.7(3.45\%)$, and when $CR = 0.9(10.34\%)$. For unimodal functions F_1 and F_3 , the larger the value of CR , the more dominant the convergence accuracy and stability of MSAO. This is because that unimodal functions contain no local optima, so have little perturbation to MSAO, and a larger CR helps to explore superior potential solutions. For multimodal functions ($F_4 \sim F_{10}$), when $CR = 0.8$, MSAO ranks first on F_5 , F_7 and F_{10} , and ranks second on F_4 and F_8 . Multimodal functions exhibit numerous local optimal points, and a larger value of CR can strengthen the algorithm's global exploration ability. For hybrid functions ($F_{11} \sim F_{20}$), MSAO continues to provide the best performance at $CR = 0.8$, except for F_{16} and F_{20} . On these two functions, the smaller value of CR outperforms the larger value of CR , whereas the convergence accuracy of the latter does not deteriorate severely and is generally acceptable. This result suggests that larger values of CR can also prevent MSAO from falling into the local optimum when confronted with complex functions. For composition functions ($F_{21} \sim F_{30}$), most of the test cases yield better results when CR is equal to 0.1 than when CR is equal to 0.8, but the difference between both is not significant. Fig. 9 presents the Friedman mean ranking of MSAO with different CR values on 29 CEC2017 benchmark functions. Clearly, when CR is fixed to 0.8, MSAO exhibits the best overall optimization performance since its ranking value of 2.93 is the smallest among all comparison schemes. Therefore, in the following experiments, $CR = 0.8$.

4.2.2. Effectiveness validation of different components

To boost the optimization performance of the conventional SAO algorithm, this study employs four improvement strategies, namely good point set, greedy selection, DE, and DLOBL. To confirm the effectiveness of each component, ten different MSAO-derived algorithms with one or more fusion strategies are designed as shown in Table 4, where 1 represents that the strategy is embedded; instead, 0 indicates that the strategy is not embedded. The performance of SAO, MSAO, and multiple MSAO-derived variants are simultaneously evaluated for ablation in the CEC2017 test set, and the findings are documented in Table 5.

As can be seen from Table 5, the full-fledged MSAO with four strategies clearly outperforms other comparison methods, yielding the most satisfactory mean fitness and standard deviation in 23 out of the 29 test cases, followed by MSAO-8 in 4, MSAO-4 in 1, and MSAO-9 in 1. According to the final Friedman mean ranking values, each MSAO-derived variant benefiting from one single strategy addition show a great improvement in overall optimization performance compared with the original SAO, but still lacks a certain degree of robustness, and the contribution effect of each strategy can be ranked as follows: greedy selection > DE > DLOBL > good point set initialization. For unimodal functions F_1 and F_3 , the solutions provided by MSAO-8 is the closest to the theoretical value except for MSAO, indicating that the greedy selection strategy and DE strategy are helpful to enhance the local search capability of SAO. For multimodal functions ($F_4 \sim F_{10}$) with many local minima, the combination of greedy selection and DE strategy remains powerful and can effectively overcome the local optimum stagnation. In addition, the inclusion of the designed DLOBL can also extend the algorithm's search depth to some extent and increase the probability of locating the global optimal point. For hybrid and composition functions

Table 3

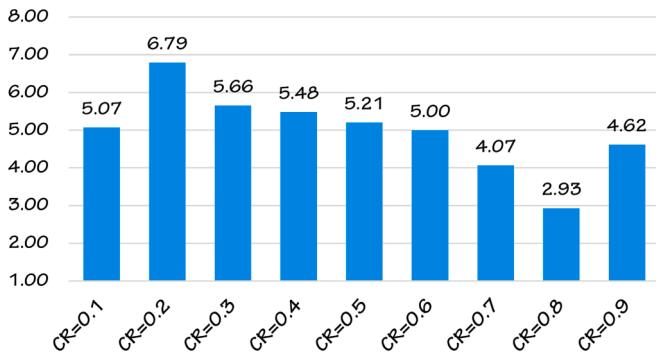
Experimental results for different CR values on 29 CEC2017 functions.

Function	Metric	CR = 0.1	CR = 0.2	CR = 0.3	CR = 0.4	CR = 0.5	CR = 0.6	CR = 0.7	CR = 0.8	CR = 0.9
F_1	Mean	2560.4304	913.9622	139.9414	101.5995	100.8533	100.2997	100.0403	100.0033	100.0002
	Std	3.0045E + 03	1.2018E + 01	8.7821E + 01	1.8216E + 00	1.4237E + 00	3.5966E-01	7.0150E-02	1.1495E-02	6.0631E-04
F_3	Rank	9	8	7	6	5	4	3	2	1
	Mean	300.1028	300.0621	300.0005	300.0030	300.0000	300.0000	300.0000	300.0000	300.0000
F_4	Std	1.8970E-01	2.7324E-01	1.9184E-03	1.6205E-02	1.1694E-07	1.7599E-09	1.4047E-11	1.4928E-14	1.8283E-14
	Rank	9	8	6	7	5	4	3	1	2
F_5	Mean	403.7924	403.4144	403.1217	402.4973	401.3810	400.6438	400.1834	400.0563	400.0154
	Std	7.6126E-01	7.8960E-01	6.2424E-01	4.3470E-01	4.8111E-01	2.6329E-01	6.9754E-02	5.6958E-02	3.5263E-02
F_6	Rank	9	8	7	6	5	4	3	2	1
	Mean	509.1855	510.4111	511.3646	510.1138	509.9650	506.3355	506.9686	506.3346	509.6179
F_7	Std	2.0793E + 00	4.0583E + 00	4.6985E + 00	4.3255E + 00	5.9257E + 00	2.9063E + 00	3.3340E + 00	1.8917E + 00	5.2410E + 00
	Rank	4	8	9	7	6	2	3	1	5
F_8	Mean	600.0000	600.0000	600.0000	600.0000	600.0000	600.0000	600.0000	600.0000	600.0000
	Std	2.2152E-06	2.0691E-06	0.0000E + 00	2.1111E-14	0.0000E + 00	1.1046E-05	2.0031E-05	1.5033E-04	5.9166E-04
F_9	Rank	5	4	1	3	1	6	7	8	9
	Mean	721.1426	723.1202	723.0296	726.8733	723.9878	722.3964	717.1882	716.3781	718.9274
F_{10}	Std	2.4795E + 00	3.0377E + 00	3.8397E + 00	4.6869E + 00	6.0105E + 00	6.0718E + 00	4.9020E + 00	2.1603E + 00	4.0855E + 00
	Rank	4	7	6	9	8	5	2	1	3
F_{11}	Mean	809.2140	809.9926	811.4149	811.6220	810.3534	807.3072	806.7657	807.1637	807.5285
	Std	3.5769E + 00	3.5481E + 00	4.9533E + 00	5.1542E + 00	6.7087E + 00	4.9758E + 00	2.0254E + 00	3.0291E + 00	3.3327E + 00
F_{12}	Rank	5	6	8	9	7	3	1	2	4
	Mean	900.0151	900.0151	900.0000	900.0151	900.0000	900.0809	900.0303	900.0333	900.1394
F_{13}	Std	8.2948E-02	8.2948E-02	0.0000E + 00	8.2948E-02	0.0000E + 00	2.1807E-01	1.1527E-01	1.1561E-01	5.2448E-01
	Rank	3	3	1	3	1	8	6	7	9
F_{14}	Mean	1581.3018	1719.3045	1758.5095	1758.2807	1784.8862	1674.1060	1581.4468	1300.3483	1340.3695
	Std	2.0974E + 02	2.0013E + 02	1.8786E + 02	3.0104E + 02	3.3071E + 02	3.6660E + 02	4.4446E + 02	1.6136E + 02	2.2221E + 02
F_{15}	Rank	3	6	8	7	9	5	4	1	2
	Mean	1103.2327	1102.4806	1102.3515	1102.0049	1101.9299	1101.9955	1102.6732	1102.5776	1103.4306
F_{16}	Std	1.5806E + 00	1.1983E + 00	1.7123E + 00	1.5208E + 00	1.1715E + 00	1.8875E + 00	2.2234E + 00	1.6518E + 00	2.6148E + 00
	Rank	8	5	4	3	1	2	7	6	9
F_{17}	Mean	37274.7658	24586.4069	16850.0861	31267.5358	16893.4455	12766.5916	9796.9822	3077.0714	4438.0314
	Std	6.4791E + 04	4.1449E + 04	3.0289E + 04	6.9442E + 04	2.6037E + 04	1.6449E + 04	9.9992E + 03	2.1080E + 03	6.0526E + 03
F_{18}	Rank	9	7	5	8	6	4	3	1	2
	Mean	3839.2806	1712.6371	1336.4342	1326.9951	1322.2808	1318.7802	1314.0091	1307.3069	1308.6183
F_{19}	Std	3.9028E + 03	7.5922E + 01	1.3299E + 01	6.3384E + 00	5.4503E + 00	4.1650E + 00	3.3222E + 00	2.8332E + 00	3.8351E + 00
	Rank	9	8	7	6	5	4	3	1	2
F_{20}	Mean	1416.7778	1418.1471	1419.6111	1420.9773	1420.2854	1416.0903	1405.4570	1403.0020	1411.0517
	Std	7.5068E + 00	4.8773E + 00	5.3055E + 00	3.9584E + 00	4.5015E + 00	7.5988E + 00	6.1777E + 00	2.9582E + 00	1.3636E + 01
F_{21}	Rank	5	6	7	9	8	4	2	1	3
	Mean	1505.9116	1504.1785	1503.9799	1504.2717	1504.0965	1503.5139	1502.2205	1501.6939	1504.6280
F_{22}	Std	2.7408E + 00	1.4107E + 00	1.1514E + 00	1.2147E + 00	1.1168E + 00	1.3880E + 00	1.3049E + 00	9.1518E-01	1.3902E + 01
	Rank	9	6	4	7	5	3	2	1	8
F_{23}	Mean	1608.7150	1610.9884	1606.9931	1611.1331	1606.3392	1611.8256	1616.0303	1627.5052	1652.7218
	Std	2.3134E + 01	3.0040E + 01	2.2154E + 01	2.9690E + 01	2.1824E + 01	3.3549E + 01	3.9339E + 01	4.9965E + 01	6.6319E + 01
F_{24}	Rank	3	4	2	5	1	6	7	8	9
	Mean	1714.8147	1720.0396	1722.1020	1721.9487	1724.6707	1718.6356	1718.3731	1711.2049	1717.8347
F_{25}	Std	8.2381E + 00	7.9449E + 00	6.0953E + 00	7.3909E + 00	9.4348E + 00	1.1003E + 00	1.0493E + 01	5.7830E + 00	1.1626E + 01
	Rank	2	6	8	7	9	5	4	1	3
F_{26}	Mean	2391.6405	1867.5061	1834.2382	1830.1910	1826.3770	1823.2360	1820.4688	1811.7660	1815.3789
	Std	6.9587E + 02	3.6584E + 01	8.0034E + 00	4.3971E + 00	9.3986E + 00	4.2144E + 00	4.1028E + 00	2.5590E + 00	2.7746E + 01
F_{27}	Rank	9	8	7	6	5	4	3	1	2
	Mean	1904.5463	1902.5900	1902.3533	1902.3967	1902.3103	1901.8503	1901.4763	1900.8559	1901.0056
F_{28}	Std	3.4035E + 00	8.2244E-01	8.2446E-01	6.8621E-01	7.0651E-01	6.8413E-01	7.1552E-01	4.8723E-01	9.7961E-01
	Rank	9	8	6	7	5	4	3	1	2
F_{29}	Mean	2000.0450	2000.1496	2000.0833	2000.2978	2000.2890	2001.0094	2003.0113	2004.7759	2008.3589
	Std	1.3538E-01	3.4865E-01	1.6258E-01	5.3478E-01	4.8856E-01	3.6110E + 00	6.3467E + 00	7.6849E + 00	9.0236E + 00
F_{30}	Rank	1	3	2	5	4	6	7	8	9
	Mean	2286.3267	2301.3326	2300.9955	2289.1774	2295.2760	2305.8136	2294.8198	2287.4408	2295.9087

(continued on next page)

Table 3 (continued)

Function	Metric	CR = 0.1	CR = 0.2	CR = 0.3	CR = 0.4	CR = 0.5	CR = 0.6	CR = 0.7	CR = 0.8	CR = 0.9
F_{22}	Std	2.0555E + 01	3.4283E + 01	3.4609E + 01	4.5160E + 01	3.7426E + 01	4.6671E + 01	3.7734E + 01	4.4553E + 01	3.8235E + 01
	Rank	1	8	7	3	5	9	4	2	6
	Mean	2294.9135	2300.9692	2300.6186	2297.8455	2300.7742	2300.7172	2300.6519	2300.7468	2298.5011
	Std	4.2038E-01	8.0653E-01	4.7965E-01	1.5552E + 01	6.3540E-01	2.2534E + 01	4.6784E-01	4.4238E-01	1.4878E + 01
F_{23}	Rank	1	9	4	2	8	6	5	7	3
	Mean	2610.3820	2611.2134	2612.8430	2608.5929	2609.5274	2609.1355	2610.0899	2609.5201	2611.4436
	Std	3.0982E + 00	3.9082E + 00	4.6336E + 00	2.7740E + 00	3.6631E + 00	2.9252E + 00	4.2654E + 00	3.2352E + 00	4.0411E + 00
F_{24}	Rank	6	7	9	1	4	2	5	3	8
	Mean	2722.1890	2731.3887	2730.9855	2740.1962	2730.0867	2737.5245	2737.8861	2723.5187	2740.0394
	Std	3.5328E + 00	4.3561E + 01	4.0682E + 01	5.4454E + 00	4.3732E + 01	5.8146E + 01	6.0935E + 01	4.4157E + 00	4.9748E + 00
F_{25}	Rank	1	5	4	9	3	6	7	2	8
	Mean	2927.8960	2935.4262	2929.6516	2928.2860	2929.0988	2932.7113	2929.5411	2927.0730	2929.3500
	Std	2.2754E + 01	1.8934E + 01	2.2601E + 01	2.2986E + 01	2.2067E + 01	2.1165E + 01	2.2435E + 01	1.4259E + 01	2.2043E + 01
F_{26}	Rank	2	9	7	3	4	8	6	1	5
	Mean	2951.8657	2973.3668	2933.2764	2922.7433	2914.2909	2962.8761	2950.9877	2928.1858	2935.8519
	Std	1.8328E + 02	2.3380E + 02	1.6132E + 02	8.8317E + 01	3.6007E + 01	1.5193E + 02	1.2861E + 02	3.8789E + 01	1.1444E + 02
F_{27}	Rank	7	9	4	2	1	8	6	3	5
	Mean	3081.6490	3082.4997	3080.5135	3080.2208	3082.1377	3082.1615	3081.6299	3082.1553	3081.0824
	Std	4.6392E + 00	1.0023E + 01	4.4788E + 00	4.3405E + 00	5.4316E + 00	6.3894E + 00	5.4310E + 00	6.1135E + 00	5.0163E + 00
F_{28}	Rank	5	9	2	1	6	8	4	7	3
	Mean	3270.6072	3277.1190	3274.1373	3273.9323	3274.2012	3273.8449	3271.6105	3273.5502	3274.2788
	Std	3.6690E-01	7.6068E + 00	3.0721E + 00	4.7949E-01	2.8892E + 00	5.4003E-01	1.3309E + 01	4.8681E-01	6.1824E-01
F_{29}	Rank	1	9	6	5	7	4	2	3	8
	Mean	3169.6758	3171.5199	3175.5709	3175.4780	3175.5630	3172.9913	3169.4994	3155.1288	3157.6569
	Std	1.1918E + 01	1.6235E + 01	1.6615E + 01	1.4499E + 01	1.6919E + 01	1.4182E + 01	1.6799E + 01	1.0700E + 01	1.6861E + 01
F_{30}	Rank	4	5	9	7	8	6	3	1	2
	Mean	6687.5122	12315.6411	9604.9140	8843.9823	14048.5588	7965.3999	6441.1789	5161.9865	4604.5481
	Std	2.8809E + 03	2.2152E + 04	4.9547E + 03	5.6477E + 03	1.6929E + 04	3.9135E + 03	2.6354E + 03	3.1821E + 03	2.1795E + 03
	Rank	4	8	7	6	9	5	3	2	1

**Fig. 9.** Friedman mean ranking for different CR values on 29 CEC2017 functions.

($F_{11} \sim F_{30}$), MSAO-4 performs best on F_{24} , MSAO-8 provides the best performance on F_{11}, F_{14} , and F_{26} , MSAO-9 obtains the optimal results on F_{27} , and MSAO is the winner on the rest of benchmark functions. These

results demonstrate that the improvement strategy introduced in this paper can balance the advantages and drawbacks. The significance of the greedy selection strategy in complex optimization challenges cannot be overstated. It plays a pivotal role in balancing exploration and exploitation. Meanwhile, the good point set initialization strategy and the DE strategy also make corresponding contributions in enhancing the global exploration trend and expanding the search horizon respectively. The impact of a single strategy may be ambiguous, but the synergistic implementation of these four strategies is significant for boosting the overall performance of SAO.

4.2.3. Qualitative results analysis

In this subsection, the optimization performance of MSAO and SAO is qualitatively investigated on six representative benchmark functions (unimodal functions F_1 and F_3 , multimodal functions F_5 and F_7 , composition functions F_{24} and F_{26}) as depicted in Fig. 10. In this figure, six essential evaluation factors are presented: (1) function topology, (2) convergence behavior, (3) trajectory of the first dimension, (4) average population fitness value, (5) search history, and (6) exploration and exploitation trends.

Table 4

Different MSAO-derived variants with four improvement strategies.

Strategy	MSAO-1	MSAO-2	MSAO-3	MSAO-4	MSAO-5	MSAO-6	MSAO-7	MSAO-8	MSAO-9	MSAO-10	MSAO
Good point set initialization	1	0	0	0	1	1	1	0	0	0	1
Greedy selection	0	1	0	0	1	0	0	1	1	0	1
DE strategy	0	0	1	0	0	1	0	1	0	1	1
DLOBL strategy	0	0	0	1	0	0	1	0	1	1	1

Table 5

Experimental results of MSAO-derived variants on the CEC2017 test suite.

Function	Metric	SAO	MSAO-1	MSAO-2	MSAO-3	MSAO-4	MSAO-5	MSAO-6	MSAO-7	MSAO-8	MSAO-9	MSAO-10	MSAO
<i>F</i> ₁	Mean	3414.9264	3113.1160	1629.0341	2664.6941	2830.7710	1418.7358	3291.6229	2190.3407	100.0031	2511.8847	1641.0253	100.0018
	Std	3.6389E + 03	2.7713E + 03	2.0333E + 03	2.9448E + 03	2.3973E + 03	1.7163E + 03	3.7021E + 03	2.1565E + 03	8.7997E-03	2.7609E + 03	2.4145E + 03	2.6388E-03
	Rank	12	10	4	8	9	3	11	6	2	7	5	1
<i>F</i> ₃	Mean	5806.3946	491.4755	300.3982	5580.4557	6883.3530	494.6283	547.5531	455.3701	300.0000	300.1738	300.0442	300.0000
	Std	2.6525E + 03	4.2517E + 02	1.0913E + 00	1.7780E + 03	4.0251E + 03	5.0856E + 02	5.7810E + 02	2.3789E + 02	2.1111E-14	5.5270E-01	9.7628E-02	1.4928E-14
	Rank	11	7	5	10	12	8	9	6	2	4	3	1
<i>F</i> ₄	Mean	404.2956	406.0014	404.0779	402.7110	404.1719	403.7308	402.7017	403.9715	400.0450	403.5495	402.7700	400.0853
	Std	1.8409E + 00	1.0054E + 01	6.5997E-01	4.7804E-01	5.4948E-01	6.1895E-01	5.4672E-01	7.6630E-01	6.0947E-02	1.1538E + 00	4.6753E-01	2.2537E-01
	Rank	11	12	9	4	10	7	3	8	1	6	5	2
<i>F</i> ₅	Mean	514.0858	514.0524	509.2373	511.8393	511.1415	509.8990	510.3966	513.4159	507.7938	510.6169	511.3181	507.3627
	Std	6.6321E + 00	7.5357E + 00	3.8069E + 00	6.6192E + 00	5.6908E + 00	4.8712E + 00	4.4403E + 00	5.3853E + 00	3.5929E + 00	4.4070E + 00	6.9019E + 00	2.4455E + 00
	Rank	12	11	3	9	7	4	5	10	2	6	8	1
<i>F</i> ₆	Mean	600.0841	600.0040	600.0083	600.0024	600.0165	600.0067	600.0000	600.0008	600.0000	600.0001	600.0023	600.0000
	Std	2.6102E-01	2.1836E-02	4.5253E-02	9.6905E-03	8.6136E-02	3.6220E-02	1.7183E-05	4.1651E-03	9.3333E-06	3.4569E-04	1.2522E-02	1.2237E-06
	Rank	12	8	10	7	11	9	3	5	2	4	6	1
<i>F</i> ₇	Mean	738.4698	718.1083	721.8245	739.2013	720.6213	724.6629	719.8603	718.0372	717.3635	724.0440	738.3797	716.9867
	Std	9.2490E + 00	3.4575E + 00	6.5433E + 00	8.2305E + 00	5.6185E + 00	8.1809E + 00	3.0210E + 00	3.2358E + 00	3.7719E + 00	7.2029E + 00	9.4248E + 00	2.5369E + 00
	Rank	11	4	7	12	6	9	5	3	2	8	10	1
<i>F</i> ₈	Mean	813.0964	812.2048	811.7215	810.9379	813.4783	809.1780	811.7983	811.4132	808.6230	810.0777	811.4208	808.0260
	Std	5.5119E + 00	5.1530E + 00	5.2145E + 00	5.6274E + 00	6.2392E + 00	4.7191E + 00	9.2246E + 00	5.2597E + 00	3.6425E + 00	6.0132E + 00	6.4139E + 00	3.6148E + 00
	Rank	11	10	8	5	12	3	9	6	2	4	7	1
<i>F</i> ₉	Mean	901.1090	900.2575	900.2193	900.1756	900.3983	900.1617	900.1242	901.0293	900.1030	900.1120	900.1999	900.0151
	Std	4.0627E + 00	1.0787E + 00	5.5269E-01	6.2173E-01	1.1303E + 00	4.5577E-01	5.2162E-01	4.4137E + 00	3.3661E-01	2.2846E-01	6.5464E-01	8.2948E-02
	Rank	12	9	8	6	10	5	4	11	2	3	7	1
<i>F</i> ₁₀	Mean	1638.5295	1564.3477	1605.5122	1580.6489	1566.2187	1636.0892	1671.8333	1616.3815	1290.4279	1526.2300	1529.5748	1285.4477
	Std	3.4149E + 02	2.6589E + 02	3.3298E + 02	3.6844E + 02	2.2848E + 02	3.7448E + 02	3.4483E + 02	2.5521E + 02	2.1710E + 02	4.2544E + 02	2.4336E + 02	2.1253E + 02
	Rank	11	5	8	7	6	10	12	9	2	3	4	1
<i>F</i> ₁₁	Mean	1110.3815	1105.5807	1105.5717	1107.5362	1106.3375	1106.0561	1104.0512	1105.5429	1102.0377	1106.0073	1103.4974	1102.4596
	Std	1.7703E + 01	3.6534E + 00	3.5264E + 00	1.9644E + 01	4.5572E + 00	3.1374E + 00	2.9431E + 00	3.9008E + 00	1.5415E + 00	3.8009E + 00	2.6749E + 00	2.2318E + 00
	Rank	12	7	6	11	10	9	4	5	1	8	3	2
<i>F</i> ₁₂	Mean	17399.4610	12814.2740	13241.6884	13049.9087	14030.6311	14374.6815	12432.4777	12937.8524	5799.6961	11693.6192	11956.1109	4408.7005
	Std	1.5207E + 04	9.3929E + 03	1.2526E + 04	1.2393E + 04	1.2717E + 04	1.2287E + 04	8.1562E + 03	9.8658E + 03	7.6766E + 03	8.6806E + 03	8.7298E + 03	5.0735E + 03
	Rank	12	6	9	8	10	11	5	7	2	3	4	1
<i>F</i> ₁₃	Mean	14957.8372	12526.1132	13062.2433	11753.4375	12049.0264	11891.8811	11622.1641	14107.2548	1309.2628	9937.9333	9099.7219	1309.2067
	Std	9.2128E + 03	1.0672E + 04	1.0462E + 04	9.1207E + 03	1.0765E + 04	8.7182E + 03	9.5234E + 03	1.0070E + 04	3.5817E + 00	6.6634E + 03	8.0936E + 03	2.6885E + 00
	Rank	12	9	10	6	8	7	5	11	2	4	3	1
<i>F</i> ₁₄	Mean	9656.6188	10700.2248	1495.5943	3068.1384	6551.5219	7407.5958	3286.9288	1528.3792	1404.9600	1487.0613	3605.7753	1406.5130
	Std	8.2564E + 03	7.6994E + 03	6.0329E + 01	2.5123E + 03	6.2636E + 03	6.9565E + 03	3.4576E + 03	7.2475E + 01	6.8096E + 00	5.2873E + 01	2.7880E + 03	8.6150E + 00
	Rank	11	12	4	6	9	10	7	5	1	3	8	2
<i>F</i> ₁₅	Mean	6736.2885	4545.1920	1896.3523	4691.9637	5421.1327	1884.9470	6638.1277	4966.7131	1501.3565	1955.4450	4473.2949	1501.3499
	Std	8.2548E + 03	5.7956E + 03	5.1172E + 02	4.3413E + 03	6.8223E + 03	4.7620E + 02	7.4367E + 03	5.6141E + 03	1.0416E + 00	5.9574E + 02	3.7107E + 03	9.4140E-01
	Rank	12	7	4	8	10	3	11	9	2	5	6	1
<i>F</i> ₁₆	Mean	1692.3994	1721.7038	1679.7733	1639.0762	1676.3343	1645.7674	1654.7116	1693.4729	1635.2233	1646.7656	1661.2103	1631.0015
	Std	1.0537E + 02	1.0671E + 02	8.6298E + 01	5.0713E + 01	9.5496E + 01	5.8410E + 01	6.6599E + 01	1.1886E + 02	7.0773E + 01	6.5463E + 01	8.2993E + 01	4.9941E + 01
	Rank	10	12	9	3	8	4	6	11	2	5	7	1
<i>F</i> ₁₇	Mean	1752.2136	1749.0233	1743.9726	1739.4180	1758.5240	1743.1470	1734.2496	1738.8210	1717.5731	1742.1941	1744.3748	1715.6115
	Std	3.4732E + 01	2.9754E + 01	2.8663E + 01	3.5908E + 01	3.8039E + 01	1.7523E + 01	2.6926E + 01	2.5291E + 01	1.1624E + 01	2.1494E + 01	2.7623E + 01	8.3644E + 00
	Rank	11	10	8	5	12	7	3	4	2	6	9	1
<i>F</i> ₁₈	Mean	21868.7121	18075.4119	16048.8481	18956.7697	18907.7983	17660.1164	13198.6385	17181.5370	1812.1000	16974.7028	18209.1238	1809.3199
	Std	1.3957E + 04	1.2425E + 04	9.9651E + 03	1.1341E + 04	1.4296E + 04	1.1866E + 04	7.6952E + 03	1.0219E + 04	1.0898E + 01	8.8531E + 03	1.2495E + 04	8.7045E + 00
	Rank	12	8	4	11	10	7	3	6	2	5	9	1
<i>F</i> ₁₉	Mean	10994.4118	10364.8119	2850.9790	8061.7002	12988.3602	2682.5608	9439.3277	10145.9343	1900.9771	2925.0515	10705.5834	1900.8887
	Std	8.4680E + 03	1.0033E + 04	1.4738E + 03	8.3833E + 03	1.2074E + 04	1.2708E + 03	1.0189E + 04	1.0616E + 04	8.3483E-01	1.6144E + 03	1.0396E + 04	7.4286E-01
	Rank	11	9	4	6	12	3	7	8	2	5	10	1

(continued on next page)

Table 5 (continued)

Function	Metric	SAO	MSAO-1	MSAO-2	MSAO-3	MSAO-4	MSAO-5	MSAO-6	MSAO-7	MSAO-8	MSAO-9	MSAO-10	MSAO
F_{20}	Mean	2071.6210	2070.2631	2017.4415	2024.5698	2068.6382	2019.7283	2046.0753	2062.1541	2004.5662	2017.2729	2022.2300	2003.8013
	Std	6.4685E + 01	6.5030E + 01	1.1545E + 01	3.9828E + 01	7.4201E + 01	2.3065E + 01	5.6728E + 01	6.1990E + 01	7.8019E + 00	2.3246E + 01	3.7489E + 01	6.6365E + 00
	Rank	12	11	4	7	10	5	8	9	2	3	6	1
F_{21}	Mean	2306.2079	2304.7036	2289.5201	2306.6792	2288.7575	2300.2701	2300.7310	2299.2907	2294.3831	2306.4884	2305.8316	2288.7168
	Std	2.9371E + 01	3.5776E + 01	4.4639E + 01	3.5207E + 01	4.9837E + 01	3.4144E + 01	3.9500E + 01	3.9283E + 01	3.7817E + 01	4.4811E + 01	3.1488E + 01	2.9112E + 01
	Rank	10	8	3	12	2	6	7	5	4	11	9	1
F_{22}	Mean	2301.6821	2301.5714	2297.8938	2301.1652	2299.6861	2301.3537	2300.8959	2339.9461	2300.6681	2301.1496	2300.8678	2296.7072
	Std	2.3667E + 00	1.7675E + 00	1.6314E + 01	1.8521E + 00	1.4012E + 01	1.0877E + 00	1.4525E + 00	2.2296E + 02	1.8645E + 01	6.6734E-01	5.6468E-01	5.0532E-01
	Rank	11	10	2	8	3	9	6	12	4	7	5	1
F_{23}	Mean	2618.3797	2615.0261	2614.6388	2613.6902	2617.8259	2611.2554	2614.9558	2616.4053	2611.6583	2612.5234	2615.4754	2609.3563
	Std	7.4903E + 00	5.8316E + 00	6.7648E + 00	4.4021E + 00	8.4801E + 00	4.4618E + 00	6.3111E + 00	6.1016E + 00	4.8895E + 00	7.1130E + 00	6.1917E + 00	4.1135E + 00
	Rank	12	8	6	5	11	2	7	10	3	4	9	1
F_{24}	Mean	2745.8426	2736.4602	2742.0522	2743.5048	2725.4697	2740.6252	2743.5973	2737.3818	2730.4879	2734.9328	2737.4552	2731.6441
	Std	7.1141E + 00	4.5270E + 01	4.5992E + 00	7.3077E + 00	4.3177E + 00	6.6989E + 01	7.0509E + 00	4.5139E + 01	4.3829E + 01	4.4453E + 01	3.5407E + 01	4.3898E + 01
	Rank	12	5	9	10	1	8	11	6	2	4	7	3
F_{25}	Mean	2940.2599	2931.9743	2937.9770	2933.0453	2929.7523	2938.7748	2931.8255	2930.9598	2936.0622	2936.3302	2931.7430	2926.7507
	Std	2.3240E + 01	2.2401E + 01	1.7940E + 01	2.1213E + 01	2.2845E + 01	1.8708E + 01	2.2251E + 01	2.3528E + 01	1.9007E + 01	1.9104E + 01	2.2130E + 01	1.6986E + 01
	Rank	12	6	10	7	2	11	5	3	8	9	4	1
F_{26}	Mean	3002.7550	3120.4534	3076.6023	2948.4807	2967.0289	2961.4797	2924.7255	2937.1075	2919.5767	2975.4882	2961.2512	2936.4801
	Std	2.3829E + 02	3.8685E + 02	3.2801E + 02	1.8557E + 02	1.9345E + 02	2.1833E + 02	5.4782E + 01	5.6917E + 01	4.7817E + 01	1.9787E + 02	1.7125E + 02	7.6916E + 01
	Rank	10	12	11	5	8	7	2	4	1	9	6	3
F_{27}	Mean	3098.0296	3098.0188	3080.0920	3089.7896	3096.3779	3080.9917	3091.4144	3095.0639	3081.8894	3079.8557	3090.2606	3079.9797
	Std	1.5442E + 01	8.3297E + 00	4.7801E + 00	3.3062E + 00	8.4420E + 00	5.5436E + 00	3.1009E + 00	3.8082E + 00	5.8040E + 00	2.5071E + 00	4.9779E + 00	6.0166E + 00
	Rank	12	11	3	6	10	4	8	9	5	1	7	2
F_{28}	Mean	3329.8256	3315.5522	3328.0032	3284.4255	3319.7346	3257.2198	3284.4535	3273.9795	3274.2223	3263.4421	3285.2742	3256.3107
	Std	1.2748E + 02	1.2531E + 02	1.3402E + 02	3.7286E + 00	1.3336E + 02	5.8246E + 01	3.5634E + 00	6.1587E + 01	6.4008E-01	4.7790E + 01	3.0748E + 00	4.7710E-01
	Rank	12	9	11	6	10	2	7	4	5	3	8	1
F_{29}	Mean	3211.0621	3209.3925	3201.3952	3189.6728	3196.9361	3205.2179	3175.3219	3192.0041	3161.9663	3192.7952	3171.5100	3157.5544
	Std	5.5688E + 01	5.0735E + 01	4.2056E + 01	5.8403E + 01	4.0434E + 01	4.4191E + 01	4.3751E + 01	4.9809E + 01	1.8521E + 01	3.4569E + 01	3.1554E + 01	1.2553E + 01
	Rank	12	11	9	5	8	10	4	6	2	7	3	1
F_{30}	Mean	4.1558E + 05	3.1476E + 05	3.5584E + 04	2.9423E + 04	3.5905E + 05	1.2109E + 04	2.0999E + 04	2.5305E + 05	7.6393E + 03	8.0791E + 03	1.0252E + 04	5.1155E + 03
	Std	5.9205E + 05	4.9568E + 05	5.1428E + 04	4.2064E + 04	5.1325E + 05	2.0657E + 04	2.1215E + 04	4.0342E + 05	9.4638E + 03	8.9873E + 03	1.1785E + 04	2.9619E + 03
	Rank	12	10	8	7	11	5	6	9	2	3	4	1
Mean rank		11.48	8.86	6.76	7.24	8.55	6.48	6.31	7.14	2.45	5.17	6.28	1.28
Final ranking		12	11	7	9	10	6	5	8	2	3	4	1

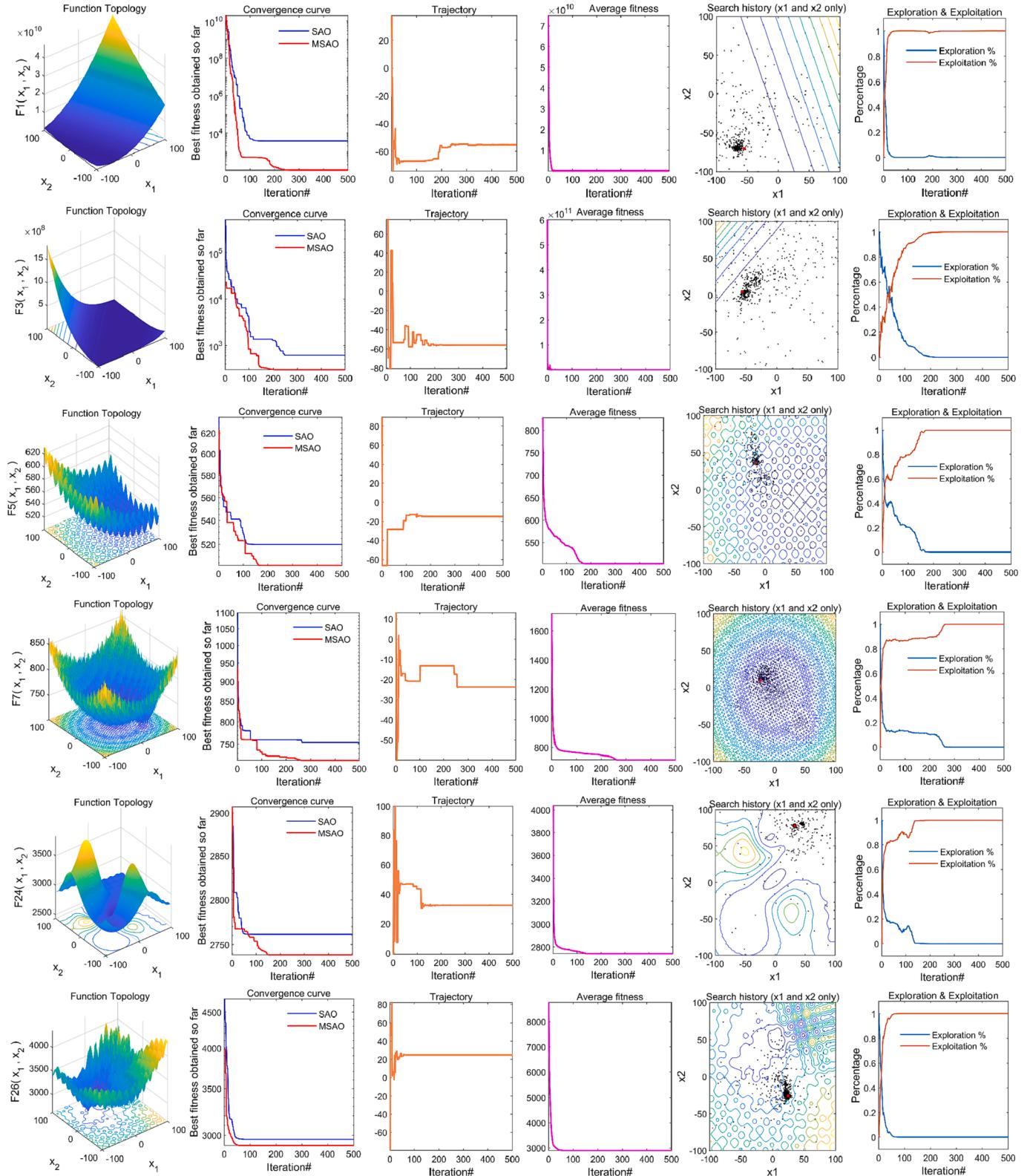


Fig. 10. Qualitative results for F1, F3, F5, F7, F24, and F26 benchmark functions.

In the 1st column of Fig. 10, a 3D view of the search space for these functions is presented. Here it can be intuitively observed that the unimodal function features only a single global minimum in the search domain, whereas the multimodal function contains numerous local optima and composition functions have a high degree of complexity. In the

2nd column of Fig. 10, the convergence behavior of the optimal solution throughout the iterations is visualized. Note that the proposed method efficiently locates promising regions in the initial search and then converges consistently towards the global optimal point in the later stages. Compared with SAO, MSAO achieves better convergence accuracy with

Table 6

Statistical results of MSAO and different algorithms on 29 CEC2017 functions.

Function	Metric	MSAO	AOA	SMA	AO	ARO	GJO	TSA	DO	SO	SAO	CMA-ES	LSHADE-SPACMA	LSHADE-cnEpSin
<i>F</i> ₁	Mean	100.0025	9.3562E + 09	6.4468E + 03	4.5246E + 07	3.6416E + 03	6.5947E + 08	3.7818E + 09	6.3652E + 03	3.1797E + 03	2.6007E + 03	8.6493E + 05	100.0039	100.0247
	Std	4.9827E-06	3.5826E + 09	4.2319E + 03	8.1395E + 07	3.6192E + 03	9.1404E + 08	3.1273E + 09	3.9976E + 03	3.6691E + 03	2.7411E + 03	1.1918E + 06	1.9108E-02	7.0039E-02
	p-value	-	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0180E-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	4.8011E-07	5.1060E-01
	Rank	1	13	8	10	6	11	12	7	5	4	9	2	3
<i>F</i> ₃	Mean	300.0000	11850.1410	303.7565	2662.8768	563.0353	4599.5418	11531.5014	300.2550	1221.8628	506.7798	61449.3636	1591.1694	300.0000
	Std	0.0000E + 00	2.9266E + 03	1.3785E + 01	8.4213E + 02	2.3829E + 02	2.9875E + 03	4.6438E + 03	4.4846E-01	1.4504E + 03	2.2654E + 02	3.3279E + 03	3.3675E + 03	9.0198E-10
	p-value	-	5.1436E-12	8.9456E-02	2.0079E-10									
	Rank	1	12	4	9	6	10	11	3	7	5	13	8	2
<i>F</i> ₄	Mean	400.0425	1095.0868	423.9179	433.5935	405.3225	458.9409	574.2078	408.0955	404.7365	403.9289	400.4812	400.1256	400.2322
	Std	2.9225E-02	4.1603E + 02	3.1017E + 01	3.0928E + 01	2.4997E + 00	4.2945E + 01	1.6083E + 01	1.2198E + 02	2.1098E + 01	8.4749E-01	2.5144E-01	4.7682E-01	3.5508E-01
	p-value	-	3.0199E-11	3.4742E-10	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	8.8910E-10	8.9934E-11	3.0199E-11	3.3384E-11	2.8913E-03	3.6439E-02
	Rank	1	13	9	10	7	11	12	8	6	5	4	2	3
<i>F</i> ₅	Mean	507.2970	565.9600	522.9427	534.9498	514.1792	531.5080	578.3246	527.8634	520.1523	512.3362	524.2626	505.0453	505.4984
	Std	2.6426E + 00	1.5704E + 01	8.6853E + 00	1.3888E + 01	5.8903E + 00	1.3745E + 01	1.9908E + 01	9.3019E + 00	6.9208E + 00	5.1045E + 00	1.0194E + 01	1.8994E + 00	2.1339E + 00
	p-value	-	2.9598E-11	2.8200E-10	2.5628E-10	4.6219E-08	4.8784E-11	2.9598E-11	3.6170E-11	1.1831E-10	5.1412E-05	1.0165E-06	3.1625E-04	6.9473E-03
	Rank	3	12	7	11	5	107	13	9	6	4	8	1	2
<i>F</i> ₆	Mean	600.0002	641.4342	600.1792	621.8902	600.1680	609.7926	648.0832	607.7613	602.0657	600.0570	600.0000	600.0108	600.0052
	Std	5.8267E-04	9.3422E + 00	1.1820E-01	5.6194E + 00	4.8360E-01	7.0726E + 00	1.0655E + 00	6.6606E + 01	2.2444E + 00	2.8601E-01	3.5778E-07	2.8032E-02	1.6071E-02
	p-value	-	2.1019E-11	1.3812E-01	6.2815E-01	4.7123E-08	3.5944E-07							
	Rank	2	12	7	11	6	10	13	9	8	5	1	4	3
<i>F</i> ₇	Mean	717.1160	801.6961	732.9575	758.0717	734.8489	759.2879	801.7668	753.5853	741.6370	720.0430	737.2781	715.7356	715.7595
	Std	2.8401E + 00	2.0826E + 01	9.8901E + 00	1.5792E + 01	8.8714E + 00	1.6102E + 01	1.7889E + 01	1.3403E + 01	1.1857E + 01	4.5455E + 01	5.6296E + 00	1.5126E + 01	1.9002E + 00
	p-value	-	3.0199E-11	8.1527E-11	3.0199E-11	4.9752E-11	3.0199E-11	3.0199E-11	3.0199E-11	9.8834E-03	4.9752E-11	4.5146E-02	4.5146E-02	
	Rank	3	12	5	10	6	11	13	9	8	4	7	1	2
<i>F</i> ₈	Mean	808.0260	837.1678	822.8345	825.2484	818.8528	827.5347	843.4033	827.8646	819.2460	812.2716	821.1078	805.6699	806.3781
	Std	3.7038E + 00	9.1289E + 00	1.1268E + 01	8.0782E + 00	5.8346E + 00	1.2038E + 01	9.2133E + 00	8.8573E + 00	6.8153E + 00	4.8979E + 00	1.1843E + 01	1.9033E + 00	2.4165E + 00
	p-value	-	2.9766E-11	3.1601E-09	8.8706E-11	1.8349E-09	3.1182E-10	2.9766E-11	8.8706E-11	7.0406E-09	5.4707E-04	1.1630E-03	1.4968E-02	1.5784E-03
	Rank	3	12	8	9	5	10	13	11	6	4	7	1	2
<i>F</i> ₉	Mean	900.1697	1423.9756	931.0620	1060.0913	902.1417	1006.1938	1601.4190	1023.9505	914.5687	900.4363	900.0000	900.1873	900.2269
	Std	5.9529E-01	1.7013E + 02	1.3329E + 02	8.6348E + 01	4.3849E + 01	1.0568E + 00	3.2979E + 02	1.9442E + 02	1.6218E + 01	1.2777E + 00	0.0000E + 00	3.7353E-01	4.5513E-01
	p-value	-	4.1110E-12	3.8231E-08	4.1110E-12	1.7238E-08	4.1110E-12	4.1110E-12	2.8201E-11	5.2692E-11	1.3682E-01	4.1926E-02	6.5789E-03	1.3108E-04
	Rank	2	12	8	11	6	9	13	10	7	5	1	3	4
<i>F</i> ₁₀	Mean	1286.6999	2235.1862	1658.4135	2018.1239	1511.1977	1928.4172	2676.0366	1924.6731	1620.6978	1651.7644	2690.0289	1316.0802	1305.5652
	Std	1.3577E + 02	2.7402E + 02	2.8105E + 02	3.4560E + 02	2.4998E + 02	3.7502E + 02	2.6439E + 02	3.2414E + 02	2.2072E + 02	3.0116E + 02	1.9842E + 02	1.4304E + 02	2.7562E + 02
	p-value	-	4.1110E-12	3.8231E-08	4.1110E-12	1.7238E-08	4.1110E-12	4.1110E-12	2.8201E-11	5.2692E-11	1.3682E-01	4.1926E-02	6.5789E-03	1.3108E-04

(continued on next page)

Table 6 (continued)

Function	Metric	MSAO	AOA	SMA	AO	ARO	GJO	TSA	DO	SO	SAO	CMA-ES	LSHADE-SPACMA	LSHADE-cnEpSin
	p-value	-	1.3289E-10	5.0912E-06	1.4294E-08	1.0576E-03	2.1947E-08	4.0746E-11	1.4294E-08	1.7479E-05	1.7479E-05	3.0199E-11	1.9073E-03	3.3285E-02
F_{11}	Rank	1	11	7	10	4	9	12	8	5	6	13	3	2
	Mean	1102.2128	3903.1367	1176.8806	1247.0690	1111.7090	1221.0142	4373.8328	1132.2280	1119.3371	1112.1852	1425.2585	1106.0755	1107.0996
	Std	1.5605E + 00	2.4288E + 03	9.4346E + 01	7.9788E + 01	6.7653E + 00	1.0535E + 00	3.8624E + 02	2.8762E + 03	1.4866E + 01	2.7085E + 01	3.6468E + 02	7.9957E + 00	1.1547E + 01
	p-value	-	3.0199E-11	5.4941E-11	3.0199E-11	1.4643E-10	3.0199E-11	3.0199E-11	3.0199E-11	1.6132E-10	7.7387E-06	3.0199E-11	1.4932E-04	1.8916E-04
F_{12}	Rank	1	12	8	10	4	9	13	7	6	5	11	2	3
	Mean	5748.0840	1.9308E + 08	3.0164E + 05	3.6698E + 06	4.9207E + 04	1.7886E + 06	2.6065E + 07	7.3007E + 05	5.5211E + 04	1.6115E + 04	5.9816E + 06	2307.1586	1942.2647
	Std	6.6603E + 03	2.2673E + 08	3.4785E + 05	3.3831E + 06	1.0030E + 05	2.4636E + 06	1.0390E + 08	9.8644E + 05	1.2930E + 05	1.6444E + 04	6.3736E + 06	2.0015E + 03	6.4414E + 02
	p-value	-	4.0772E-11	4.1997E-10	3.0199E-11	1.1674E-05	3.3384E-11	3.0199E-11	1.0937E-10	1.0277E-06	9.2113E-05	3.0199E-11	2.2780E-05	1.4298E-05
F_{13}	Rank	3	13	7	10	5	9	12	8	6	4	11	2	1
	Mean	1309.7350	1.2434E + 04	1.4592E + 04	1.3705E + 04	2762.4654	1.5479E + 04	2.4226E + 07	1.2787E + 04	5918.2630	9809.9452	8.1898E + 04	1349.4233	1325.3894
	Std	3.4039E + 00	9.7322E + 03	1.3177E + 04	1.0140E + 04	3.1328E + 03	9.6080E + 03	1.0852E + 08	9.9924E + 03	6.1159E + 03	8.9605E + 03	5.9499E + 04	5.6402E + 01	1.8816E + 01
	p-value	-	3.0199E-11	4.6159E-10	3.0199E-11	5.4620E-06	8.2919E-06							
F_{14}	Rank	1	7	10	9	4	11	13	8	5	6	12	3	2
	Mean	1407.8219	11360.2995	2813.5485	2453.9109	1409.4242	3426.8355	4702.3138	3584.6193	1559.6409	7528.1760	5316.0239	1417.1450	1422.0299
	Std	6.3516E + 00	8.7642E + 03	3.3626E + 03	1.0115E + 03	8.8376E + 00	1.7513E + 03	1.4278E + 03	2.0349E + 03	1.0821E + 03	7.3651E + 02	6.6665E + 03	1.0599E + 01	1.2434E + 01
	p-value	-	3.0085E-11	3.0085E-11	3.0085E-11	1.9102E-02	3.0085E-11	3.0085E-11	3.0085E-11	7.3624E-11	3.0085E-11	4.3467E-05	1.2834E-06	
F_{15}	Rank	1	13	7	6	2	8	10	9	5	12	11	3	4
	Mean	1501.5138	18063.0024	9570.1737	7468.9756	1553.0833	4840.3077	12041.6234	3942.9651	2290.7499	5475.8990	23981.6151	1508.8100	1510.0832
	Std	8.0737E-01	6.6113E + 03	7.0081E + 03	3.3229E + 03	2.2904E + 02	3.0210E + 03	6.7005E + 03	3.1975E + 03	9.1584E + 02	6.4259E + 03	2.2613E + 04	8.2345E + 00	1.2702E + 01
	p-value	-	3.0199E-11	3.0199E-11	3.0199E-11	1.9568E-10	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	1.2860E-06	4.7445E-06	
F_{16}	Rank	1	12	10	9	4	7	11	6	5	8	13	2	3
	Mean	1604.0606	2057.6090	1744.0013	1837.0004	1710.9600	1800.9048	2120.0700	1804.4114	1712.4336	1697.0532	1688.3684	1636.0122	1626.6194
	Std	5.3788E + 00	1.5641E + 02	1.2711E + 02	1.4482E + 02	1.0666E + 02	1.6689E + 02	1.0732E + 02	1.4260E + 02	1.0368E + 02	1.2425E + 02	4.9711E + 01	5.1501E + 01	5.7668E + 01
	p-value	-	3.0199E-11	1.9568E-10	3.0199E-11	2.5721E-07	3.0199E-11	3.0199E-11	3.0199E-11	1.1023E-08	9.0632E-08	7.3891E-11	1.6351E-05	6.5486E-04
F_{17}	Rank	1	12	8	11	6	9	13	10	7	5	4	3	2
	Mean	1711.4358	1875.7361	1761.1927	1778.5738	1727.7283	1781.8581	1864.5659	1766.1940	1759.2963	1740.2411	1795.6833	1710.0150	1718.7022
	Std	1.0370E + 01	1.2631E + 02	4.3090E + 01	2.6540E + 01	2.5917E + 01	3.7119E + 01	9.3369E + 01	3.6377E + 01	4.4135E + 01	2.4999E + 01	2.6613E + 01	9.1416E + 00	1.4761E + 01
	p-value	-	3.0199E-11	1.4110E-09	3.3384E-11	8.6634E-05	3.6897E-11	3.0199E-11	2.8716E-10	6.1210E-10	9.0632E-08	3.0199E-11	8.1875E-01	1.3832E-02
F_{18}	Rank	2	13	7	9	4	10	12	8	6	5	11	1	3
	Mean	1810.5145	6.6795E + 06	3.4489E + 04	3.3220E + 04	2926.2301	3.7839E + 04	2.0844E + 05	1.9833E + 04	9.1820E + 03	1.8536E + 04	2.5692E + 05	1846.6019	1839.6896
	Std	9.5443E + 00	3.0358E + 07	1.0281E + 04	1.3901E + 04	2.9838E + 03	9.5333E + 03	2.1761E + 05	1.4269E + 04	7.0546E + 03	1.1548E + 04	2.2895E + 05	4.3611E + 01	2.8577E + 01
	p-value	-	3.0199E-11	3.0199E-11	3.0199E-11	2.1947E-08	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	7.3803E-10	2.6015E-08
	Rank	1	13	9	8	4	10	11	7	5	6	12	3	2

(continued on next page)

Table 6 (continued)

Function	Metric	MSAO	AOA	SMA	AO	ARO	GJO	TSA	DO	SO	SAO	CMA-ES	LSHADE-SPACMA	LSHADE-cnEpSin
F_{19}	Mean	1900.9203	1.0246E + 05	1.0975E + 04	3.2439E + 04	1917.3139	1.9445E + 04	5.0299E + 05	8416.5131	4302.3775	11303.4184	2.4596E + 04	1903.8815	1906.1946
	Std	7.0158E-01	8.3414E + 04	1.0488E + 04	7.0874E + 04	6.7674E + 01	4.6774E + 04	6.2192E + 05	6.7780E + 03	3.6325E + 03	1.0454E + 04	3.6555E + 04	3.8751E + 00	4.7996E + 00
	p-value	-	3.0199E-11	3.0199E-11	3.0199E-11	1.4733E-07	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	3.3384E-11	3.0199E-11	4.4440E-07	5.9673E-09
	Rank	1	12	7	11	4	9	13	6	5	8	10	2	3
F_{20}	Mean	2003.0026	2178.0681	2041.2359	2140.6053	2016.3732	2137.3453	2239.9050	2127.7009	2050.7893	2059.7163	2080.6238	2005.6542	2012.5675
	Std	6.0064E + 00	7.7552E + 01	3.6964E + 01	6.4780E + 01	1.1952E + 01	5.5014E + 01	1.1118E + 02	8.4814E + 01	2.8465E + 01	5.4439E + 01	2.8811E + 01	8.5095E + 00	9.1259E + 00
	p-value	-	2.8682E-11	1.8663E-10	2.8682E-11	4.9172E-08	2.8682E-11	2.8682E-11	2.8682E-11	4.2820E-11	1.3517E-09	2.8682E-11	1.8040E-02	1.8113E-06
	Rank	1	12	5	11	4	10	13	9	6	7	8	2	3
F_{21}	Mean	2282.6273	2351.5781	2312.7343	2319.0658	2271.5130	2321.4856	2359.4267	2324.1953	2315.7709	2297.1005	2327.4130	2281.9581	2293.1636
	Std	4.6432E + 01	3.2885E + 01	4.4963E + 01	3.7538E + 01	8.8648E + 00	2.8985E + 01	5.0286E + 01	3.5908E + 01	1.2815E + 01	4.3970E + 01	5.5199E + 01	4.5476E + 01	3.7183E + 01
	p-value	-	6.4527E-09	1.7152E-07	3.6128E-08	2.6034E-02	2.8908E-09	1.2436E-07	3.4607E-09	1.5432E-08	3.5735E-05	1.5297E-09	2.5174E-01	3.4769E-01
	Rank	3	12	6	8	1	9	13	10	7	5	11	2	4
F_{22}	Mean	2295.5619	3126.2585	2323.5826	2311.5860	2298.0310	2358.5908	2665.0046	2357.2532	2321.1878	2299.1519	3180.9685	2300.8088	2297.5648
	Std	4.8683E-01	3.2086E + 02	9.8769E + 01	1.4254E + 01	1.6644E + 01	7.3338E + 01	4.3174E + 01	2.1806E + 02	1.0194E + 02	1.4263E + 01	8.7005E + 01	2.1578E + 01	1.6313E + 01
	p-value	-	3.0123E-11	4.1127E-06	8.8720E-10	8.1391E-05	7.3644E-10	3.0123E-11	4.4375E-07	3.5664E-06	4.3760E-01	5.3672E-02	4.8230E-02	6.3511E-02
	Rank	1	12	8	6	3	10	11	9	7	4	13	5	2
20 F_{23}	Mean	2610.8983	2749.1917	2621.2124	2649.0515	2618.7015	2642.4800	2727.6698	2643.9521	2626.3722	2616.9006	2634.9003	2608.9125	2607.8350
	Std	3.8092E + 00	3.1080E + 01	6.5805E + 00	1.6073E + 01	7.9895E + 00	1.5122E + 01	3.2753E + 01	1.1407E + 01	9.9366E + 01	7.6565E + 00	5.9634E + 00	3.5688E + 00	3.3116E + 00
	p-value	-	3.0199E-11	9.0632E-08	3.0199E-11	7.2208E-06	4.0772E-11	3.0199E-11	3.0199E-11	1.6132E-10	9.0307E-04	3.0199E-11	2.3533E-02	1.1738E-03
	Rank	3	13	6	11	5	9	12	10	7	4	8	2	1
F_{24}	Mean	2697.7844	2865.6933	2757.0303	2760.1255	2673.9072	2764.0514	2828.5430	2742.9750	2751.4373	2745.0381	2748.9461	2739.8475	2712.4240
	Std	9.0009E + 01	6.0246E + 01	9.3779E + 00	5.0577E + 01	5.1316E + 00	4.8939E + 01	7.7848E + 01	9.8075E + 01	8.7737E + 00	5.5575E + 00	1.3183E + 01	1.1574E + 02	7.2058E + 01
	p-value	-	3.0199E-11	4.6159E-10	1.5465E-09	3.2553E-01	1.0702E-09	1.1077E-06	1.1937E-06	4.4440E-07	2.1265E-04	1.3832E-02	7.9590E-03	5.5611E-04
	Rank	2	13	9	10	1	11	12	5	8	6	7	4	3
F_{25}	Mean	2926.4858	3404.7122	2934.2836	2941.2886	2928.2892	2959.6159	3173.2108	2923.4435	2929.0830	2930.6880	2942.9745	2927.6769	2927.2995
	Std	2.3467E + 01	2.3271E + 02	3.1430E + 01	2.9363E + 01	2.2450E + 01	4.8012E + 01	2.0961E + 02	8.5462E + 00	2.2628E + 00	2.2833E + 01	3.1475E + 01	2.2555E + 01	2.2401E + 01
	p-value	-	3.0161E-11	2.3240E-02	4.9800E-04	4.5528E-01	1.3013E-03	4.9931E-09	3.4781E-01	2.2822E-01	2.6032E-02	1.4127E-01	6.0473E-01	2.0619E-01
	Rank	2	13	9	10	1	11	12	5	8	6	7	4	3
F_{26}	Mean	2910.5714	4075.3554	3182.0087	3124.0258	2933.3320	3128.4762	3946.7576	3136.7805	3230.6915	2988.3925	3230.7305	2917.1737	2931.3202
	Std	3.8437E + 01	3.4220E + 02	4.5581E + 02	1.9908E + 02	6.3625E + 02	1.4435E + 01	4.9543E + 02	4.8390E + 02	3.8387E + 02	2.2830E + 02	2.2647E + 02	4.0641E + 01	1.6806E + 02
	p-value	-	1.5798E-11	1.2513E-07	2.7751E-05	4.3745E-04	4.3857E-11	1.5798E-11	2.3100E-02	6.6036E-06	3.0002E-04	3.2806E-08	1.2588E-03	1.3563E-02
	Rank	2	13	8	9	5	11	12	1	6	7	10	4	3
F_{27}	Mean	3081.8566	3266.2592	3097.3190	3104.3420	3096.8638	3105.9303	3206.0740	3114.7206	3107.2870	3096.2451	3111.4951	3094.8085	3094.6218
	Std	2.1748E + 00	7.7346E + 01	1.8500E + 01	5.6355E + 00	3.3320E + 00	1.3946E + 01	6.2789E + 02	2.3038E + 01	9.6529E + 00	6.9873E + 01	5.4456E + 00	4.0659E + 00	3.2211E + 00
	p-value	-	3.0199E-11	3.8249E-09	3.3384E-11	1.7769E-10	9.9186E-11	3.0199E-11	4.9752E-11	3.6897E-11	2.0317E-09	7.6779E-02	9.8839E-09	3.1586E-09

(continued on next page)

Table 6 (continued)

Function	Metric	MSAO	AOA	SMA	AO	ARO	GJO	TSA	DO	SO	SAO	CMA-ES	L SHADE-SPACMA	L SHADE-cnEpSin	
F_{28}	Rank	1	13	6	7	5	8	12	11	9	4	10	3	2	
	Mean	3273.9939	3781.8732	3391.4168	3454.2433	3222.8332	3409.2289	3548.8730	3300.9339	3329.9144	3355.8814	3274.3182	3254.4694	3315.8481	
	Std	1.2683E+02	1.6624E+02	1.8809E+02	5.3992E+01	3.5674E+01	1.1533E+02	1.8906E+02	1.5610E+02	9.6621E+01	1.2910E+02	6.0918E+01	1.4818E+02	1.2253E+02	
F_{29}	p-value	-	3.0199E-11	7.9590E-03	3.0199E-11	2.7086E-02	3.9881E-04	9.0632E-08	3.7904E-01	6.7556E-05	3.7154E-04	5.4143E-09	6.6253E-01	7.9590E-03	
	Rank	3	3152.2718	3422.0669	3217.1985	3262.5851	3189.4065	3232.3068	3423.5914	3251.7727	3196.6312	3198.8149	3284.1069	3167.3877	3158.9272
	Mean	1.4713E+01	1.4277E+02	5.9188E+01	5.8772E+01	3.1545E+01	5.7989E+01	1.2202E+01	6.3190E+01	3.0215E+01	5.9241E+01	6.6150E+01	2.2011E+01	1.7567E+01	
F_{30}	p-value	-	3.0199E-11	1.1567E-07	3.6897E-11	5.5999E-07	1.1737E-09	3.0199E-11	1.6947E-09	3.6459E-08	3.0059E-04	1.0937E-10	4.8560E-03	1.3345E-01	
	Rank	1	12	7	10	4	8	13	9	5	6	11	3	2	
	Mean	5085.3727	3.1596E+07	3.9813E+05	1.1916E+06	1.3349E+05	1.7207E+06	1.1522E+07	4.3266E+05	1.3308E+05	3.0410E+05	3.8178E+04	1.5570E+05	1.7237E+05	
F_{31}	Std	1.7586E+03	3.0283E+07	4.9430E+05	1.2362E+06	3.2217E+05	2.2831E+06	1.7443E+07	5.7602E+05	3.5900E+05	4.6970E+05	4.3578E+04	4.0031E+05	4.0335E+05	
	p-value	-	3.0199E-11	8.1527E-11	8.9934E-11	8.1527E-11	7.3891E-11	3.0199E-11	7.6950E-08	4.1825E-09	1.6132E-10	5.3221E-03	1.9112E-02		
	Rank	1	13	8	10	4	11	12	9	3	7	2	5	6	
+/-/	+/-/-	-	29.0/0.0	29.0/0.0	27.0/0.2	29.0/0.0	29.0/0.0	27.0/0.2	28.0/0.1	26.0/0.3	25.0/0.4	24.0/0.5	24.0/0.5		
	Mean rank	1.66	12.24	7.48	9.41	4.31	9.55	12.21	7.90	6.34	5.62	8.72	2.76	2.79	
	Final ranking	1	13	7	10	4	11	12	8	6	5	9	2	3	

fewer iterations on unimodal functions. Meanwhile, the superior local optimum avoidance of MSAO is verified on multimodal and composition functions. In the 3rd column of Fig. 10, the search trajectory of the first dimension is illustrated. From these diagrams, the trajectory of individuals exhibits abrupt and large changes in the early iteration phase, followed by very little fluctuation, which suggests that MSAO is more likely to explore and find higher-quality candidate solutions. The 4th column of Fig. 10 plots the average fitness values of all individuals that characterize the potential of MSAO to evolve candidate solutions during the search process. It is clear that all the curves share a significant decay rate, which indicates that the proposed optimizer can rapidly guide the population to the global optimum. This again demonstrates the excellent convergence speed of MSAO. In the 5th column, a visual overview for the location history of all search agents is available. Looking closely at these search history diagrams, MSAO samples almost the entire search space before exploiting promising regions on a small scale, and ultimately clustering around the global minimum, which amply justifies the good search breadth of MSAO in solving optimization issues. In the last column of Fig. 10, the exploration and exploitation trends of MSAO over the iteration procedure are shown. We determine the percentages of exploration and exploitation separately using the dimension-wise diversity model proposed by Hussain et al. [100] as follows:

$$\text{Exploration}(\%) = \frac{\text{Div}(t)}{\text{Div}_{\max}} \times 100 \quad (23)$$

$$\text{Exploitation}(\%) = \frac{|\text{Div}(t) - \text{Div}_{\max}|}{\text{Div}_{\max}} \times 100 \quad (24)$$

$$\text{Div}(t) = \frac{1}{D} \sum_{j=1}^D \frac{1}{N} \sum_{i=1}^N |\text{median}(x_j(t)) - x_{ij}(t)| \quad (25)$$

where $\text{median}(x_j(t))$ denotes the median of the j -th dimension, and Div_{\max} represents the maximum diversity. In the initial iterations, MSAO retains a high level of exploration ratio, suggesting that it can effectively explore the unknown space. Later, there is a gradual increase in the exploitation percentage, highlighting the strong local exploitation capability of MSAO. Throughout the whole minimization process, MSAO achieves a dynamic balance between exploration and exploitation.

4.2.4. Statistical results analysis

In this subsection, a systematic quantitative investigation for the optimization capability of our proposed optimizer is presented. For MSAO and other twelve intelligent algorithms, namely AOA, SMA, AO, ARO, GJO, TSA, DO, SO, SAO, CMA-ES, L SHADE-SPACMA, and L SHADE-cnEpSin, 30 independent runs are implemented on each of the 10-dimensional test functions $F_1 \sim F_{30}$, and the resulting statistical data can be found in Table 6. The last three rows of the table contain the results of the Wilcoxon rank-sum test (+/-/-), Friedman ranking test ('Mean rank'), and the final rankings for all considered algorithms.

Based on the outcomes from unimodal functions (F_1 and F_3), the mean fitness and standard deviation of MSAO are considerably improved over SAO. MSAO can successfully reach the global optimal solutions for these two test cases, which performs best in all algorithms. To be sure, the searchability of L SHADE-cnEpSin is also noteworthy, as it achieves the same mean value as MSAO on F_3 , but its standard deviation is worse than the latter. Considering that the unimodal functions feature a single extreme solution, it can be concluded that MSAO has a powerful exploitation capability. This is largely attributed to the good point set strategy, which provides MSAO with a high-quality initial population, granting it a notable advantage in the initial iteration. In addition, the DE strategy actively extends the unknown search domains. For multimodal functions ($F_4 \sim F_{10}$), MSAO outperforms the other comparison methods on F_4 and F_{10} . This superiority is evident in both the mean and standard deviation results. On F_6 and F_9 , the performance of CMA-ES is ranked first among all algorithms, followed by MSAO. Similarly, on F_5 ,

F_7 , and F_8 the proposed method performs slightly inferior to LSHADE-SPACMA, obtaining the second place. For hybrid benchmark functions ($F_{11} \sim F_{20}$), MSAO obtains the most satisfactory outcomes on 8 out of 10 test functions, in large part because the DLOBL strategy raises the algorithm's local optimum avoidance. On F_{12} , MSAO demonstrates superior convergence accuracy compared to SAO and other optimization techniques, albeit weaker than LSHADE-SPACMA and LSHADE-cnEpSin. On F_{17} , MSAO is ranked second, lagging behind LSHADE-SPACMA, but the difference between them is not that obvious. For composition functions ($F_{21} \sim F_{30}$), MSAO provides the optimal solutions on F_{22} , F_{26} , F_{27} , F_{29} and F_{30} . On F_{21} , F_{23} and F_{28} , MSAO ranks third. On F_{24} and F_{25} , MSAO ranks second. These results demonstrate the stronger search capability of MSAO in solving complex problems, possibly due that the greedy selection strategy effectively maintains a good exploration-exploitation balance.

Furthermore, based on the outcomes of Wilcoxon rank-sum test, MSAO outperforms AOA on 29 problems, SMA on 29 problems, AO on 29 problems, ARO on 27 problems, GJO on 29 problems, TSA on 29 problems, DO on 27 problems, SO on 28 problems, SAO on 26 problems, CMA-ES on 25 problems, LSHADE-SPACMA on 24 problems, and LSHADE-cnEpSin on 24 problems. Finally, the proposed method attains a Friedman mean ranking of 1.66, ranking first, followed by LSHADE-SPACMA. Fig. 11 illustrates the ranking radar charts of all methods. It can be seen that AOA, AO, GJO, and TSA are roughly in a circle surrounding a larger shaded area, which suggests that they are relatively stable in resolving optimization problems, but the accuracy quality leaves something to be desired. Whereas, the has the smallest shaded area, proving its superior ranking and performance over its peers. Overall, MSAO has a stronger competitive edge on the CEC2017 test suite.

4.2.5. Convergence behavior analysis

The convergence curves of MSAO, AOA, SMA, AO, ARO, GJO, TSA, DO, SO, SAO, CMA-ES, LSHADE-SPACMA, and LSHADE-cnEpSin on 29 CEC2017 benchmark functions are presented in Fig. 12. It can be observed from this figure that, in most test cases, MSAO is capable of converging to the optimal solution in 500 consecutive iterations with little stagnation, which indicates its outstanding exploration and exploitation capabilities. Concerning unimodal functions (F_1 and F_3), the convergence accuracy and speed of MSAO are significantly strengthened compared with SAO. It is notable that LSHADE-SPACMA and LSHADE-cnEpSin also perform competitively, converging slightly faster than MSAO on F_3 . In the case of multimodal functions ($F_4 \sim F_{10}$), MSAO sustains its exceptional convergence rate and local optima avoidance potential. On these seven functions, MSAO achieves the highest convergence precision. Particularly with regard to F_{10} , MSAO initially trails behind SO, but gradually SO converges towards the local optimum, while MSAO continues its journey towards the global optimal point and ultimately achieves superior solution accuracy. For hybrid functions ($F_{11} \sim F_{20}$), MSAO can quickly converse to the optimal position in the initial iterations, demonstrating its high convergence. For composition functions ($F_{21} \sim F_{30}$), MSAO smoothly transitions from exploration to exploitation, meanwhile the convergence accuracy and speed outperform its competitors in varying degrees. Due to the high complexity of composition functions, SAO cannot shed local optima, leading to premature convergence on F_{23} , F_{24} , F_{28} and F_{30} . In contrast, MSAO converges faster and more accurately than SAO. The above results suggest that the strategies introduced in this paper improve the convergence rate and comprehensive search performance of the basic SAO in solving CEC2017 functions while avoiding the local optimum stagnation. The DLOBL strategy accelerates the convergence of MSAO towards the global optimum by updating the optimal solution dimension by dimension. Therefore, it is reasonable to believe that MSAO can effectively provide better convergence patterns than other algorithms

when confronted with complex numerical optimization challenges in the future.

4.2.6. Boxplot behavior analysis

The boxplot is a visual tool for recognizing outliers in the data and thus offer insight into the stability of an algorithm. Fig. 13 presents the boxplots of all algorithms on several iconic CEC2017 problems. In this figure, each box is marked with its centre representing the median, the top and bottom edges indicating the maximum and minimum values, and the presence of a '+' sign signifying the presence of outliers. Compared with SAO and other metaheuristics, MSAO features narrower boxes, lower positions, and smaller median values on almost all test problems. Specifically, MSAO does not generate any outliers, and the convergence results show favorable consistency on F_1 , F_3 , F_{11} , F_{13} , F_{14} , F_{15} , F_{18} , F_{19} , and F_{30} . For F_{10} and F_{29} , MSAO obtains the best median value among all methods. For F_6 , MSAO generates a few outliers, but the overall distribution of median, maximum, and minimum values is more concentrated than its opponents. These experimental results confirm that the multi-strategy improved MSAO has excellent robustness in handling test functions, and its solution accuracy also outperforms SAO. The good point set initialization allows the search agent to traverse the entire solution space more uniformly, and the DE strategy further balances the exploration and exploitation trends to make the algorithm perform its search capability more consistently during the iteration process.

4.2.7. Computational time analysis

To investigate the computational cost of MSAO, Table 7 lists the average computational time for each algorithm after 30 independent operations on 29 CEC2017 benchmark functions. We have counted the total execution time of the thirteen algorithms and ranked them as follows: TSA < AOA < SO < SAO < ARO < GJO < LSHADE-SPACMA < AO < LSHADE-cnEpSin < SMA < DO < MSAO < CMA-ES. It can be found that MSAO requires more computational time compared with SAO, which is mainly due to the introduction of multiple correction strategies that add the algorithm's execution steps and time complexity while boosting the optimization performance. Compared with CMA-ES, MSAO is dominant both in terms of solution accuracy and computation time when faced with the majority of test functions. TSA is the most responsive optimizer with the shortest runtime, but its performance is weak. Taking into account the no-free-lunch theorem and the large amount of time consumption of function evaluation in solving real-world optimization tasks, the proposed MSAO is acceptable by sacrificing some runtime to obtain a more reliable and accurate solution.

4.2.8. Scalability analysis

To comprehend the impact of spatial dimensionality on the performance of the proposed algorithm, a scalability analysis is undertaken in this subsection. The dimension (D) of the 29 CEC2017 benchmark functions is extended from 10 to 30, 50, and 100, respectively. The experimental results of MSAO, AOA, SMA, AO, ARO, GJO, TSA, DO, SO, SAO, CMA-ES, LSHADE-SPACMA, and LSHADE-cnEpSin are presented in Tables 8–10.

According to Table 8, 9 and 10, MSAO obtains the optimal mean fitness and standard deviation on 15 out of 29 30-dimensional benchmark functions (51.72 %), which is higher than LSHADE-cnEpSin (17.24 %), CMA-ES (13.79 %), SAO (6.90 %), LSHADE-SPACMA (6.90 %), SO (3.45 %), and the rest (0 %). For 50-dimensional benchmark functions, MSAO reveals the optimal results on 17 out of 29 (58.62 %), higher than CMA-ES (17.24 %), LSHADE-cnEpSin (13.79 %), SMA (6.90 %), LSHADE-SPACMA (3.45 %), and others (0 %). In addition, for 100-dimensional benchmark functions, MSAO performs best on 14 out of 29 (48.28 %), higher than CMA-ES (20.69 %), SMA (13.79 %), LSHADE-cnEpSin (10.34 %), ARO (3.45 %), TSA (3.45 %), and the

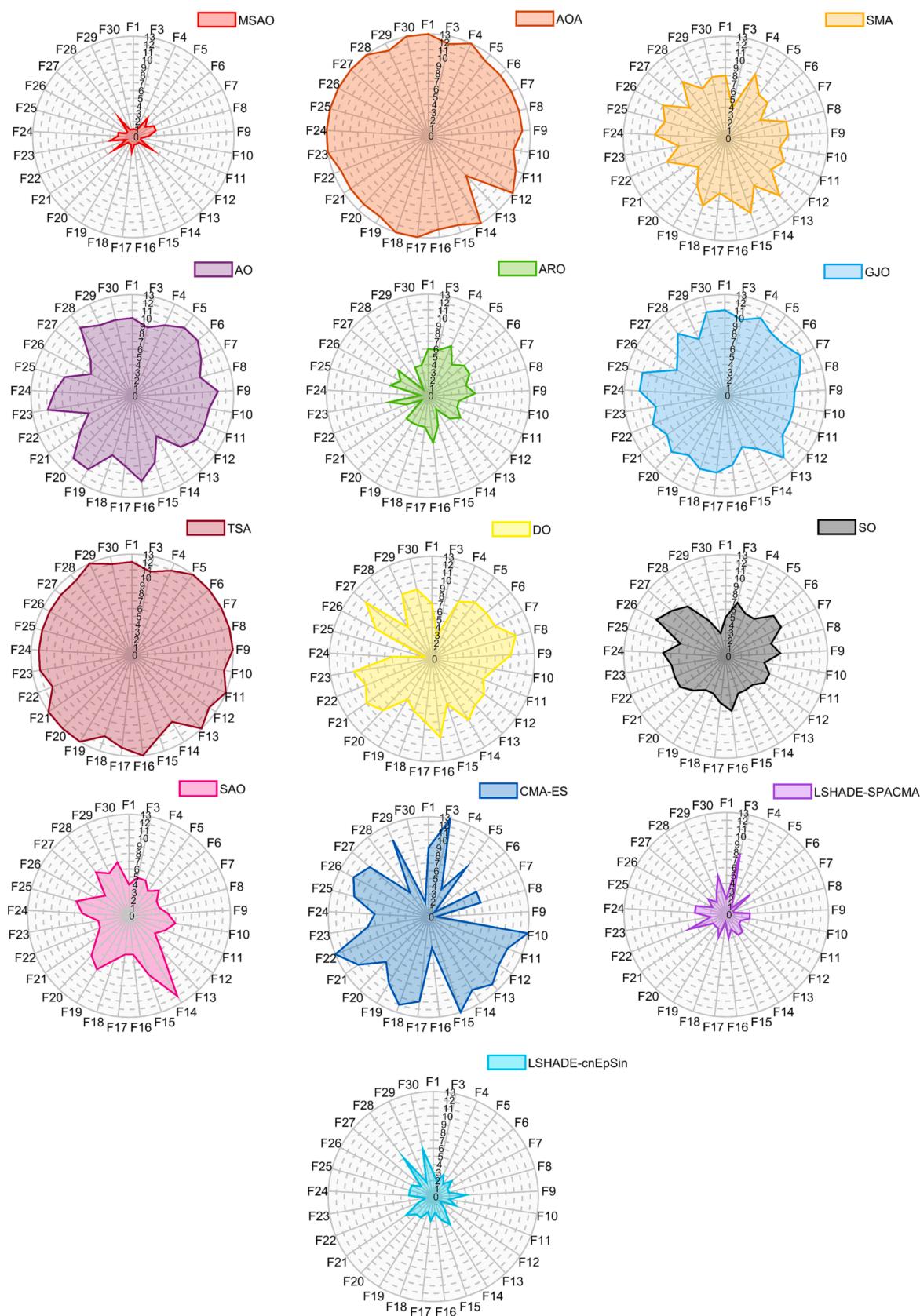


Fig. 11. Radar plots of different optimization algorithms on 29 CEC2017 functions.

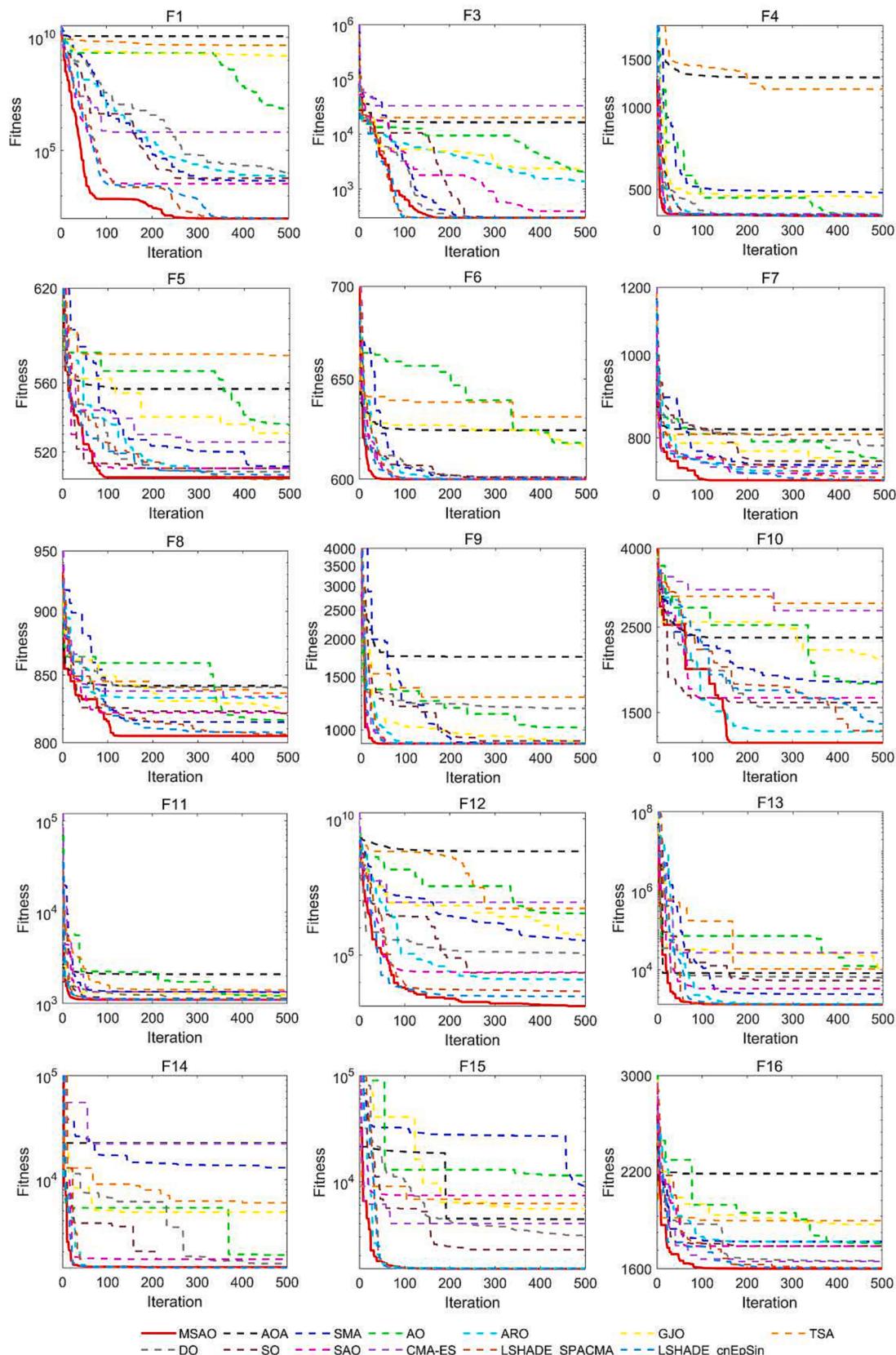


Fig. 12. Convergence curves of different optimization algorithms on 29 CEC2017 functions.

remainder (0 %). With the increase in dimensions, the solution accuracy of all algorithms shows an overall decreasing trend, as larger dimensions require more variables to be optimized. Despite the different

dimensions, MSAO always provides superior results on F_1 , F_5 , F_8 , F_{12} , F_{21} , F_{23} , F_{24} , and F_{29} than other competing methods. On the other hand, the performance of SAO is mediocre. This is primarily because the

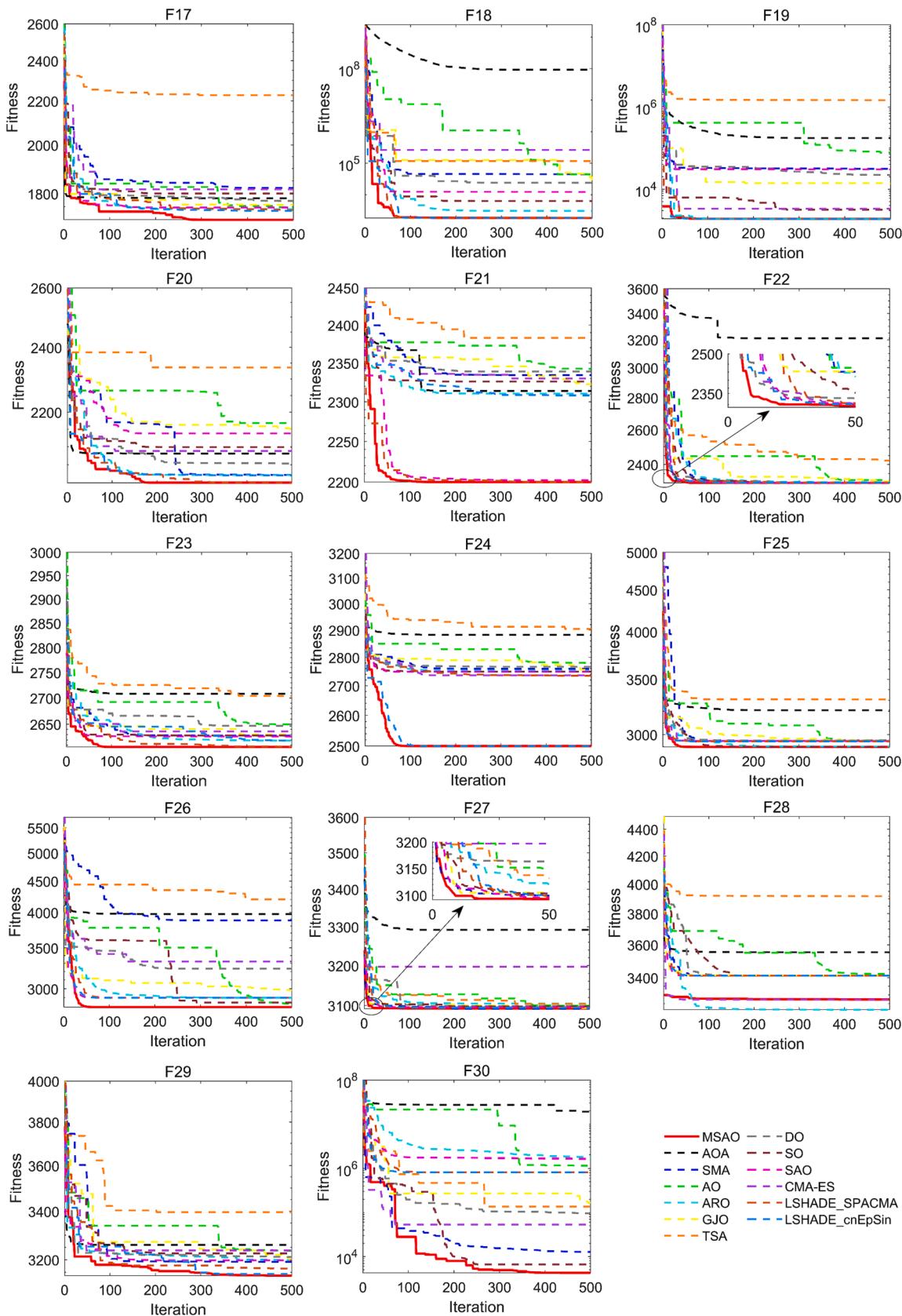
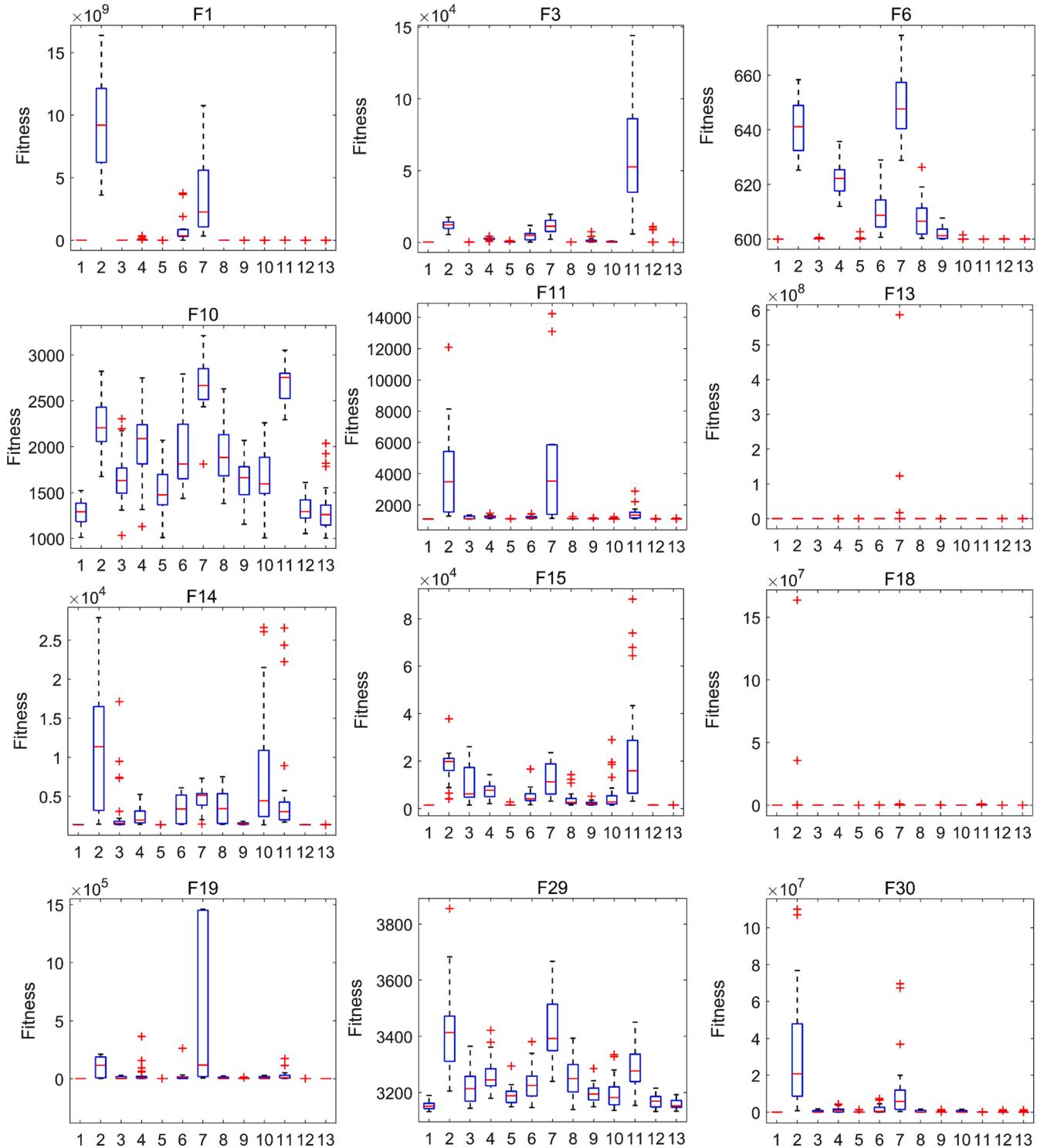


Fig. 12. (continued).

greedy selection strategy better balances exploration and exploitation, and DLOBL enhances the algorithm's local optima avoidance capability in tackling complex large-scale optimization problems. Meanwhile, the

performance of CMA-ES is commendable, as it provides the closest solutions to the theoretical values among all algorithms on F_4 , F_6 , F_9 , and F_{25} . Fig. 14 illustrates the Friedman mean ranking of MSAO and its



1: MSAO 2: AOA 3: SMA 4: AO 5: ARO 6: GJO 7: TSA 8: DO 9: SO 10: SAO 11: CMA-ES 12: LSHADE-SPACMA 13: LSHADE-cnEpSin

Fig. 13. Boxplots of MSAO and other algorithms on some typical CEC2017 functions.

competitors on these scalable functions. The ranking values of MSAO in the three dimensions are 1.79, 1.52, and 2.03 respectively, lower than those of other algorithms. Greedy selection preserves the elite solution in each iteration, which guarantees that the solution accuracy does not deteriorate while solving high-dimensional problems, and the DE strategy compensates for the lack of local exploitation of SAO. MSAO has good portability and can make full use of its exploration and exploitation

potential to discover higher-quality solutions for high-dimensional problems.

4.3. Results comparisons and analysis of the latest CEC2022 benchmark functions

To further verify the novelty and competitiveness of MSAO, five

Table 7

Average computation time of MSAO and other optimization methods on 29 CEC2017 functions (unit: s).

Function	MSAO	AOA	SMA	AO	ARO	GJO	TSA	DO	SO	SAO	CMA-ES	LSHADE-SPACMA	LSHADE-cnEpSin
F_1	3.5706E-01	1.2191E-01	3.1468E-01	2.6117E-01	1.7907E-01	1.9011E-01	1.0618E-01	3.3871E-01	1.2677E-01	1.3224E-01	5.8522E-01	2.6684E-01	2.3308E-01
F_3	2.6362E-01	4.8464E-02	1.5951E-01	1.1801E-01	7.7590E-02	9.0972E-02	4.8462E-02	1.8123E-01	6.1791E-02	7.3618E-02	3.4970E-01	1.4996E-01	1.5537E-01
F_4	2.9894E-01	6.8366E-02	1.6960E-01	9.8102E-02	6.4343E-02	7.4336E-02	3.7314E-02	1.4997E-01	5.0820E-02	6.2154E-02	3.3080E-01	1.4528E-01	1.4377E-01
F_5	1.6377E-01	4.2946E-02	1.2646E-01	8.9013E-02	6.1405E-02	6.7853E-02	4.0215E-02	1.2476E-01	4.7977E-02	5.6287E-02	2.7853E-01	1.3125E-01	1.7015E-01
F_6	2.0554E-01	5.4059E-02	1.4107E-01	1.1824E-01	7.3905E-02	7.6504E-02	5.4248E-02	1.3975E-01	5.9839E-02	5.9640E-02	2.5056E-01	1.0590E-01	1.1321E-01
F_7	1.8626E-01	4.5142E-02	1.2631E-01	1.0320E-01	6.2672E-02	6.6861E-02	3.9188E-02	1.2306E-01	5.0039E-02	5.3606E-02	2.4412E-01	1.1880E-01	1.5756E-01
F_8	1.9540E-01	4.1738E-02	1.3165E-01	1.0438E-01	6.5523E-02	6.8443E-02	4.0552E-02	1.3205E-01	5.0312E-02	6.1345E-02	2.6785E-01	1.3497E-01	1.7657E-01
F_9	1.6196E-01	4.4381E-02	1.3587E-01	1.1105E-01	7.2901E-02	7.8283E-02	4.7963E-02	1.4403E-01	4.5540E-02	5.8242E-02	2.5225E-01	7.2853E-02	7.8144E-02
F_{10}	1.7452E-01	4.8681E-02	1.2948E-01	1.0147E-01	6.6433E-02	6.7698E-02	4.1252E-02	1.2255E-01	4.6700E-02	4.9656E-02	2.3083E-01	1.3075E-01	1.6451E-01
F_{11}	1.5653E-01	4.3594E-02	1.3104E-01	9.8421E-02	6.1754E-02	6.3923E-02	3.8765E-02	1.3073E-01	4.7749E-02	5.1680E-02	2.4399E-01	9.3434E-02	1.1557E-01
F_{12}	1.7443E-01	4.8524E-02	1.4392E-01	9.9689E-02	6.5942E-02	6.7718E-02	3.8917E-02	1.3410E-01	4.6626E-02	4.9887E-02	2.4805E-01	8.8290E-02	8.0743E-02
F_{13}	2.0251E-01	5.2650E-02	1.6173E-01	1.0786E-01	7.0125E-02	7.1837E-02	4.1493E-02	1.3114E-01	4.3270E-02	5.1761E-02	2.3326E-01	8.0415E-02	1.0226E-01
F_{14}	2.0225E-01	5.1188E-02	1.5235E-01	1.0590E-01	6.3161E-02	7.0737E-02	4.0337E-02	1.2590E-01	4.1371E-02	4.3637E-02	2.3299E-01	1.0599E-01	1.4681E-01
F_{15}	2.0318E-01	5.4133E-02	1.5927E-01	1.0748E-01	7.0348E-02	8.1574E-02	4.6343E-02	1.5384E-01	4.6969E-02	4.7245E-02	2.5878E-01	1.0682E-01	1.1095E-01
F_{16}	2.1252E-01	5.5867E-02	1.5811E-01	1.1325E-01	6.9007E-02	8.2135E-02	4.6564E-02	1.5498E-01	5.0367E-02	5.5283E-02	3.1455E-01	1.6880E-01	1.9346E-01
F_{17}	2.2350E-01	6.0914E-02	1.5271E-01	1.3539E-01	7.7403E-02	9.5326E-02	5.6615E-02	1.5463E-01	6.1697E-02	6.3384E-02	3.0885E-01	1.5353E-01	1.7047E-01
F_{18}	2.2438E-01	6.2890E-02	1.8502E-01	1.2236E-01	7.7308E-02	8.3483E-02	4.7176E-02	1.4459E-01	4.7980E-02	4.8399E-02	2.6668E-01	1.2922E-01	1.6259E-01
F_{19}	5.1498E-01	1.3398E-02	2.2292E-01	3.0000E-01	1.9378E-02	2.1326E-01	1.6653E-02	2.6894E-01	1.6139E-01	1.8513E-01	4.4109E-01	2.1043E-01	2.1189E-01
F_{20}	2.7223E-01	6.3090E-02	1.6261E-01	1.5579E-01	9.9583E-02	1.0947E-01	7.3212E-02	2.0069E-01	7.5976E-02	7.5957E-02	3.3147E-01	1.7276E-01	2.0776E-01
F_{21}	2.2663E-01	5.7425E-02	1.3873E-01	1.2053E-01	7.6028E-02	8.2365E-02	5.4453E-02	1.4148E-01	6.0962E-02	6.3024E-02	2.7691E-01	1.4096E-01	1.6943E-01
F_{22}	3.2873E-01	7.5409E-02	1.6649E-01	1.6502E-01	1.0139E-01	1.1286E-01	8.0345E-02	1.9805E-01	8.8768E-02	9.4985E-02	3.2898E-01	1.3007E-01	1.4287E-01
F_{23}	2.7911E-01	7.4591E-02	1.6568E-01	1.6902E-01	1.0223E-01	1.0977E-01	8.0139E-02	1.8409E-01	7.8329E-02	8.4036E-02	3.2280E-01	1.6164E-01	1.7508E-01
F_{24}	2.5929E-01	7.8096E-02	1.7503E-01	1.8795E-01	1.0881E-01	1.1609E-01	8.4885E-02	1.9616E-01	9.5768E-02	9.8065E-02	2.8676E-01	1.1642E-01	1.3693E-01
F_{25}	3.0922E-01	8.1296E-02	1.8244E-01	1.6264E-01	1.0309E-01	1.1456E-01	7.7125E-02	1.8000E-01	7.8251E-02	8.4584E-02	3.3266E-01	1.3351E-01	1.6245E-01
F_{26}	3.1968E-01	8.1031E-02	1.8738E-01	2.0676E-01	1.2350E-01	1.3434E-01	9.6769E-02	2.0510E-01	1.0740E-01	1.2387E-01	3.7218E-01	1.2959E-01	1.1987E-01
F_{27}	3.6506E-01	8.9691E-02	1.9629E-01	2.2264E-01	1.2537E-01	1.4299E-01	1.0276E-01	2.1102E-01	1.0381E-01	1.2147E-01	3.7178E-01	1.3107E-01	1.6214E-01
F_{28}	3.2309E-01	8.4417E-02	1.8600E-01	1.8758E-01	1.1263E-01	1.2239E-01	9.9396E-02	2.2576E-01	9.7910E-02	1.0243E-01	3.3515E-01	1.2570E-01	1.6998E-01
F_{29}	3.0770E-01	9.0082E-02	1.7879E-01	1.8160E-01	1.0266E-01	1.2329E-01	9.9570E-02	2.1416E-01	9.2908E-02	1.0111E-01	3.4896E-01	1.4633E-01	1.7879E-01
F_{30}	5.7772E-01	1.5088E-02	2.6425E-01	3.6319E-01	1.9900E-01	2.2202E-01	1.7638E-01	2.7486E-01	1.6814E-01	1.6971E-01	4.4740E-01	1.7037E-01	1.8935E-01
	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1

modified high-performance optimizers, including enhanced hybrid Arithmetic Optimization Algorithm (CSOAOA) [13], ameliorated Young's double-slit experiment optimizer (IYDSE) [52], Improved hybrid Aquila Optimizer and African Vultures Optimization Algorithm (IHAOA VOA) [67], enhanced Snake Optimizer (ESO) [72], memetic Harris Hawks Optimization (EESHHO) [101], as well as hybrid Slime Mold and Arithmetic Optimization Algorithm with random center learning and restart mutation (RCLSMOA) [102] are used in this section to compare with MSAO on the IEEE CEC2022 test set. The vital parameter configurations for each algorithm can be found in Table 11.

The mean, standard deviation, Wilcoxon rank-sum test p -value, Friedman mean ranking, and final ranking of each algorithm obtained

from 30 independent runs on the 12 IEEE CEC2022 benchmark functions are tabulated in Table 12. From this table, MSAO achieves the optimal solution accuracy and standard deviation on 10 out of 12 test problems ($F_{31} \sim F_{39}, F_{42}$), whereas IYDSE and CSOAOA reveal the optimal results on F_{40} and F_{41} , respectively, which demonstrates that the proposed MSAO has a very competitive search capability over other comparison methods. Statistically, MSAO significantly outperforms CSOAOA on 11 functions, IYDSE on 12 functions, IHAOA VOA on 11 functions, ESO on 10 functions, EESHHO on 11 functions, and RCLSMOA on 12 functions. In the Friedman ranking test, the mean ranking value of MSAO is 1.25, which ranks first among all algorithms, followed by ESO and IYDSE. These results show that MSAO improved by multiple strategies

Table 8

Comparison of MSAO with different algorithms on 30-dimensional CEC2017 functions.

Function	Metric	MSAO	AOA	SMA	AO	ARO	GJO	TSA	DO	SO	SAO	CMA-ES	LSHADE-SPACMA	LSHADE-cnEpSin
F_1	Mean	1.0278E + 03	5.8245E + 10	2.6663E + 04	3.7415E + 09	1.8920E + 08	1.2270E + 10	3.3302E + 10	9.8808E + 05	1.6426E + 07	2.9714E + 04	2.4437E + 04	3.0781E + 04	3.5443E + 04
	Std	7.8891E + 02	7.8728E + 09	1.1200E + 04	1.1811E + 04	2.3673E + 08	4.3699E + 09	8.1878E + 09	7.2474E + 05	2.2222E + 05	2.9492E + 04	3.8385E + 04	4.0631E + 04	3.7926E + 04
F_3	Mean	4.4436E + 04	8.0711E + 04	1.2838E + 05	6.8629E + 04	6.3968E + 04	6.3828E + 04	8.4865E + 04	3.7170E + 04	7.0537E + 04	1.2866E + 05	4.8360E + 05	4.9992E + 04	3.0513E + 04
	Std	1.5833E + 04	7.4795E + 03	3.4898E + 04	8.3831E + 03	7.4596E + 03	7.5158E + 03	1.8321E + 04	1.1580E + 04	1.1546E + 04	3.3546E + 04	1.8248E + 05	5.5717E + 04	7.3291E + 03
F_4	Mean	4.8754E + 02	1.4646E + 04	5.0574E + 02	1.1823E + 03	5.9179E + 02	1.4968E + 03	7.8774E + 02	5.1401E + 02	5.7367E + 02	5.0189E + 02	4.2287E + 02	5.0746E + 02	5.0207E + 02
	Std	1.8795E + 01	3.6005E + 03	2.5126E + 01	2.6275E + 01	4.8096E + 02	8.3605E + 02	2.7789E + 03	1.8110E + 02	4.3186E + 01	2.1609E + 01	8.0605E-01	2.9219E + 01	2.1220E + 01
F_5	Mean	5.5658E + 02	8.9624E + 02	6.3820E + 02	7.2548E + 02	6.4623E + 02	7.2009E + 02	8.7039E + 02	6.7861E + 02	5.9167E + 02	5.8847E + 02	6.0809E + 02	5.7354E + 02	5.9099E + 02
	Std	1.3755E + 01	3.8537E + 01	4.0433E + 01	3.2021E + 01	3.4616E + 01	5.9121E + 01	3.1185E + 01	3.9345E + 01	2.0651E + 01	4.2613E + 01	6.7806E + 01	3.2066E + 01	2.6102E + 01
F_6	Mean	6.0016E + 02	6.8051E + 02	6.1719E + 02	6.5650E + 02	6.1683E + 02	6.4186E + 02	6.8351E + 02	6.4687E + 02	6.2039E + 02	6.0112E + 02	6.0002E + 02	6.0203E + 02	6.0441E + 02
	Std	2.3628E-01	7.5971E + 00	1.0482E + 01	1.1338E + 01	7.1628E + 01	9.1069E + 01	5.0910E + 01	1.0346E + 01	7.3692E + 00	8.4781E-01	7.4086E-03	1.6994E + 00	2.1591E + 00
F_7	Mean	8.2390E + 02	1.4032E + 03	9.1036E + 02	1.1554E + 03	9.5800E + 02	1.0780E + 03	1.3756E + 03	1.0331E + 03	9.1592E + 02	9.7137E + 02	9.0959E + 02	8.2578E + 02	8.3870E + 02
	Std	1.9991E + 01	6.0318E + 01	5.4311E + 01	6.3629E + 01	6.7085E + 01	6.3980E + 01	5.8898E + 01	7.6907E + 01	4.3351E + 01	2.6183E + 01	5.4238E + 01	2.0808E + 01	2.6426E + 01
28 F_8	Mean	8.6914E + 02	1.1193E + 03	9.3256E + 02	9.9269E + 02	9.1003E + 02	9.9142E + 02	1.0947E + 03	9.6247E + 02	8.9537E + 02	8.8392E + 02	8.9626E + 02	8.7687E + 02	8.7531E + 02
	Std	1.4295E + 01	3.9064E + 01	2.4917E + 01	2.5721E + 01	2.2854E + 02	4.2730E + 02	2.5185E + 03	4.1739E + 02	1.7660E + 02	4.1350E + 01	6.8292E + 01	1.5168E + 01	3.9281E + 01
F_9	Mean	9.3080E + 02	8.0177E + 03	4.9268E + 03	7.8417E + 03	3.1915E + 03	6.2581E + 03	1.0779E + 04	5.5967E + 03	2.3607E + 03	1.0456E + 04	9.0000E + 02	1.3437E + 03	1.3327E + 03
	Std	3.4984E + 01	1.0292E + 03	1.5535E + 03	1.6729E + 03	7.7237E + 03	1.8823E + 03	1.2596E + 03	2.0092E + 03	9.6806E + 03	4.4566E + 03	1.6393E-02	4.6667E + 02	3.9520E + 02
F_{10}	Mean	4.4710E + 03	8.0312E + 03	5.0382E + 03	6.1995E + 03	7.4993E + 03	7.4201E + 03	8.6119E + 03	5.1955E + 03	4.1883E + 03	4.3357E + 03	8.9341E + 03	5.0275E + 03	5.0065E + 03
	Std	5.9578E + 02	4.7739E + 02	7.3850E + 02	6.5808E + 02	1.2442E + 03	1.4998E + 03	8.4348E + 03	7.1992E + 02	2.8029E + 02	9.8132E + 02	3.3553E + 02	8.0724E + 02	8.3472E + 02
F_{11}	Mean	1.1468E + 03	1.0796E + 04	1.2720E + 03	4.2156E + 03	1.4451E + 03	3.9905E + 03	8.1003E + 03	1.2370E + 03	1.4565E + 03	1.2598E + 03	1.3769E + 03	1.2739E + 03	1.2842E + 03
	Std	3.2115E + 01	4.1332E + 03	5.4913E + 01	1.5867E + 03	1.6713E + 02	1.6630E + 03	2.2748E + 03	4.4526E + 02	1.2787E + 02	7.8101E + 02	1.1126E + 01	8.3291E + 01	6.3732E + 01
F_{12}	Mean	1.0561E + 05	1.4530E + 10	3.6252E + 06	3.6404E + 08	8.7786E + 08	9.7825E + 08	6.1875E + 09	7.5133E + 09	3.3871E + 06	1.0748E + 06	8.2109E + 05	4.1638E + 05	6.7634E + 05
	Std	9.6341E + 04	4.1210E + 09	2.7410E + 06	3.2561E + 08	1.1557E + 04	9.4051E + 08	3.6771E + 09	7.1964E + 06	4.1029E + 06	1.0673E + 06	5.8688E + 05	4.1142E + 05	8.4051E + 05
F_{13}	Mean	1.6964E + 04	1.5911E + 10	4.8975E + 04	1.1563E + 07	2.4704E + 04	3.8265E + 08	5.1953E + 09	1.3059E + 05	4.4700E + 04	2.1668E + 04	5.1266E + 04	1.8737E + 04	1.3900E + 04
	Std	1.7173E + 04	6.1482E + 09	2.8530E + 04	1.4167E + 07	1.7186E + 04	7.6144E + 08	3.4769E + 09	1.9424E + 05	2.3132E + 05	1.7987E + 04	3.7727E + 06	1.1126E + 04	1.0770E + 04
F_{14}	Mean	1.5739E + 03	4.1106E + 06	2.4711E + 05	1.1371E + 06	8.7168E + 04	1.0176E + 05	2.6634E + 05	1.5233E + 05	1.1689E + 05	4.0150E + 04	4.0397E + 05	1.5917E + 03	1.6026E + 03
	Std	4.6475E + 01	5.8001E + 06	2.0747E + 05	1.1029E + 06	1.0526E + 05	8.9300E + 05	2.5771E + 06	1.8102E + 05	2.1429E + 05	4.0578E + 05	2.5471E + 06	6.6767E + 01	7.6420E + 01

(continued on next page)

Table 8 (continued)

Function	Metric	MSAO	AOA	SMA	AO	ARO	GJO	TSA	DO	SO	SAO	CMA-ES	L SHADE-SPACMA	L SHADE-cnEpSin
F_{15}	Mean	5.1738E + 03	4.6708E + 08	2.4577E + 04	2.1903E + 05	6.6290E + 03	1.3020E + 07	3.2340E + 08	5.3217E + 04	1.6275E + 04	6.2738E + 03	5.7387E + 06	2.8163E + 03	2.6579E + 03
	Std	5.2356E + 03	6.4366E + 08	1.5046E + 04	1.4489E + 05	3.9051E + 03	2.8036E + 07	6.2136E + 08	3.6250E + 04	1.4896E + 04	6.7978E + 03	4.6615E + 06	1.6492E + 03	6.8422E + 02
F_{16}	Mean	2.3333E + 03	5.5720E + 03	2.6477E + 03	3.3173E + 03	2.6298E + 03	3.2192E + 03	4.3092E + 03	2.8574E + 03	2.6683E + 03	2.4481E + 03	2.9102E + 03	2.5319E + 03	2.3525E + 03
	Std	1.8353E + 02	1.1291E + 03	2.9616E + 02	4.2832E + 02	2.3816E + 02	4.3108E + 02	6.4989E + 02	3.4861E + 02	2.7507E + 02	2.8804E + 02	2.4524E + 02	2.4040E + 02	1.9047E + 02
F_{17}	Mean	1.8659E + 03	4.9123E + 03	2.3120E + 03	2.4590E + 03	2.0803E + 03	2.2299E + 03	3.4595E + 03	2.3992E + 03	2.2700E + 03	2.0886E + 03	2.3255E + 03	2.0015E + 03	1.9502E + 03
	Std	9.6717E + 01	3.2797E + 03	2.5951E + 02	2.4992E + 02	2.0224E + 02	2.7379E + 02	1.7059E + 03	2.2272E + 02	1.6835E + 02	1.9783E + 02	1.9859E + 02	1.1659E + 02	1.0959E + 02
F_{18}	Mean	1.8369E + 05	3.9312E + 07	3.3949E + 06	7.0934E + 05	4.1133E + 06	4.5021E + 06	3.9199E + 07	2.0671E + 06	9.8920E + 05	1.0144E + 06	5.8721E + 06	5.0115E + 04	3.8515E + 04
	Std	1.5993E + 05	2.9066E + 07	3.4230E + 06	5.9567E + 06	4.7858E + 07	1.1889E + 06	4.3776E + 07	1.7502E + 06	9.2168E + 05	9.6281E + 05	4.0932E + 06	5.4556E + 04	3.1602E + 04
F_{19}	Mean	5.1414E + 03	4.5855E + 08	2.0718E + 04	2.9587E + 06	6.9691E + 05	1.7050E + 07	6.6086E + 04	1.2916E + 07	1.8860E + 04	5.1964E + 04	1.6207E + 07	2.8929E + 03	2.3308E + 03
	Std	3.1570E + 03	5.5813E + 08	2.0494E + 04	3.1850E + 05	5.2087E + 05	3.6729E + 06	8.8935E + 05	1.1631E + 06	2.4234E + 05	3.0933E + 05	1.0975E + 07	2.9140E + 03	5.8070E + 02
F_{20}	Mean	2.1825E + 03	2.8420E + 03	2.6691E + 03	2.6634E + 03	2.4521E + 03	2.6423E + 03	3.0674E + 03	2.6009E + 03	2.5149E + 03	2.3848E + 03	2.6686E + 03	2.4415E + 03	2.2818E + 03
	Std	9.8598E + 01	1.8921E + 02	1.4080E + 02	1.7148E + 03	1.6257E + 02	1.6609E + 02	3.0240E + 02	2.6805E + 02	1.8266E + 02	2.0475E + 02	1.6491E + 02	1.2061E + 02	1.1493E + 02
F_{21}	Mean	2.3580E + 03	2.6964E + 03	2.4454E + 03	2.5118E + 03	2.4062E + 03	2.4871E + 03	2.6761E + 03	2.4695E + 03	2.4049E + 03	2.3789E + 03	2.3986E + 03	2.3730E + 03	2.3830E + 03
	Std	1.3059E + 01	5.1988E + 01	3.9393E + 01	2.7764E + 01	2.9612E + 01	3.8425E + 01	4.7479E + 01	4.1673E + 01	1.8405E + 01	3.5279E + 01	6.4728E + 01	1.3174E + 01	1.4228E + 01
F_{22}	Mean	3.4696E + 03	9.2599E + 03	5.6675E + 03	6.8279E + 03	2.4196E + 03	6.3569E + 03	9.5118E + 03	5.7033E + 03	4.3266E + 03	3.6052E + 03	1.0264E + 03	2.3134E + 03	3.3229E + 03
	Std	5.1847E + 02	7.1844E + 02	1.5577E + 03	3.3486E + 03	8.2942E + 03	2.3749E + 01	1.2131E + 03	2.0180E + 03	2.1108E + 03	1.6645E + 03	2.7732E + 02	3.2176E + 01	1.8805E + 03
F_{23}	Mean	2.6987E + 03	3.5310E + 03	2.7661E + 03	3.0263E + 03	2.8089E + 03	2.9114E + 03	3.3838E + 03	2.8952E + 03	2.8047E + 03	2.7238E + 03	2.7128E + 03	2.7340E + 03	2.7404E + 03
	Std	1.6690E + 01	1.5680E + 02	2.7553E + 01	6.6956E + 01	4.2939E + 01	5.6926E + 01	2.0046E + 02	3.8282E + 01	4.6464E + 01	1.9905E + 01	4.9822E + 01	2.1137E + 01	2.3198E + 01
F_{24}	Mean	2.8820E + 03	3.8811E + 03	2.9590E + 03	3.1394E + 03	2.9768E + 03	3.0899E + 03	3.5314E + 03	3.0760E + 03	2.9522E + 03	2.8959E + 03	2.8903E + 03	2.9006E + 03	2.9087E + 03
	Std	1.4882E + 01	2.0963E + 02	4.2001E + 01	5.3084E + 01	4.5213E + 01	6.2621E + 01	1.2613E + 02	6.2443E + 01	3.6105E + 01	1.5411E + 01	6.0564E + 01	1.8495E + 01	3.2693E + 01
F_{25}	Mean	2.8836E + 03	5.8849E + 03	2.9027E + 03	3.1001E + 03	2.9911E + 03	3.2254E + 03	4.2025E + 03	2.9140E + 03	2.9502E + 03	2.8890E + 03	2.8784E + 03	2.8954E + 03	2.9036E + 03
	Std	1.0727E + 01	8.6709E + 02	1.8397E + 01	7.1461E + 01	3.4897E + 01	1.7513E + 01	5.3444E + 01	1.8442E + 02	3.5896E + 01	4.5003E + 01	1.1023E-01	1.2062E + 01	2.1798E + 01
F_{26}	Mean	4.1700E + 03	1.0864E + 04	5.0035E + 03	6.4077E + 03	5.0724E + 03	6.0647E + 03	9.6711E + 03	6.1129E + 03	5.5230E + 03	4.2842E + 03	4.2557E + 03	4.1033E + 03	4.5665E + 03
	Std	6.0188E + 02	8.1174E + 02	3.7164E + 03	1.1830E + 03	1.0196E + 03	6.4744E + 02	9.8226E + 02	1.0773E + 03	4.5053E + 02	4.7240E + 02	6.0205E + 02	1.8843E + 02	4.7785E + 02
F_{27}	Mean	3.2000E + 03	4.6157E + 03	3.2389E + 03	3.4620E + 03	3.2798E + 03	3.3785E + 03	3.8956E + 03	3.2912E + 03	3.3060E + 03	3.2236E + 03	3.2000E + 03	3.2379E + 03	3.2283E + 03
	Std	1.2107E-05	3.6369E + 02	1.7447E + 01	9.2033E + 01	3.0754E + 01	6.2832E + 01	4.3683E + 01	5.1991E + 01	3.8584E + 01	1.6639E + 01	6.5918E-05	1.9762E + 01	1.4616E + 01
F_{28}	Mean	3.2640E + 03	7.0781E + 03	3.2702E + 03	3.8039E + 03	3.3790E + 03	3.9513E + 03	5.2166E + 03	3.2969E + 03	3.3566E + 03	3.2442E + 03	3.3000E + 03	3.2462E + 03	3.2511E + 03
	Std	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03

(continued on next page)

Table 8 (continued)

Function	Metric	MSAO	AOA	SMA	AO	ARO	GJO	TSA	DO	SO	SAO	CMA-ES	L SHADE-SPACMA	L SHADE-cmEPStn
F_{29}	Std	2.5019E +	7.5999E +	5.0504E +	1.5103E +	6.1745E +	3.2491E +	7.0715E +	2.4225E +	5.8538E +	1.1863E +	6.8575E +	2.6947E + 01	3.4152E + 01
	Mean	01	02	01	02	01	02	02	01	01	01	01	01	3.6715E + 03
F_{30}	Std	3.4864E +	7.8881E +	4.0955E +	4.8905E +	3.8722E +	4.4714E +	6.0566E +	4.1246E +	4.0601E +	3.7401E +	3.9908E +	3.6417E + 03	3.6417E + 03
	Mean	03	03	03	03	03	03	03	03	03	03	03	03	3.6715E + 03

also has exceptional proficiency in addressing intricate numerical optimization challenges. Fig. 15 illustrates the ranking radar charts of MSAO and other six improved algorithms on 12 CEC2022 functions. Compared with competitor methods, the radar plot of MSAO has the smallest shadow area, which once again proves the its excellent optimization performance and overall ranking.

Fig. 16 presents the convergence curves of MSAO, CSOAOA, IYDSE, IHAAVOA, ESO, EESHHO, and RCLSMAOA on the IEEE CEC2022 test suite. It can be found that MSAO tends to converge faster and obtain higher accuracy solutions in most test cases. On F_{31} , MSAO, ESO, and IYDSE converge to the same value, but the curve of MSAO has the greatest decreasing speed. For multimodal functions ($F_{32} \sim F_{35}$), MSAO also possesses good convergence performance. In particular, when the rest methods are stuck in local optimum stagnation on F_{34} , MSAO still continues to converge at later stages. For hybrid and composition functions ($F_{36} \sim F_{42}$), MSAO is able to switch smoothly between exploration and exploitation, and its convergence precision and speed outperform comparison algorithms to varying degrees. Fig. 17 depicts the boxplots of each algorithm on different benchmark functions. The overall distribution of mean, maximum, and minimum values of MSAO is more centralized than other comparison algorithms. On F_{31} , F_{32} , F_{36} , F_{37} , F_{38} and F_{40} , MSAO generates no outliers. It can be concluded that the convergence speed and stability of MSAO are highly competitive even in the face of more complicated optimization problems.

The average computational time of MSAO and these algorithms for 30 independent runs to solve each CEC2022 function is provided in Table 13. Similarly, we have counted the total runtime of all optimization methods and ranked them as follows: EESHHO < IYDSE < ESO < CSOAOA < MSAO < IHAAVOA < RCLSMAOA. It can be seen that MSAO obtains the third last place, requiring a little more computational time than EESHHO, IYDSE, ESO, and CSOAOA, but the local optimum avoidance ability of MSAO and its provided solution quality are better. Hence, MSAO is worth promoting, and it is particularly suitable for optimization scenarios where solution accuracy is a key concern.

In this section, the optimization performance of MSAO is comprehensively evaluated on 29 CEC2017 and 12 CEC2022 benchmark functions, where MSAO outperforms other advanced algorithms with respect to convergence speed, solution accuracy, and robustness in the vast majority of test cases. Each improvement strategy enhances the exploitation & exploration abilities of MSAO to some extent. Good point set initialization provides the high-quality initial population for global search, greedy selection and DE strategies effectively overcome local optimum stagnation for better solution accuracy, and DLOBL accelerates convergence. The promising performance of MSAO in numerical experiments gives us reason to believe that it has the potential to address complex nonlinear optimization problems, and in the next section, the proposed method will be applied to real-world projects.

5. MSAO for practical engineering optimization challenges

In this section, MSAO is implemented to tackle seven real-world engineering optimization challenges to assess its effectiveness and superiority in practical applications. The first six optimization tasks fall within the realm of industrial engineering design problems, encompassing pressure vessel design, step-cone pulley design, cantilever beam design, speed reducer design, robot gripper design, and rolling element bearing design. Since there are several equality constraints and inequality constraints in these problems, it is convenient to use the penalty function method [11] to exert penalties on candidate solutions far away from the feasible domain, thus transforming constrained optimization into unconstrained optimization and decreasing the computational cost. In addition, a more pivotal and valuable problem of parameter identification for PV models is considered. Likewise,

MSAO and sixteen other well-organized MAs are executed independently 30 times for each optimization assignment with the population size of $N = 30$ and the maximum iteration count of $t_{\max} = 500$. The

Table 9

Comparison of MSAO with different algorithms on 50-dimensional CEC2017 functions.

Function	Metric	MSAO	AOA	SMA	AO	ARO	GJO	TSA	DO	SO	SAO	CMA-ES	LSHADE-SPACMA	LSHADE-cnEpSin
F_1	Mean	2.3046E + 03	1.1132E + 11	2.3381E + 06	2.0700E + 10	7.1336E + 09	3.6773E + 10	7.9784E + 07	1.9230E + 07	8.7747E + 08	7.3030E + 07	1.2778E + 05	6.9271E + 07	4.4032E + 07
	Std	2.6287E + 03	9.3106E + 09	9.6295E + 05	4.2688E + 09	2.7053E + 09	5.4662E + 09	9.5998E + 09	1.0633E + 07	5.5078E + 08	3.2684E + 07	5.7760E + 04	6.2752E + 07	2.6860E + 07
F_3	Mean	1.4859E + 05	1.8051E + 05	2.0494E + 05	3.3029E + 05	1.6294E + 05	1.5121E + 05	1.7804E + 05	2.9155E + 05	1.6891E + 05	3.4207E + 05	9.1975E + 05	1.5935E + 05	1.0865E + 05
	Std	8.6234E + 04	1.7338E + 04	8.4944E + 04	8.8994E + 04	2.0262E + 04	1.7354E + 04	4.8106E + 04	6.2993E + 04	1.9469E + 04	9.3855E + 04	3.1766E + 05	2.8744E + 04	1.3649E + 04
F_4	Mean	5.3225E + 02	3.4673E + 04	6.1013E + 02	3.8957E + 03	1.4182E + 03	5.3879E + 03	1.6293E + 04	6.0079E + 02	8.8815E + 02	6.1644E + 02	4.5088E + 02	6.7764E + 02	6.4588E + 02
	Std	5.0313E + 01	8.6837E + 03	3.9959E + 01	8.8757E + 02	3.2426E + 02	2.1029E + 03	4.9743E + 03	3.9398E + 01	1.2409E + 02	3.7557E + 01	1.5983E + 01	5.8108E + 01	5.9555E + 01
F_5	Mean	6.6734E + 02	1.1636E + 03	8.2110E + 02	9.6560E + 02	8.0548E + 02	9.2153E + 02	1.1217E + 03	8.3798E + 02	7.1713E + 02	8.6141E + 02	8.5249E + 02	7.1631E + 02	7.1285E + 02
	Std	2.7937E + 01	3.6061E + 01	5.0519E + 01	4.6129E + 01	4.2928E + 01	6.8896E + 01	9.7212E + 01	6.1310E + 01	4.2990E + 01	8.3639E + 01	2.8415E + 01	3.7801E + 01	4.3308E + 01
F_6	Mean	6.0064E + 02	6.9498E + 02	6.4280E + 02	6.7647E + 02	6.3701E + 02	6.5818E + 02	6.9835E + 02	6.5752E + 02	6.3202E + 02	6.1044E + 02	6.0036E + 02	6.0724E + 02	6.1677E + 02
	Std	4.6086E-01	7.6257E + 00	1.2806E + 01	7.1486E + 00	7.1823E + 00	9.7011E + 00	7.1046E + 00	8.7068E + 00	5.2402E + 00	4.1137E + 00	1.1609E-01	3.0638E + 00	6.4078E + 00
F_7	Mean	1.0332E + 03	1.9862E + 03	1.1760E + 03	1.6639E + 03	1.3593E + 03	1.4572E + 03	1.9505E + 03	1.3791E + 03	1.2020E + 03	1.2856E + 03	1.1194E + 03	1.0948E + 03	1.1345E + 03
	Std	1.2601E + 01	6.3850E + 01	1.1347E + 01	8.6619E + 01	1.3294E + 01	8.3537E + 01	5.8916E + 01	1.1187E + 01	6.5639E + 01	4.3119E + 01	1.8868E + 01	6.5090E + 01	8.7484E + 01
F_8	Mean	9.8371E + 02	1.5055E + 03	1.1067E + 03	1.2621E + 03	1.1495E + 03	1.2313E + 03	1.4474E + 03	1.1332E + 03	1.0069E + 03	1.1419E + 03	1.1437E + 03	1.0180E + 03	1.0441E + 03
	Std	2.8587E + 01	3.9006E + 01	4.9155E + 01	4.1179E + 01	5.0804E + 01	4.7546E + 01	3.7697E + 01	4.6700E + 01	3.0361E + 01	9.5289E + 01	5.4614E + 01	1.0174E + 02	4.0407E + 01
F_9	Mean	1.5662E + 03	3.2232E + 04	1.5684E + 04	2.8291E + 04	1.2452E + 04	2.2361E + 04	3.7554E + 04	1.7377E + 04	6.8596E + 03	3.3665E + 03	9.0133E + 02	4.9195E + 03	5.9803E + 03
	Std	8.5886E + 02	4.2466E + 03	4.7693E + 03	4.5365E + 03	3.0511E + 03	6.4340E + 03	2.6047E + 03	4.9175E + 03	2.5845E + 03	9.2235E + 03	3.9941E-01	1.9265E + 03	2.7346E + 03
F_{10}	Mean	8.4706E + 03	1.3937E + 04	8.0547E + 03	1.0880E + 04	1.4424E + 04	1.1444E + 04	1.4666E + 04	8.4800E + 04	9.6823E + 03	1.0403E + 04	1.5891E + 04	9.4140E + 03	9.0458E + 03
	Std	8.3463E + 02	7.0496E + 02	6.8304E + 02	1.1278E + 03	1.1683E + 03	2.3622E + 03	9.6664E + 03	1.0511E + 03	2.5073E + 03	2.9742E + 03	1.1189E + 03	1.7199E + 03	1.0928E + 03
F_{11}	Mean	1.3205E + 03	2.4377E + 04	1.4895E + 03	6.8166E + 03	3.4903E + 03	1.0756E + 04	2.2095E + 04	1.4584E + 04	3.4348E + 03	3.1007E + 03	1.2466E + 05	2.6584E + 03	1.5928E + 03
	Std	8.7062E + 01	4.4246E + 03	1.2304E + 02	1.3607E + 03	1.1570E + 03	3.1205E + 03	5.1843E + 03	1.5351E + 03	8.7572E + 02	1.6564E + 03	3.1626E + 03	3.6315E + 03	2.5674E + 02
F_{12}	Mean	1.9153E + 06	7.7359E + 10	3.0942E + 07	6.3459E + 09	2.2991E + 08	1.0415E + 10	3.5598E + 07	6.6381E + 07	4.5319E + 07	1.7750E + 07	1.1501E + 08	1.2154E + 07	1.2363E + 07
	Std	1.0890E + 06	1.4032E + 10	1.4563E + 07	2.8705E + 08	1.6630E + 09	4.8632E + 09	9.3691E + 08	3.9573E + 09	2.4673E + 07	9.8605E + 05	6.7510E + 06	7.5426E + 06	6.6435E + 06
F_{13}	Mean	5.0894E + 03	4.5820E + 10	1.1619E + 05	1.2653E + 09	1.7910E + 06	2.7617E + 09	1.9523E + 10	2.3978E + 05	1.0829E + 06	8.0707E + 03	2.8143E + 04	1.9956E + 04	3.3900E + 04
	Std	4.6264E + 03	1.2565E + 10	7.3573E + 04	8.8140E + 08	1.4026E + 06	3.6584E + 09	1.2920E + 10	1.4801E + 05	2.2601E + 06	5.7329E + 03	1.2395E + 03	7.3185E + 03	1.9189E + 04
F_{14}	Mean	6.7807E + 04	9.4441E + 07	9.9221E + 05	9.2477E + 06	1.2102E + 06	2.4733E + 08	3.0133E + 07	6.2727E + 07	7.4146E + 05	2.3166E + 06	4.0744E + 04	1.7889E + 04	1.2250E + 04
	Std	6.3461E + 04	6.6872E + 07	5.7243E + 05	7.7998E + 06	1.1870E + 06	2.2168E + 06	3.0230E + 07	3.7743E + 05	1.1307E + 06	1.9104E + 05	2.0360E + 04	1.4068E + 04	1.3661E + 04

(continued on next page)

Table 9 (continued)

Function	Metric	MSAO	AOA	SMA	AO	ARO	GJO	TSA	DO	SO	SAO	CMA-ES	LSHADE-SPACMA	LSHADE-cnEpSin
F_{15}	Mean	1.0270E + 04	8.8327E + 09	3.0207E + 04	2.5880E + 07	2.7128E + 04	3.1299E + 08	3.0764E + 09	5.9247E + 04	4.3124E + 04	1.3248E + 04	2.6407E + 04	1.0145E + 04	1.2005E + 04
	Std	6.0832E + 03	3.4777E + 09	1.6937E + 04	4.6622E + 07	1.9323E + 04	4.5125E + 08	2.1000E + 09	3.4353E + 04	3.0071E + 04	5.9839E + 03	1.5572E + 04	4.3308E + 03	5.5098E + 03
F_{16}	Mean	3.3655E + 03	7.8798E + 03	3.8832E + 03	4.8780E + 03	3.4298E + 03	4.2647E + 03	6.6856E + 03	3.9109E + 03	3.3848E + 03	3.3711E + 03	4.5304E + 03	3.5772E + 03	3.1482E + 03
	Std	8.3105E + 02	1.2030E + 03	6.3037E + 03	5.1780E + 02	3.4580E + 02	7.1637E + 02	1.1222E + 03	4.8791E + 02	3.4599E + 03	6.6780E + 02	4.6948E + 02	3.7948E + 02	3.0602E + 02
F_{17}	Mean	2.7592E + 03	1.6189E + 04	3.3218E + 03	4.0770E + 03	3.0657E + 03	3.6532E + 03	5.9661E + 03	3.4212E + 03	3.1778E + 03	3.0288E + 03	4.5666E + 03	3.2664E + 03	2.8865E + 03
	Std	2.0699E + 02	1.0334E + 04	4.1016E + 02	4.5831E + 02	3.4688E + 02	6.0595E + 02	1.6646E + 03	4.0263E + 02	3.2580E + 02	2.8449E + 02	4.0486E + 02	2.3673E + 02	3.1605E + 02
F_{18}	Mean	2.6559E + 06	1.7150E + 08	6.8623E + 06	3.1661E + 07	3.9686E + 06	1.6972E + 07	8.7850E + 07	3.4558E + 06	4.8235E + 06	2.9945E + 06	1.6912E + 06	2.0728E + 05	1.5108E + 05
	Std	1.8694E + 06	1.4033E + 08	5.9414E + 06	1.8249E + 07	3.7730E + 06	2.2383E + 07	6.7157E + 07	3.5977E + 06	5.6985E + 06	2.4705E + 06	7.8638E + 06	1.2753E + 05	6.4670E + 04
F_{19}	Mean	1.4582E + 04	5.3560E + 09	1.8354E + 04	4.8768E + 04	1.9278E + 05	1.8975E + 08	1.8311E + 07	4.1202E + 05	6.1361E + 05	1.9881E + 05	8.3890E + 05	1.6676E + 04	1.7513E + 04
	Std	9.1030E + 03	1.6916E + 09	1.6479E + 04	7.1967E + 09	9.9353E + 04	2.8654E + 08	1.6284E + 07	4.0971E + 05	9.0709E + 05	1.0750E + 05	4.4860E + 05	9.6057E + 03	9.5250E + 03
F_{20}	Mean	2.7933E + 03	3.9994E + 03	3.2917E + 03	3.4693E + 03	3.0989E + 03	3.4786E + 03	4.0214E + 03	3.4563E + 03	3.3042E + 03	3.1977E + 03	4.1028E + 03	3.5212E + 03	3.0385E + 03
	Std	2.1128E + 02	4.8817E + 03	3.6780E + 02	3.0625E + 03	3.3603E + 02	4.1986E + 03	3.3395E + 02	3.5623E + 02	3.8692E + 02	5.3470E + 02	2.3544E + 02	2.5314E + 02	2.1385E + 02
F_{21}	Mean	2.4506E + 03	3.0981E + 03	2.5834E + 03	2.8338E + 03	2.5808E + 03	2.7095E + 03	3.0305E + 03	2.6374E + 03	2.5176E + 03	2.6458E + 03	2.6267E + 03	2.5219E + 03	2.5299E + 03
	Std	3.1493E + 01	7.2590E + 01	5.7996E + 01	7.2031E + 01	3.6140E + 01	5.3319E + 01	5.0256E + 01	5.5556E + 01	9.9817E + 01	1.0370E + 01	3.5255E + 01	3.7701E + 01	4.3924E + 01
F_{22}	Mean	1.0294E + 04	1.6532E + 04	9.5633E + 03	1.3163E + 04	1.5853E + 04	1.4085E + 04	1.6702E + 04	1.0162E + 04	1.1649E + 04	1.0896E + 04	1.7286E + 04	1.0603E + 04	1.1710E + 04
	Std	1.5511E + 03	5.6665E + 02	3.3125E + 02	1.0049E + 03	7.5379E + 02	2.4908E + 03	8.5513E + 02	1.8120E + 03	1.9681E + 03	3.0396E + 03	1.1152E + 03	3.2288E + 03	1.6057E + 03
F_{23}	Mean	2.8991E + 03	4.5862E + 03	3.0506E + 03	3.6461E + 03	3.1334E + 03	3.3077E + 03	4.2613E + 03	3.2923E + 03	3.0804E + 03	2.9804E + 03	3.0614E + 03	3.0058E + 03	3.0049E + 03
	Std	4.1044E + 01	2.6354E + 02	9.3827E + 01	1.1603E + 02	7.7074E + 01	7.9551E + 01	2.1582E + 02	1.3059E + 02	7.3002E + 02	8.9894E + 01	4.3334E + 01	4.9583E + 01	5.6566E + 01
F_{24}	Mean	3.0814E + 03	5.0278E + 03	3.1833E + 03	3.6606E + 03	3.3121E + 03	3.5457E + 03	4.3517E + 03	3.4835E + 03	3.2428E + 03	3.2050E + 03	3.2299E + 03	3.1633E + 03	3.1804E + 03
	Std	4.6630E + 01	2.2618E + 02	6.4745E + 01	9.8960E + 01	6.0706E + 01	9.4739E + 01	1.7661E + 01	1.3589E + 02	1.1530E + 03	1.3681E + 03	5.3640E + 01	5.6554E + 01	6.2885E + 01
F_{25}	Mean	3.0784E + 03	1.6135E + 04	3.1117E + 03	4.8503E + 03	3.9512E + 03	6.1030E + 03	9.8699E + 03	3.1597E + 03	3.3251E + 03	3.1111E + 03	2.9313E + 03	3.1449E + 03	3.1247E + 03
	Std	2.6142E + 01	1.5170E + 03	3.9598E + 01	5.1129E + 02	3.4978E + 01	6.8359E + 03	2.1289E + 02	3.2518E + 03	7.8467E + 01	3.6735E + 03	5.6720E-02	4.5587E + 01	4.1847E + 01
F_{26}	Mean	5.1101E + 03	1.7212E + 04	6.2393E + 03	1.1555E + 04	9.8334E + 03	9.7487E + 03	1.5957E + 04	9.7712E + 03	7.8014E + 03	5.8622E + 03	7.2403E + 03	6.4016E + 03	6.5749E + 03
	Std	5.2349E + 02	1.1992E + 03	2.4549E + 03	1.7918E + 03	1.6907E + 03	8.5548E + 03	9.3321E + 02	1.1652E + 03	7.0691E + 03	6.2120E + 03	5.6714E + 02	7.9144E + 02	7.3989E + 02
F_{27}	Mean	3.2000E + 03	7.1011E + 03	3.5322E + 03	4.5058E + 03	3.8327E + 03	4.1942E + 03	5.5784E + 03	3.8796E + 03	3.8379E + 03	3.4037E + 03	3.2000E + 03	3.5587E + 03	3.5430E + 03
	Std	5.7491E-05	7.4596E + 02	1.2572E + 03	3.0894E + 02	1.4579E + 02	2.2636E + 02	6.2106E + 02	2.6879E + 02	9.5986E + 03	8.2611E + 03	1.2220E-04	1.0404E + 02	9.8371E + 01
F_{28}	Mean	3.3013E + 03	1.2724E + 04	3.3821E + 03	6.0905E + 03	4.6407E + 03	6.2823E + 03	9.1675E + 03	3.4475E + 03	4.1614E + 03	3.3547E + 03	3.3000E + 03	3.5135E + 03	3.4450E + 03
	Std	0.03	0.04	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03

(continued on next page)

Table 9 (continued)

Function	Metric	MSAO	AOA	SMA	AO	ARO	GJO	TSA	DO	SO	SAO	CMA-ES	L SHADE-SPACMA	L SHADE-cnEPSSn
F_{29}	Std	6.9604E + 00	1.6455E + 03	4.2607E + 01	4.4729E + 02	3.8579E + 02	6.8283E + 02	1.2274E + 03	5.7203E + 01	3.1378E + 02	2.9691E + 01	6.2935E - 05	9.9930E + 01	8.3725E + 01
	Mean	3.7123E + 03	7.0177E + 04	5.0600E + 03	7.5096E + 03	4.8169E + 03	6.1775E + 03	2.6943E + 04	5.1770E + 03	5.0404E + 03	4.2687E + 03	5.8679E + 03	4.2563E + 03	4.3674E + 03
F_{30}	Std	2.6984E + 02	6.5063E + 04	3.5977E + 02	9.7177E + 02	3.4932E + 02	5.2309E + 02	3.5315E + 04	4.4106E + 02	3.8539E + 02	3.3221E + 02	4.2187E + 02	2.9007E + 02	2.8836E + 02
	Mean	3.4862E + 05	8.1198E + 09	9.0242E + 06	2.4138E + 08	1.6315E + 07	5.1333E + 08	3.4583E + 09	2.7780E + 07	1.4759E + 07	1.1616E + 06	1.3636E + 07	3.5323E + 06	5.1871E + 06

obtained optimal results are documented and analyzed.

5.1. Pressure vessel design problem

The design of pressure vessels constitutes a prevalent constrained optimization issue in the fields of physical and chemical engineering, which entails four decision variables, namely, shell thickness ($T_s = x_1$), thickness of the head ($T_h = x_2$), internal radius ($R = x_3$), and length of a cylindrical portion excluding the head ($L = x_4$). As depicted in Fig. 18, the primary goal of this issue is minimizing the overall expense of molding, material, and welding for the pressure vessel while not violating any constraints. The mathematical model is illustrated below.

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

$$\text{Minimize } f(\mathbf{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \quad (26)$$

$$\text{Subject to } \begin{cases} g_1(\mathbf{x}) = -x_1 + 0.0193x_3 \leq 0, \\ g_2(\mathbf{x}) = -x_3 + 0.00954x_3 \leq 0, \\ g_3(\mathbf{x}) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0, \\ g_4(\mathbf{x}) = x_4 - 240 \leq 0. \end{cases} \quad (27)$$

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

Table 14 summarizes the numerical results obtained by MSAO and different comparative algorithms for the pressure vessel design. MSAO ranks first as superior to other methods, which brings variable values of the optimal solution $\mathbf{x} = [0.7782, 0.3846, 40.3196, 200.0000]$ with the corresponding fitness $f(\mathbf{x}) = 5885.3328$. Therefore, MSAO offers more visible advantages in solving the pressure vessel design problem.

5.2. Step-cone pulley design problem

The ultimate goal in addressing the step-cone pulley design challenge is to decrease the weight of four step-cone pulleys as much as possible by optimizing five decision variables, where the first four parameters denote the diameter of each pulley stage and the last one pertains to the pulley width, as illustrated in Fig. 19. To ensure that the transmit power remains at or above 0.75 hp, there are three equality constraints and eight inequality constraints associated with this problem. These constraints are applicable when the input speed is 350 rpm and the output speeds are 750, 450, 250, and 150 rpm. This example is a challenging engineering optimization task. It can be mathematically defined as follows:

Consider $\mathbf{x} = [x_1, x_2, x_3, x_4, x_5] = [d_1, d_2, d_3, d_4, \omega]$

$$\text{Minimize } f(\mathbf{x}) = \rho x_5 \left[x_1^2 \left\{ 11 + \left(\frac{N_1}{N} \right)^2 \right\} + x_2^2 \left\{ 1 + \left(\frac{N_2}{N} \right)^2 \right\} + x_3^2 \left\{ 1 + \left(\frac{N_3}{N} \right)^2 \right\} + x_4^2 \left\{ 1 + \left(\frac{N_4}{N} \right)^2 \right\} \right] \quad (28)$$

$$\text{Subject to } \begin{cases} h_1(\mathbf{x}) = C_1 - C_2 = 0, \\ h_2(\mathbf{x}) = C_1 - C_3 = 0, \\ h_3(\mathbf{x}) = C_1 - C_4 = 0, \\ g_{1,2,3,4}(\mathbf{x}) = R_i \geq 2, \\ g_{5,6,7,8}(\mathbf{x}) = P_i \geq 0.75 * 745.6998. \end{cases} \quad (29)$$

where $40 \leq x_1, x_2, x_3, x_4 \leq 500, 16 \leq x_5 \leq 100$. C_i denotes the belt length required to achieve speed N_i , R_i denotes the tension ratio, and P_i denotes the power transmitted at each step, which can be calculated as follows:

Table 10

Comparison of MSAO with different algorithms on 100-dimensional CEC2017 functions.

Function	Metric	MSAO	AOA	SMA	AO	ARO	GJO	TSA	DO	SO	SAO	CMA-ES	L SHADE-SPACMA	L SHADE-cnEpSin
F_1	Mean	1.6864E + 07	2.6705E + 11	4.2464E + 08	9.0922E + 10	7.7693E + 11	1.3871E + 11	2.0105E + 08	8.3691E + 10	1.4309E + 09	8.8158E + 07	7.8761E + 07	9.0009E + 09	6.2166E + 09
	Std	1.5584E + 07	1.2912E + 10	1.2704E + 08	9.7622E + 09	1.4771E + 10	1.4290E + 10	1.3956E + 10	2.4001E + 08	3.0403E + 09	2.4595E + 09	1.6538E + 07	3.5677E + 09	2.2345E + 09
F_3	Mean	3.7113E + 05	3.7228E + 05	8.1991E + 05	3.5965E + 05	3.6277E + 05	4.0850E + 05	3.4985E + 05	6.4541E + 05	8.1576E + 05	9.3355E + 05	2.5890E + 06	4.4069E + 05	3.7347E + 05
	Std	3.3161E + 04	4.0336E + 04	2.9968E + 05	1.1003E + 04	1.7248E + 04	5.6884E + 04	8.5348E + 03	9.2186E + 04	1.6139E + 05	2.0158E + 05	1.0671E + 06	1.3625E + 05	8.0636E + 04
F_4	Mean	8.3587E + 02	9.5461E + 04	9.9708E + 02	1.8355E + 04	9.7039E + 03	1.9942E + 04	4.7980E + 04	1.1697E + 03	2.7982E + 03	1.3887E + 03	5.0972E + 02	1.9043E + 03	1.5377E + 03
	Std	6.1144E + 01	1.4968E + 04	8.2898E + 02	2.6942E + 03	2.3736E + 03	3.7816E + 03	1.2011E + 04	9.6448E + 03	5.3867E + 03	1.8070E + 02	1.3903E + 02	3.8096E + 02	2.9372E + 02
F_5	Mean	1.1271E + 03	2.0609E + 03	1.3535E + 03	1.7014E + 03	1.5340E + 03	1.6165E + 03	1.9741E + 03	1.4198E + 03	1.1756E + 03	1.6397E + 03	1.4082E + 03	1.2601E + 03	1.2831E + 03
	Std	3.0822E + 01	6.5103E + 01	8.1232E + 01	8.5930E + 01	7.0695E + 01	1.0781E + 02	5.8343E + 01	1.0680E + 02	7.2163E + 01	6.1808E + 01	2.6455E + 02	1.1429E + 02	1.1440E + 02
F_6	Mean	6.0893E + 02	7.0929E + 02	6.6333E + 02	6.9069E + 02	6.6164E + 02	6.7577E + 02	7.0775E + 02	6.7345E + 02	6.4825E + 02	6.3731E + 02	6.0420E + 02	6.2590E + 02	6.4119E + 02
	Std	3.1481E + 00	5.2781E + 00	6.5234E + 00	4.6926E + 00	6.3262E + 00	9.0137E + 00	4.1922E + 00	1.0083E + 01	5.2111E + 00	7.4805E + 00	4.5482E-01	3.2166E + 00	7.4405E + 00
F_7	Mean	1.8285E + 03	3.9889E + 03	2.4236E + 03	3.4502E + 03	3.0321E + 03	2.9822E + 03	3.7874E + 03	2.8452E + 03	2.2574E + 03	2.3767E + 03	1.7312E + 03	2.2665E + 03	2.4912E + 03
	Std	2.6346E + 02	5.9428E + 01	2.4095E + 02	1.9221E + 02	1.9320E + 03	1.7913E + 03	1.1754E + 03	2.1151E + 02	1.4482E + 02	1.2233E + 02	2.9721E + 02	2.1737E + 02	2.6576E + 02
F_8	Mean	1.4485E + 03	2.5209E + 03	1.7017E + 03	2.1389E + 03	1.9106E + 03	1.9862E + 03	2.4324E + 03	1.7739E + 03	1.5426E + 03	1.9060E + 03	1.7058E + 03	1.5857E + 03	1.5808E + 03
	Std	2.7849E + 01	7.2123E + 01	1.0734E + 02	8.5639E + 01	7.4686E + 01	1.3702E + 02	7.7189E + 01	1.2422E + 02	7.6070E + 01	9.1507E + 01	3.0938E + 02	1.0938E + 02	9.1569E + 01
F_9	Mean	2.4047E + 04	7.3568E + 04	3.4649E + 04	6.6293E + 04	4.1994E + 04	6.5311E + 04	7.9308E + 04	5.2811E + 04	2.8807E + 04	3.3233E + 04	1.0766E + 04	3.0517E + 04	3.3062E + 04
	Std	6.2368E + 03	6.7997E + 03	2.7562E + 03	5.2772E + 03	6.3044E + 03	8.7627E + 03	4.2188E + 03	1.3959E + 04	7.6162E + 03	1.1913E + 04	3.7956E + 03	8.9158E + 03	6.3077E + 03
F_{10}	Mean	2.1436E + 04	3.0976E + 04	1.8100E + 04	2.5956E + 04	3.1805E + 04	2.7054E + 04	3.1460E + 04	1.8822E + 04	3.1063E + 04	3.0695E + 04	3.4255E + 04	2.3393E + 04	2.3540E + 04
	Std	1.2666E + 03	1.2045E + 03	6.3786E + 02	2.0871E + 03	1.1913E + 03	4.7368E + 03	1.2310E + 03	1.1631E + 03	1.7855E + 03	2.5728E + 03	1.8067E + 03	3.1274E + 03	2.5088E + 03
F_{11}	Mean	5.0467E + 04	1.6595E + 05	3.0115E + 04	3.7744E + 05	9.1948E + 04	1.0366E + 05	1.8522E + 05	3.1138E + 04	1.3104E + 05	2.3983E + 05	9.4854E + 05	9.7578E + 04	5.0219E + 04
	Std	3.3058E + 04	2.2549E + 04	8.2449E + 03	7.0778E + 04	1.4074E + 04	1.6986E + 04	1.8960E + 04	1.1900E + 04	2.4478E + 04	7.3477E + 04	3.0492E + 04	4.0609E + 04	3.9221E + 04
F_{12}	Mean	3.0199E + 07	1.9279E + 11	2.7781E + 08	3.6393E + 010	8.4787E + 09	4.6567E + 10	1.1263E + 11	5.2801E + 11	1.7587E + 09	6.1274E + 08	2.1073E + 08	7.2911E + 08	5.7787E + 08
	Std	1.2851E + 07	1.7498E + 10	8.4915E + 07	6.4843E + 09	4.1372E + 09	1.1002E + 07	2.0900E + 07	1.8173E + 08	7.4341E + 08	3.6056E + 08	5.4848E + 08	2.7835E + 08	2.6647E + 08
F_{13}	Mean	5.3537E + 03	4.5443E + 10	8.8871E + 05	3.5276E + 09	1.6164E + 08	8.1446E + 09	2.7042E + 10	3.2016E + 06	6.5436E + 06	7.3741E + 04	3.7827E + 06	1.4792E + 06	7.6535E + 04
	Std	3.1671E + 03	7.5577E + 09	3.1311E + 06	9.9275E + 08	1.5246E + 08	3.3627E + 09	5.9394E + 09	8.8939E + 08	7.3832E + 06	6.1847E + 04	1.3315E + 06	2.1934E + 06	3.6584E + 04
F_{14}	Mean	1.6437E + 06	9.7492E + 07	4.1876E + 06	2.1400E + 07	7.3900E + 06	1.5752E + 07	3.8460E + 07	4.9956E + 06	8.3062E + 06	2.5695E + 06	7.2451E + 05	7.8435E + 05	5.9901E + 05
	Std	1.0673E + 06	5.1360E + 07	2.2527E + 06	9.1284E + 06	2.6274E + 06	8.4602E + 06	2.0620E + 07	2.5363E + 06	4.4754E + 06	1.1788E + 06	1.5538E + 05	4.9321E + 05	4.1990E + 05

(continued on next page)

Table 10 (continued)

Function	Metric	MSAO	AOA	SMA	AO	ARO	GJO	TSA	DO	SO	SAO	CMA-ES	LSHADE-SPACMA	LSHADE-cnEpSin
F_{15}	Mean	3.2412E + 03	2.5538E + 10	3.1943E + 05	4.8007E + 08	1.3533E + 06	2.7757E + 09	1.1851E + 10	1.4155E + 05	3.2614E + 05	7.7647E + 03	3.4342E + 08	2.6900E + 04	3.6686E + 04
	Std	1.7406E + 03	4.5597E + 09	8.1171E + 05	2.8093E + 08	1.0442E + 06	1.8855E + 09	5.1978E + 09	1.3335E + 05	2.8698E + 05	3.3161E + 03	1.2967E + 08	1.6186E + 04	1.6871E + 04
F_{16}	Mean	7.2262E + 03	2.2239E + 04	6.6522E + 03	1.1522E + 04	7.4079E + 03	9.5784E + 03	1.7184E + 04	7.9545E + 03	6.7373E + 03	7.4333E + 03	1.1314E + 04	7.2861E + 03	6.8548E + 03
	Std	7.1918E + 02	3.6208E + 03	6.6509E + 02	1.5040E + 03	7.3106E + 02	8.5173E + 02	2.3056E + 03	2.2803E + 03	1.1681E + 03	2.1542E + 03	7.2574E + 02	1.0579E + 03	8.0221E + 02
F_{17}	Mean	5.7989E + 03	7.3073E + 06	6.1506E + 03	2.0932E + 04	5.7959E + 03	2.6308E + 04	1.1104E + 06	5.9094E + 03	5.6940E + 03	6.6303E + 03	1.5855E + 04	5.6453E + 03	5.3032E + 03
	Std	6.9801E + 02	6.5590E + 06	1.3401E + 03	1.2313E + 04	6.3677E + 02	3.8627E + 04	1.3613E + 06	5.2426E + 02	4.2512E + 02	1.4072E + 03	2.0720E + 02	5.8986E + 02	3.5133E + 02
F_{18}	Mean	4.7169E + 06	1.6136E + 08	8.4897E + 06	2.1134E + 07	6.0771E + 06	1.7613E + 07	4.4569E + 07	6.2572E + 06	1.2605E + 07	1.0305E + 07	1.2964E + 08	1.0933E + 06	9.5086E + 05
	Std	2.8968E + 06	9.9229E + 07	3.8023E + 06	1.0497E + 07	2.7395E + 06	1.0411E + 07	2.6158E + 07	3.1919E + 06	5.0569E + 06	5.3518E + 06	4.4697E + 07	7.2513E + 05	6.0085E + 05
F_{19}	Mean	5.0661E + 03	2.2930E + 10	2.1292E + 05	5.1659E + 08	3.9535E + 06	1.7906E + 09	1.3949E + 10	2.4083E + 06	2.5777E + 06	6.9430E + 05	3.5034E + 08	1.5890E + 05	3.1902E + 05
	Std	5.1701E + 03	5.5435E + 09	1.7234E + 05	3.3136E + 08	2.7934E + 06	1.8816E + 08	4.1358E + 09	1.7255E + 06	2.4703E + 06	5.2722E + 06	1.0755E + 08	2.6285E + 05	3.6216E + 05
F_{20}	Mean	5.8962E + 03	7.4162E + 03	7.0808E + 03	6.1585E + 03	5.3167E + 03	6.5320E + 03	7.5261E + 03	5.8395E + 03	7.2061E + 03	6.7779E + 03	8.0271E + 03	6.6383E + 03	5.9616E + 03
	Std	6.4006E + 02	5.9433E + 02	6.3307E + 02	4.9979E + 02	3.2091E + 02	1.1607E + 03	5.2309E + 02	4.8638E + 02	4.2655E + 02	1.6838E + 02	3.2571E + 02	4.6993E + 02	6.1165E + 02
F_{21}	Mean	2.8030E + 03	4.7556E + 03	3.1384E + 03	4.2430E + 03	3.3284E + 03	3.5324E + 03	4.3113E + 03	3.3671E + 03	3.1034E + 03	3.4581E + 03	3.2475E + 03	3.1328E + 03	3.1915E + 03
	Std	3.2311E + 01	1.8037E + 02	1.0760E + 02	2.6201E + 02	9.3839E + 01	1.1153E + 02	1.5712E + 02	1.3173E + 02	7.9110E + 01	6.5244E + 01	2.6180E + 01	7.4619E + 01	9.5538E + 01
F_{22}	Mean	2.5457E + 04	3.3941E + 04	2.0400E + 04	2.8719E + 04	2.4911E + 04	3.0043E + 04	3.3746E + 04	2.1940E + 04	3.2532E + 04	3.0009E + 04	3.5995E + 04	2.6400E + 04	3.4234E + 04
	Std	2.0991E + 03	7.4184E + 02	5.2315E + 02	1.7029E + 03	1.4722E + 03	5.3052E + 03	1.2420E + 03	1.3555E + 03	2.8188E + 03	4.9742E + 03	1.5361E + 03	2.7929E + 03	7.4638E + 02
F_{23}	Mean	3.1787E + 03	7.3351E + 03	3.5694E + 03	5.0030E + 03	3.9285E + 03	4.4883E + 03	6.0604E + 03	4.0236E + 03	3.7009E + 03	3.6094E + 03	3.7655E + 03	3.6396E + 03	3.7947E + 03
	Std	2.5372E + 01	5.7205E + 02	1.1732E + 02	2.0110E + 01	9.5944E + 01	2.2519E + 02	3.8735E + 02	1.6390E + 02	6.6562E + 01	1.4045E + 02	6.3180E + 01	9.7451E + 01	1.5424E + 02
F_{24}	Mean	3.7266E + 03	1.1542E + 04	4.1711E + 03	6.6745E + 03	4.9997E + 03	5.9757E + 03	9.3496E + 03	4.8579E + 03	4.7088E + 03	4.2371E + 03	4.1989E + 03	4.4442E + 03	4.5817E + 03
	Std	2.6067E + 01	8.1253E + 02	1.2429E + 02	4.3497E + 03	1.9550E + 02	2.7739E + 03	1.1003E + 03	2.3645E + 02	1.8925E + 02	1.4865E + 02	1.8841E + 02	1.2778E + 02	2.5792E + 02
F_{25}	Mean	3.5166E + 03	2.9808E + 04	3.6868E + 03	9.8281E + 03	8.1538E + 03	1.2181E + 04	2.1053E + 04	3.8240E + 03	5.3747E + 03	4.6061E + 03	3.3568E + 03	4.4008E + 03	4.2888E + 03
	Std	9.9337E + 01	2.9294E + 03	5.7414E + 03	7.8557E + 03	8.8710E + 03	1.5951E + 03	2.9292E + 03	9.1304E + 03	5.4661E + 03	2.8386E + 03	3.0006E + 03	2.8381E + 02	2.2761E + 02
F_{26}	Mean	1.1065E + 04	5.2515E + 04	1.6263E + 04	3.3851E + 04	2.7326E + 04	2.8461E + 04	4.6516E + 04	2.3749E + 04	1.9508E + 04	1.5990E + 04	1.5196E + 04	1.7691E + 04	1.7017E + 04
	Std	2.5174E + 02	4.4294E + 03	2.0528E + 03	2.6062E + 03	2.1924E + 03	1.9995E + 03	3.4476E + 03	2.1147E + 03	1.2516E + 03	1.1428E + 03	1.7723E + 03	2.6318E + 03	1.7130E + 03
F_{27}	Mean	3.2000E + 03	1.3410E + 04	3.7402E + 03	7.2286E + 03	4.6449E + 03	5.7048E + 03	9.5122E + 03	4.2599E + 03	4.3195E + 03	3.5296E + 03	3.2000E + 03	3.9678E + 03	4.0136E + 03
	Std	2.3887E-05	1.4088E + 03	9.1783E + 03	6.8666E + 03	2.4513E + 03	4.7307E + 03	1.0500E + 03	2.6087E + 03	1.5595E + 02	5.5068E + 03	7.3761E-05	1.7059E + 02	2.1698E + 02
F_{28}	Mean	3.3147E + 03	3.3844E + 04	3.7445E + 03	1.4768E + 04	1.1418E + 04	1.5473E + 04	2.8021E + 04	4.0076E + 04	1.0116E + 04	4.5279E + 03	3.3000E + 03	5.5476E + 03	5.0764E + 03
	Std	0.03	0.04	0.03	0.04	0.04	0.04	0.04	0.04	0.03	0.04	0.03	0.03	0.03

(continued on next page)

Table 10 (continued)

Function	Metric	MSAO	AOA	SMA	AO	ARO	GJO	TSA	DO	SO	SAO	CMA-ES	L SHADE-SPACMA	L SHADE-cnEpSin	
F_{29}	Std	8.0413E +	2.8172E +	6.0235E +	1.6076E +	1.4156E +	1.5643E +	3.2562E +	1.5684E +	1.5720E +	3.7961E +	7.4761E -05	5.5889E + 02	5.6971E + 02	
	Mean	01	03	01	03	03	03	03	02	03	02	03	8.3642E + 03		
F_{30}	Std	7.0862E +	5.9599E +	8.0553E +	1.8247E +	9.7899E +	1.8064E +	1.9542E +	9.0714E +	9.1172E +	7.2100E +	1.6439E +	7.9118E + 03		
	Mean	03	05	03	04	03	04	05	03	03	03	04	1.3366E + 07		

$$\left\{ \begin{array}{l} C_i = \frac{\pi x_i}{2} \left(1 + \frac{N_i}{N} \right) + \frac{\left(\frac{N_i}{N} - 1 \right)^2}{4a} + 2a, i = 1, 2, 3, 4 \\ R_i = \exp \left[\mu \left\{ \pi - 2 \sin^{-1} \left\{ \left(\frac{N_i}{N} - 1 \right) \frac{x_i}{2a} \right\} \right\} \right], i = 1, 2, 3, 4 \\ P_i = s t x_5 \left[1 - \exp \left[-\mu \left\{ \pi - 2 \sin^{-1} \left\{ \left(\frac{N_i}{N} - 1 \right) \frac{x_i}{2a} \right\} \right\} \right] \right] \frac{\pi x_i N_i}{60}, i = 1, 2, 3, 4 \\ \rho = 7200 \text{ kg/m}^3, a = 3 \text{ m}, \mu = 0.35, s = 1.75 \text{ MPa}, t = 8 \text{ mm}. \end{array} \right. \quad (30)$$

The design problem of step-cone pulleys is solved by the proposed MSAO as well as AOA, SMA, AO, ARO, GJO, TSA, DO, SO, SAO, CMA-ES, L SHADE-SPACMA, L SHADE-cnEpSin, CSOAOA, IYDSE, IHAOAVOA, ESO, ESHHO, and RCLMSAOA. The minimum weight and optimal values for the five decision variables obtained by each optimization method are recorded in Table 15. The optimal candidate solution revealed by MSAO ranks first among all algorithms, followed by ESO and L SHADE-cnEpSin. When d_1, d_2, d_3, d_4 , and ω are set to 38.4140, 52.8586, 70.4727, 84.4957, and 90.0000 respectively, the step-cone pulley has the minimum weight $f(\mathbf{x}) = 16.0903$. It can be concluded that MSAO exhibits strong competitiveness in solving this problem as well.

5.3. Cantilever beam design problem

The cantilever beam design is a widespread unconstrained optimization paradigm in civil engineering. As can be seen from Fig. 20, this design is composed of five hollow components with square cross-sections, each defined by a single variable. The thickness remains constant, resulting in a total of five structural parameters to consider, denoted as x_1, x_2, x_3, x_4 , and x_5 . The mathematical model is as follows:

Consider $\mathbf{x} = [x_1, x_2, x_3, x_4, x_5]$

$$\text{Minimize } f(\mathbf{x}) = 0.6224(x_1 + x_2 + x_3 + x_4 + x_5) \quad (31)$$

$$\text{Subject to } g(\mathbf{x}) = \frac{61}{x_1^3} + \frac{27}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0 \quad (32)$$

where $0.01 \leq x_1, x_2, x_3, x_4, x_5 \leq 100$.

The optimal values of the five decision variables as well as the corresponding solutions obtained by MSAO alongside sixteen other competing techniques are summarized in Table 16. The data in this table indicates that the proposed MSAO is the most effective optimization technique for addressing the cantilever beam design problem since MSAO reveals the best design assurance with minimum weight $f(\mathbf{x}) = 1.33996$ among all methods. In view of the above, MSAO can provide a superior solution and perform well for the cantilever beam design problem.

5.4. Speed reducer design problem

Speed reducer design is a complex nonlinear optimization project in mechanical braking systems, aimed at minimizing the total weight of a speed reducer subject to the gear teeth's bending stress, stress in the shaft, surface stress, and transverse shaft deflection. As illustrated in Fig. 21, this problem contains seven decision variables: surface width ($b = x_1$), gear module ($m = x_2$), pinion teeth count ($p = x_3$), length of the 1st shaft and 2nd shaft between bearings ($l_1 = x_4, l_2 = x_5$), and diameters of the 1st and 2nd shafts ($d_1 = x_6, d_2 = x_7$). The mathematical model is as follows:

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

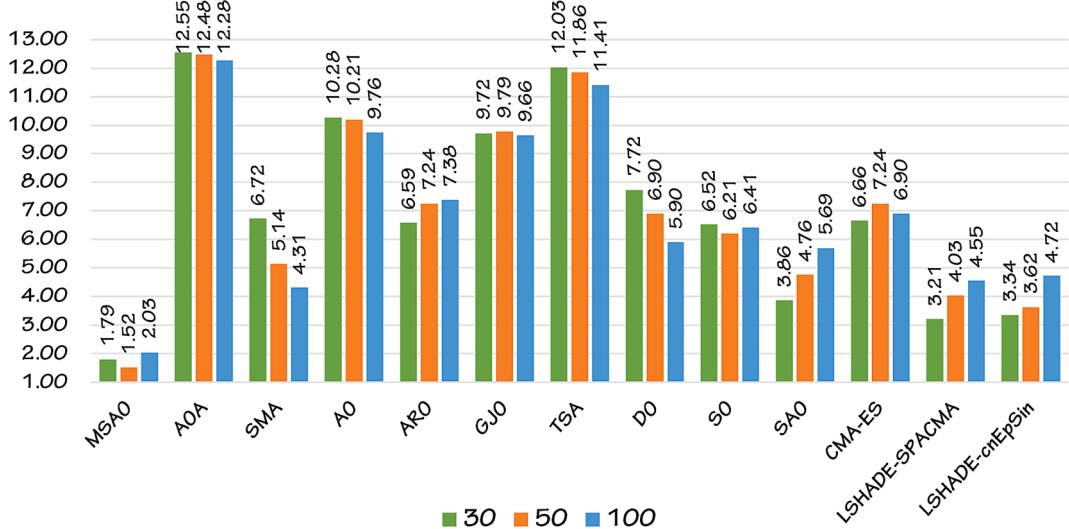


Fig. 14. Friedman mean ranking of MSAO and comparison methods on CEC2017 functions with different dimensions.

Table 11
Parameter configurations for different modified high-performance optimizers.

Algorithm	Reference	Year	Parameter settings
CSOOOA	[13]	2022	$\alpha = 5; \mu = 0.499; \delta \in [0.001, 0.01]$
IYDSE	[52]	2023	$\alpha_F = 0.1;$ $\lambda = 5 \times 10^{-6};$ $d = 5 \times 10^{-3};$ $L = 1;$ $I = 0.01;$ $\delta = 0.38$
IHAOAVOA	[67]	2022	$L_1 = 0.8;$ $L_2 = 0.2;$ $w = 2.5;$ $P_2 = 0.4;$ $P_3 = 0.6;$ $U = 0.00565;$ $r = 10;$ $\omega = 0.05;$ $k = 12000$
ESO	[72]	2023	$c_1 = 0.5; c_2 = 0.05; c_3 = 2$
EESHHO	[101]	2021	$sp \in (-1, 1); E_0 \in (-1, 1)$
RCLSMAOA	[102]	2023	$z = 0.03; \alpha = 5; \mu = 0.499$

$$\begin{aligned} \text{Minimize } f(\mathbf{x}) = & 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 \\ & + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) \end{aligned} \quad (33)$$

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

Table 17 reports the numerical results achieved by different algorithms in solving the speed reducer problem. The proposed MSAO, LSHADE-SPACMA, EESHHO, and ESO yield the same optimal candidate solution $f(\mathbf{x}) = 2994.4711$, which ranks tied for first place among all algorithms. On the other hand, the performance of SAO is not satisfactory, and its outcome ranks third to last. Therefore, it is reasonable to believe that compared with SAO, MSAO has a significant improvement in convergence accuracy and is preferable for solving such problem.

5.5. Robot gripper design problem

Robot gripper design focuses on how to minimize the difference between the maximum and minimum forces generated by the gripper. This problem encompasses six nonlinear constraints and seven continuous design variables, including the lengths of three rods ($a = x_1, b = x_2, c = x_3$), vertical links displacement ($e = x_4$), vertical distance from the 1st mech-arm node to the actuator ($f = x_5$), horizontal distance from the actuator to the articulation point ($l = x_6$), and the angle between the 2nd and 3rd rods ($\delta = x_7$). As depicted in Fig. 22, the mathematical model for this topic is specified below:

$$\text{Consider } \mathbf{x} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7] = [a, b, c, e, f, l, \delta]$$

$$\left\{ \begin{array}{l} g_1(\mathbf{x}) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0, g_2(\mathbf{x}) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0, \\ g_3(\mathbf{x}) = \frac{1.93x_4^2}{x_2x_6^2x_3} - 1 \leq 0, g_4(\mathbf{x}) = \frac{1.93x_5^2}{x_2x_7^4x_3} - 1 \leq 0, \\ g_5(\mathbf{x}) = \frac{\sqrt{\left(\frac{745x_5}{x_2x_3}\right)^2 + 16.9 \times 10^6}}{110x_6^3} - 1 \leq 0, g_6(\mathbf{x}) = \frac{\sqrt{\left(\frac{745x_5}{x_2x_3}\right)^2 + 157.5 \times 10^6}}{85x_7^3} - 1 \leq 0, \\ g_7(\mathbf{x}) = \frac{x_2x_3}{40} - 1 \leq 0, g_8(\mathbf{x}) = \frac{5x_2}{x_1} - 1 \leq 0, \\ g_9(\mathbf{x}) = \frac{x_1}{12x_2} - 1 \leq 0, g_{10}(\mathbf{x}) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0, \\ g_{11}(\mathbf{x}) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0. \end{array} \right. \quad (34)$$

Table 12

Statistical results of MSAO and different modified algorithms on 12 CEC2022 functions.

Function	Metric	MSAO	CSOOAOA	IYDSE	IHAOAVOA	ESO	EESHHO	RCLSMOA
F_{31}	Mean	300.0000	1062.9905	300.3068	512.1750	355.1756	415.2608	1108.4244
	Std	0.0000E + 00	7.5079E + 02	3.3733E-01	3.4116E + 02	1.0760E + 02	1.4191E + 02	4.1905E + 03
	p-value	–	3.1578E-12	3.1578E-12	3.1578E-12	3.1578E-12	3.1578E-12	3.1578E-12
	Rank	1	6	2	5	3	4	7
F_{32}	Mean	402.1621	415.2515	406.0385	423.6183	403.6475	418.1614	633.9864
	Std	1.8787E + 00	2.7441E + 01	2.6249E + 00	3.3366E + 01	3.8470E + 00	2.7119E + 01	6.9730E + 02
	p-value	–	3.2651E-02	4.2175E-04	8.7710E-02	2.7071E-01	5.9706E-05	1.6980E-08
	Rank	1	4	3	6	2	5	7
F_{33}	Mean	600.0000	600.7131	606.1866	629.2536	600.1334	605.5897	602.9506
	Std	2.7247E-05	1.3100E + 00	3.5241E + 00	1.2377E + 01	2.4854E-01	7.0623E + 00	6.8097E + 00
	p-value	–	5.2039E-12	5.2039E-12	5.2039E-12	5.2039E-12	5.2039E-12	5.2039E-12
	Rank	1	3	6	7	2	5	4
F_{34}	Mean	808.3245	826.2924	813.7229	828.3245	813.6569	823.9121	832.0309
	Std	3.5681E + 00	6.0367E + 00	4.2904E + 00	6.4738E + 00	4.0847E + 00	7.8538E + 00	1.8942E + 01
	p-value	–	7.2391E-11	1.2376E-05	2.9561E-11	4.0749E-06	3.7483E-10	2.3258E-10
	Rank	1	5	3	6	2	4	7
F_{35}	Mean	900.0633	1089.1030	917.5977	1338.7428	901.9092	1183.0850	1108.6072
	Std	1.3933E-01	1.7513E + 02	2.3123E + 01	1.4606E + 02	7.6573E + 00	2.0349E + 02	6.9231E + 02
	p-value	–	2.2521E-11	2.0005E-10	2.2521E-11	4.7469E-06	2.2521E-11	2.6727E-10
	Rank	1	4	3	7	2	6	5
F_{36}	Mean	1811.6503	3003.5480	1813.0594	2914.9535	2286.7256	3213.4650	473906.0280
	Std	6.0152E + 00	1.3567E + 03	1.5578E + 01	1.2638E + 03	5.2025E + 02	1.7133E + 03	1.2951E + 06
	p-value	–	4.1997E-10	1.1199E-02	3.0199E-11	6.0658E-11	4.9752E-11	3.0199E-11
	Rank	1	5	2	4	3	6	7
F_{37}	Mean	2009.0177	2017.6228	2029.7413	2051.0622	2019.1897	2029.6524	2047.9334
	Std	7.6386E-01	1.0391E + 01	4.4838E + 00	2.4221E + 01	9.2364E + 00	1.6509E + 01	4.8424E + 01
	p-value	–	4.7138E-04	4.5043E-11	1.2057E-10	1.5292E-05	1.0666E-07	1.4643E-10
	Rank	1	2	5	7	3	4	6
F_{38}	Mean	2202.4200	2217.7609	2220.9323	2227.4311	2220.2121	2221.2409	2270.2293
	Std	1.0148E-02	6.6836E + 00	5.0165E + 00	6.7134E + 00	7.4842E + 00	1.1737E + 00	1.3799E + 02
	p-value	–	5.7929E-01	1.5638E-02	2.7829E-07	7.0127E-02	5.1060E-01	1.3594E-07
	Rank	1	2	4	6	3	5	7
F_{39}	Mean	2485.6519	2531.1656	2529.3050	2554.7181	2529.3673	2530.8048	2577.5180
	Std	1.9155E-02	3.8555E + 00	3.3672E-02	3.7280E + 01	1.9439E-01	3.7249E + 00	8.9734E + 01
	p-value	–	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11
	Rank	1	5	2	6	3	4	7
F_{40}	Mean	2533.0867	2557.7923	2500.5715	2555.0689	2538.9073	2563.6964	2576.0168
	Std	1.5120E + 00	6.7770E + 01	1.2949E-01	6.3388E + 01	6.0805E + 01	6.0243E + 01	1.5056E + 02
	p-value	–	6.2027E-04	1.0315E-02	2.5974E-05	1.0763E-02	4.6390E-05	3.9881E-04
	Rank	2	5	1	4	3	6	7
F_{41}	Mean	2676.8121	2639.8472	2648.9917	2690.2573	2680.8730	2685.8631	2916.4796
	Std	1.1285E + 02	8.6883E + 01	8.8126E + 01	1.2635E + 02	1.1753E + 02	1.5290E + 02	4.6781E + 02
	p-value	–	4.0971E-02	2.0812E-02	9.9333E-04	9.9107E-03	6.9791E-03	2.5090E-05
	Rank	3	1	2	6	4	5	7
F_{42}	Mean	2848.0838	2867.3408	2863.0709	2867.2724	2864.9155	2880.9237	2876.9957
	Std	6.9537E-01	5.8572E + 00	1.4363E + 00	4.2681E + 00	1.9325E + 00	2.1818E + 01	1.7699E + 01
	p-value	–	3.0199E-11	3.0199E-11	3.0199E-11	4.5043E-11	3.0199E-11	3.0199E-11
	Rank	1	5	2	4	3	7	6
+/-	–	–	11/0/1	12/0/0	11/0/1	10/0/2	11/0/1	12/0/0
Mean rank		1.25	3.92	2.92	5.67	2.75	5.08	6.42
Final ranking		1	4	3	6	2	5	7

$$\text{Minimize } f(\mathbf{x}) = -\min_z F_k(\mathbf{x}, z) + \max_z F_k(\mathbf{x}, z) \quad (35)$$

$$\text{Subject to} \left\{ \begin{array}{l} g_1(\mathbf{x}) = -Y_{\min} + h((\mathbf{x}), Z_{\max}) \leq 0, g_2(\mathbf{x}) = -h((\mathbf{x}), Z_{\max}) \leq 0, \\ g_3(\mathbf{x}) = Y_{\max} - h((\mathbf{x}), 0) \leq 0, g_4(\mathbf{x}) = h((\mathbf{x}), 0) - Y_G \leq 0, \\ g_5(\mathbf{x}) = l^2 + e^2 - (a+b)^2 \leq 0, g_6(\mathbf{x}) = b^2 - (a-e)^2 - (l-Z_{\max})^2 \leq 0, \\ g_7(\mathbf{x}) = Z_{\max} - l \leq 0. \end{array} \right. \quad (36)$$

where $10 \leq x_1, x_2 \leq 150$, $100 \leq x_3 \leq 200$, $0 \leq x_4 \leq 50$, $10 \leq x_5 \leq 150$, $100 \leq x_6 \leq 300$, $1 \leq x_7 \leq 3.14$. The remaining parameter terms are calculated as:

$$\left\{ \begin{array}{l} F_k = \frac{Pb \sin(\alpha + \beta)}{2c \cos(\alpha)}, \\ \alpha = \cos^{-1} \left(\frac{a^2 + g^2 - b^2}{2ag} \right) + \phi, g = \sqrt{e^2 + (z-l)^2}, \\ \beta = \cos^{-1} \left(\frac{b^2 + g^2 - a^2}{2ag} \right) - \phi, \phi = \tan^{-1} \left(\frac{e}{l-z} \right), \\ h(\mathbf{x}, z) = 2(f + e + c \sin(\beta + \delta)), \\ Y_{\min} = 50, Y_{\max} = 100, Y_G = 150, Z_{\max} = 100, P = 100. \end{array} \right. \quad (37)$$

The minimum cost, decision variables, and ranking obtained by MSAO and other different optimization approaches for addressing this design problem are compared in Table 18. As we can see, MSAO again ranks first, suggesting that our suggested technique can outperform competitors in terms of search performance. MSAO provided a minimum cost of 2.5687, which is 5.30 % less than the second-ranked CSOOAOA. These numerical results demonstrate that MSAO is an outstanding tool used to deal with the design of robotic grippers.



Fig. 15. Radar plots of different modified optimization algorithms on 12 CEC2022 functions.

5.6. Rolling element bearing design problem

The last optimization theme is the design of a rolling element bearing, which requires maximizing its own dynamic load capacity. As illustrated in Fig. 23, there are a total of ten parameters to be considered, namely pitch diameter ($D_m = x_1$), ball diameter ($D_b = x_2$), ball number ($Z = x_3$), inner and outer raceway curvature coefficient ($f_i = x_4, f_o = x_5$), $K_{D\min}(x_6)$, $K_{D\max}(x_7)$, $\varepsilon(x_8)$, $e(x_9)$, and $\zeta(x_{10})$. This design is subject to nine nonlinear constraints and the mathematical representation is shown as follows:

$$\begin{aligned} \text{Consider } \mathbf{x} &= [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}] \\ &= [D_m, D_b, Z, f_i, f_o, K_{D\min}, K_{D\max}, \varepsilon, e, \zeta] \end{aligned}$$

$$\text{Maximize } f(\mathbf{x}) = \begin{cases} f_o Z^{2/3} D_b^{1.8}, & \text{if } D_b \leq 25.4 \text{ mm} \\ 3.647 f_o Z^{2/3} D_b^{1.4}, & \text{otherwise} \end{cases} \quad (38)$$

$$\left. \begin{array}{l} g_1(\mathbf{x}) = \frac{\phi_0}{2\sin^{-1}(D_b/D_m)} - Z + 1 \leq 0, \\ g_2(\mathbf{x}) = 2D_b - K_{D\min}(D - d) > 0, \\ g_3(\mathbf{x}) = K_{D\max}(D - d) - 2D_b \geq 0, \\ g_4(\mathbf{x}) = \zeta B_w - D_b \leq 0, \\ g_5(\mathbf{x}) = D_m - 0.5(D + d) \geq 0, \\ g_6(\mathbf{x}) = (0.5 + e)(D + d) - D_m \geq 0, \\ g_7(\mathbf{x}) = 0.5(D - D_m - D_b) - \varepsilon D_b \geq 0, \\ g_8(\mathbf{x}) = f_i \geq 0.515, \\ g_9(\mathbf{x}) = f_o \geq 0.515. \end{array} \right\} \text{Subject to} \quad (39)$$

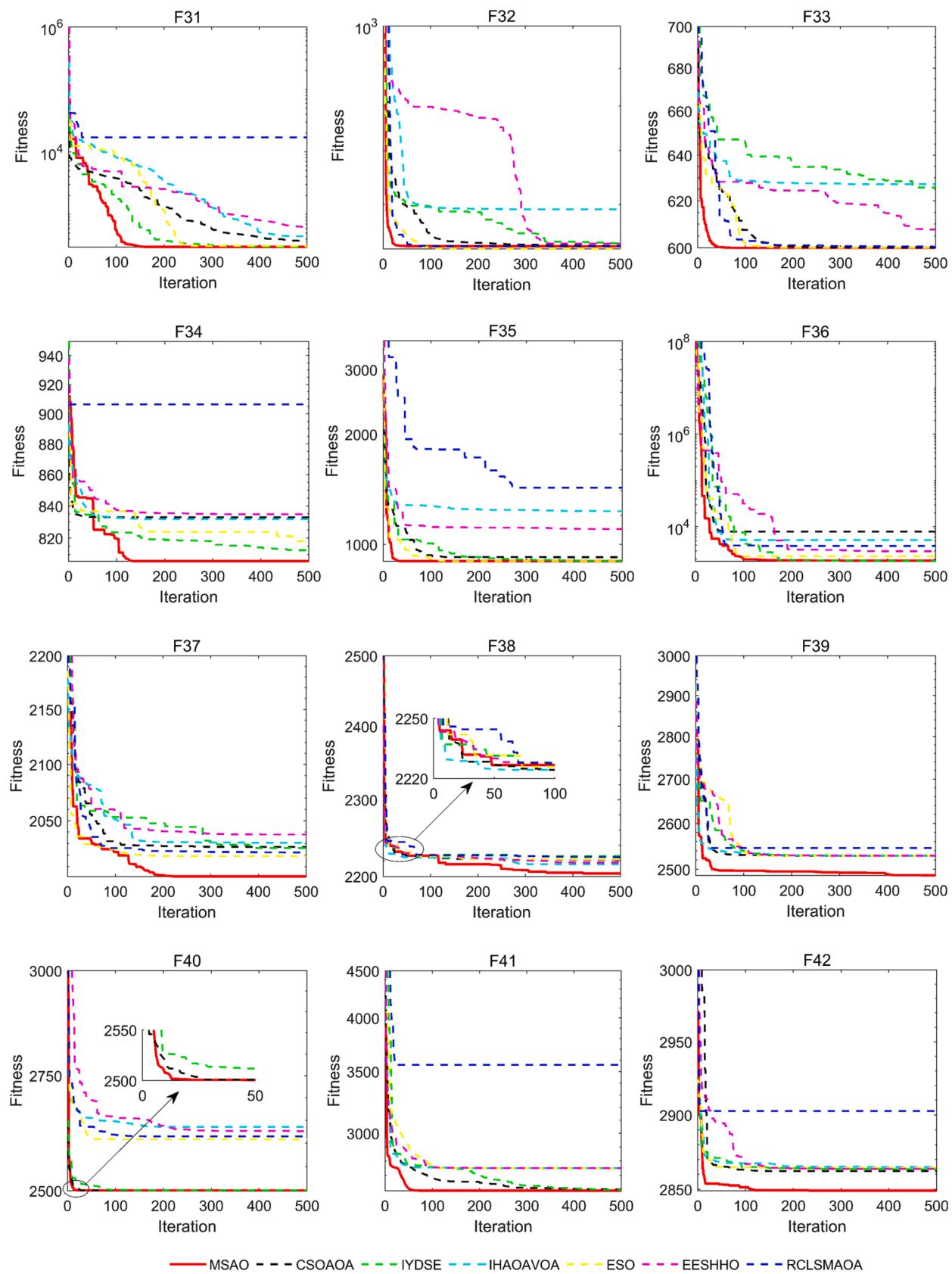
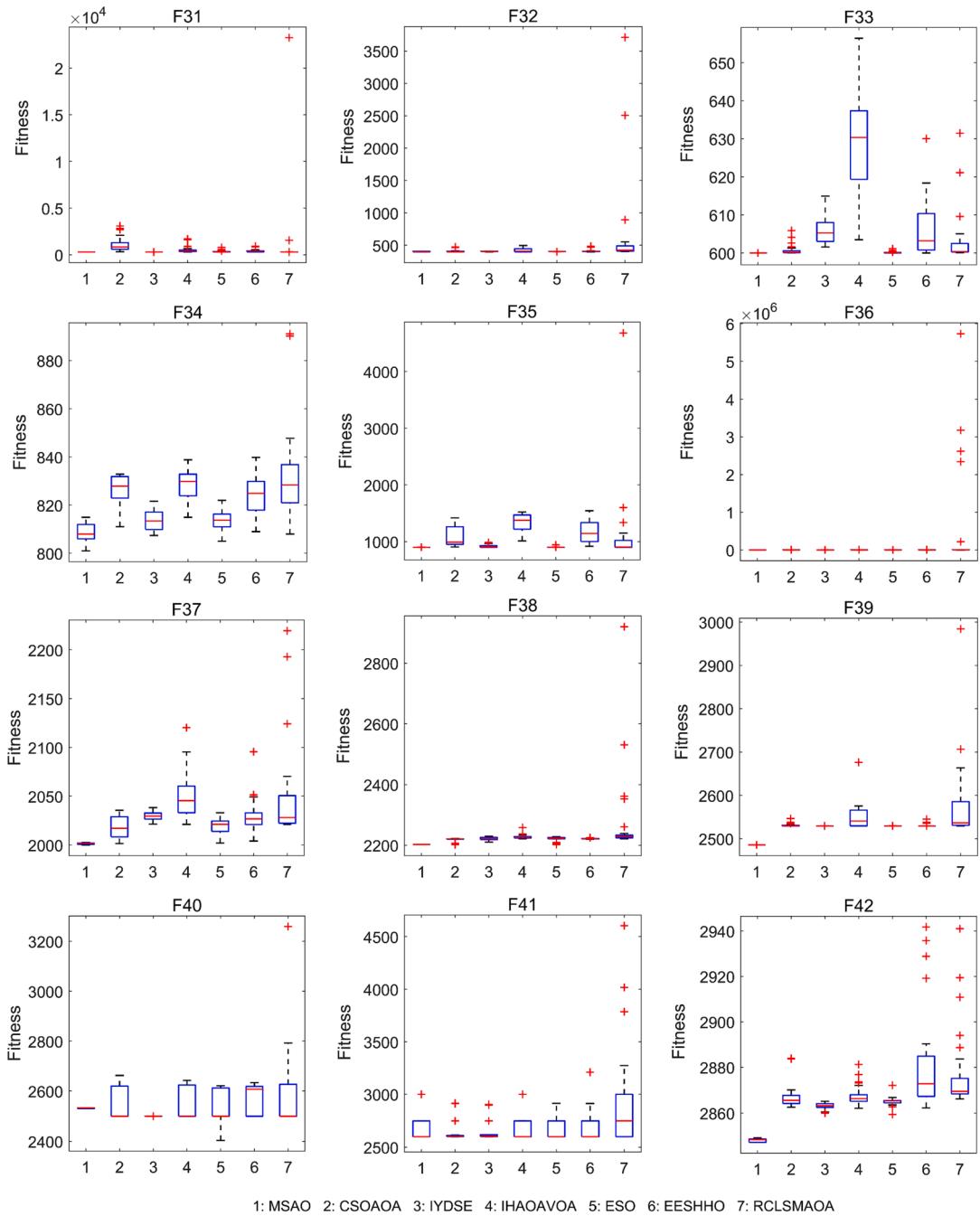


Fig. 16. Convergence curves of different modified optimization algorithms on 12 CEC2022 functions.



1: MSAO 2: CSOAOA 3: IYDSE 4: IHАОAVOA 5: ESO 6: EESHHO 7: RCLSMAOA

Fig. 17. Boxplots of different modified optimization algorithms on 12 CEC2022 functions.

Table 13

Average computation time of MSAO and other modified methods on 12 CEC2022 functions (unit: s).

Function	MSAO	CSOAOA	IYDSE	IHАОAVOA	ESO	EESHHO	RCLSMAOA
F_{31}	3.5610E-01	3.7522E-01	3.5220E-01	5.2674E-01	4.5568E-01	2.2219E-01	7.8844E-01
F_{32}	2.2989E-01	2.0425E-01	1.3231E-01	2.5732E-01	1.9178E-01	9.0971E-02	3.2910E-01
F_{33}	2.0283E-01	1.7817E-01	1.5004E-01	2.2145E-01	1.8699E-01	9.3927E-02	3.8992E-01
F_{34}	1.2326E-01	1.1429E-01	1.0607E-01	1.6277E-01	1.3662E-01	8.3608E-02	2.8800E-01
F_{35}	1.3312E-01	1.2030E-01	1.0436E-01	1.3588E-01	1.2829E-01	6.8725E-02	2.7478E-01
F_{36}	1.1489E-01	9.6211E-02	7.5464E-02	1.3823E-01	9.9416E-02	6.0639E-02	2.4042E-01
F_{37}	2.8546E-01	2.4069E-01	1.3866E-01	1.7760E-01	1.7290E-01	7.0087E-02	3.8670E-01
F_{38}	2.7682E-01	1.9807E-01	1.4227E-01	2.1190E-01	1.6316E-01	8.6715E-02	4.7687E-01
F_{39}	1.9451E-01	1.6371E-01	1.2246E-01	1.8522E-01	1.3511E-01	7.2223E-02	3.6631E-01
F_{40}	1.7272E-01	1.4658E-01	1.1191E-01	1.7244E-01	1.3310E-01	6.7836E-02	3.3744E-01
F_{41}	2.8557E-01	2.4827E-01	1.7129E-01	2.3671E-01	1.6963E-01	9.0746E-02	4.3952E-01
F_{42}	2.5341E-01	2.2933E-01	1.8806E-01	2.2636E-01	1.7689E-01	9.3315E-02	4.7393E-01

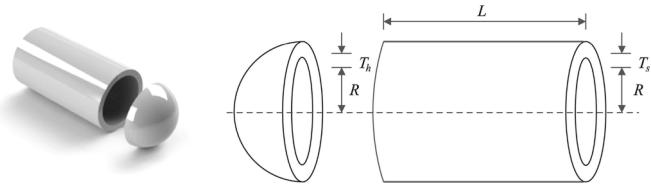


Fig. 18. Graphical representation illustrating the design of a pressure vessel.

$$\left\{ \begin{array}{l} f_c = 37.91 \left[1 + \left\{ 1.04 \left(\frac{1-\gamma}{1+\gamma} \right)^{1.72} \left(\frac{f_i(2f_o-1)}{f_o(2f_i-1)} \right)^{0.41} \right\}^{10/3} \right]^{-0.3} \times \left[\frac{\gamma^{0.3}(1-\gamma)^{1.39}}{(1+\gamma)^{1/3}} \right] \left[\frac{2f_i}{2f_i-1} \right]^{0.41}, \\ \phi_0 = 2\pi - \cos^{-1} \frac{\{(D-d)/2 - 3(T/4)\}^2 + \{D/2 - T/4 - D_b\}^2 - \{d/2 + T/4\}^2}{2\{(D-d)/2 - 3(T/4)\}\{D/2 - T/4 - D_b\}}, \\ \gamma = \frac{D_b}{D_m}, f_i = \frac{r_i}{D_b}, f_o = \frac{r_o}{D_b}, T = D - d - 2D_b, D = 160, d = 90, B_w = 30, r_i = r_o = 11.033. \end{array} \right. \quad (40)$$

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

Table 19 presents the numerical outcomes of MSAO and competing algorithms in relation to this issue. The proposed optimizer outperforms other well-regarded competitors because it provides the optimal design solution $\mathbf{x} = [125.7191, 21.4256, 10.9363, 0.5150, 0.5150, 0.4001, 0.6911, 0.3000, 0.0517, 0.6710]$ corresponding to the fitness value $f(\mathbf{x}) = 85549.2391$ that ranks first in all algorithms, followed by IHAOAVOA, and the third for SMA.

5.7. Parameter identification problem for photovoltaic models

Solar energy stands as a renewable power source endowed with abundant, clean, and eco-friendly features, rendering it a compelling alternative to conventional fossil fuel resources. Photovoltaic (PV) systems are the most common tool for converting solar energy into electricity. Further, the performance of PV systems is primarily determined by the selected PV model and uncertain model parameters, so it is an essential optimization mission to estimate accurate model parameters based on measured current–voltage data to design an efficient PV

system. In this part, the proposed MSAO will be employed to identify the core parameters of the single diode model (SDM), a classical PV model in practice.

Fig. 24 illustrates the equivalent circuit diagram of the SDM model, where there are five core variables that need to be estimated: photocurrent source (I_{ph}), reverse saturation current (I_{sd}), series resistance (R_s), parallel resistance (R_{sh}), and diode ideal factor (n). On the basis of Kirchhoff's current law, the output current (I_L) can be calculated as in the following equation:

$$I_L = I_{ph} - I_{sh} - I_d = I_{ph} - \frac{V_L + I_L \cdot R_s}{R_{sh}} - I_{sd} \cdot \left[\exp \left(\frac{q \cdot (V_L + R_s \cdot I_L)}{n \cdot k \cdot T} \right) - 1 \right] \quad (41)$$

where I_{sh} represents the parallel resistance current, I_d denotes the diode current, V_L denotes the output voltage, q is the electron charge ($1.60217646E-19$), k indicates the Boltzmann's constant ($1.3806503E-23J/k$), and T is the temperature measured in Kelvin.

The principal aim of this problem is minimizing the disparity between the experimental data estimated by an algorithm and the real measured data as much as possible, thus the root mean square error (RMSE) metric is adopted as the objective function as follows:

$$\text{minRMSE}(\mathbf{x}) = \sqrt{\frac{1}{N} \sum_{i=1}^N f_i(\mathbf{x}, I_L, V_L)^2} \quad (42)$$

where N signifies the quantity of experimental data and \mathbf{x} denotes the solution vector of the five parameters defined in Eq. (43).

Table 14
Experimental results of pressure vessel design.

Algorithm	Value of optimal variable				Minimum cost	Ranking
	$T_s(x_1)$	$T_h(x_2)$	$R(x_3)$	$L(x_4)$		
MSAO	0.7782	0.3846	40.3196	200.0000	5885.3328	1
AOA	1.0419	0.7044	52.7832	200.0000	12159.2145	19
SMA	0.7786	0.3849	40.3439	199.6630	5886.1388	6
AO	0.8738	0.4393	45.0088	146.2908	6198.7004	16
ARO	0.7790	0.3852	40.3427	199.6861	5890.1823	9
GJO	0.7804	0.3920	40.4048	198.8340	5911.4636	13
TSA	0.7898	0.4604	40.5428	200.0000	6228.0600	17
DO	0.7786	0.3849	40.3417	199.6935	5886.3659	7
SO	0.7782	0.3847	40.3201	200.0000	5885.7773	5
SAO	0.7831	0.3871	40.5726	196.5081	5893.7309	10
CMA-ES	0.8133	0.4153	41.7030	181.5993	6045.3125	15
LSHADe-SPACMA	0.7795	0.3856	40.3851	199.1068	5889.3456	8
LSHADe-cnEpSin	0.9221	0.5421	47.3759	120.6300	6567.2984	18
CSOAOA	0.7782	0.3847	40.3199	199.9978	5885.3843	4
YDZE	0.7799	0.3855	40.3881	199.1947	5894.4969	11
IHAOAVOA	0.7835	0.3872	40.5835	196.3596	5896.3469	12
ESO	0.7782	0.3846	40.3196	200.0000	5885.3330	2
EESHHO	0.8136	0.4022	42.1565	175.9179	5948.7368	14
RCLSMMAO	0.7782	0.3847	40.3197	199.9993	5885.3405	3

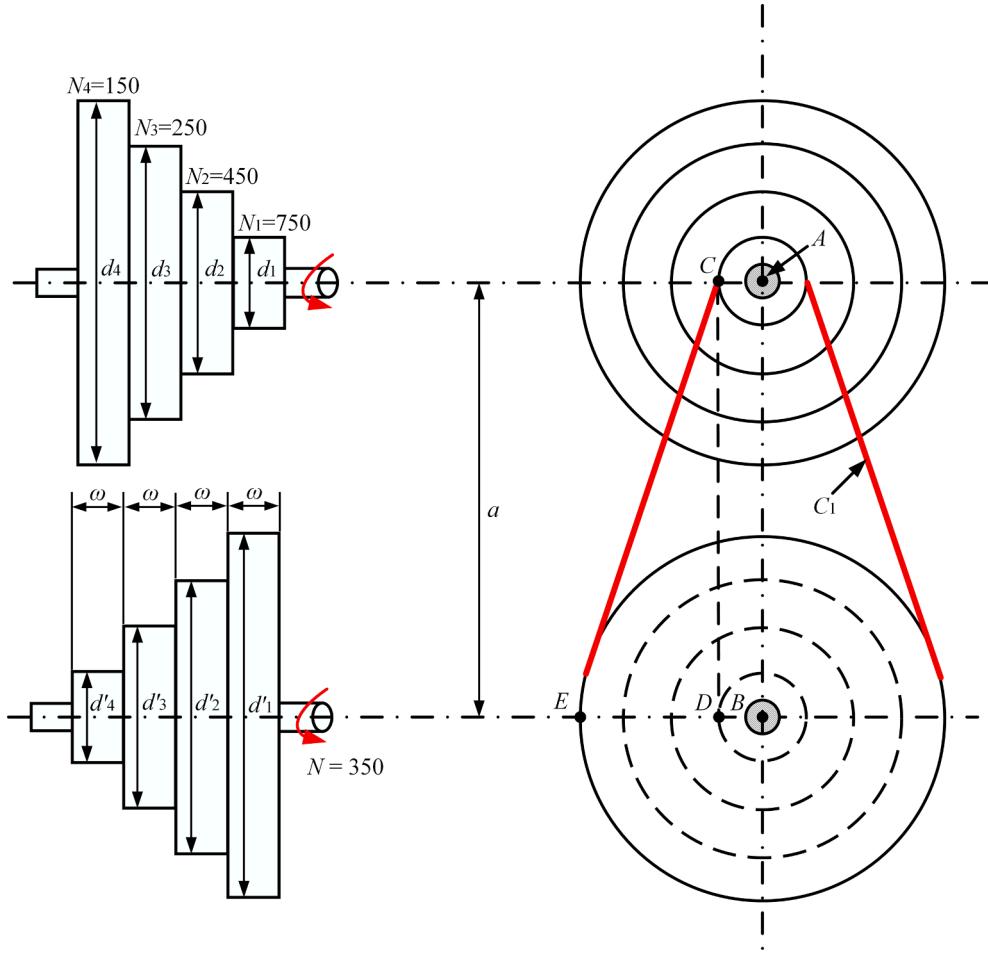


Fig. 19. Graphical representation illustrating the design of a step-cone pulley.

Table 15
Experimental results of step-cone pulley design.

Algorithm	Value of optimal variable					Minimum weight	Ranking
	$d_1(x_1)$	$d_2(x_2)$	$d_3(x_3)$	$d_4(x_4)$	$\omega(x_5)$		
MSAO	38.4140	52.8586	70.4727	84.4957	90.0000	16.0903	1
AOA	40.9168	56.3054	75.0676	90.0000	84.4856	17.1445	11
SMA	40.6568	55.9473	74.5902	89.4282	85.0237	17.0282	8
AO	40.6750	55.9723	74.6236	89.4682	85.5603	17.1508	12
ARO	40.9090	56.2946	75.0532	89.9827	85.9065	17.4190	13
GJO	40.5986	55.8671	74.4833	89.3001	85.1463	17.0059	7
TSA	40.6321	55.9132	74.5448	89.3737	90.0000	18.0027	18
DO	40.9130	56.3001	75.0606	89.9916	86.3196	17.5062	14
SO	40.9032	56.2866	75.0426	89.9700	84.5267	17.1343	10
SAO	40.8061	56.1528	74.8643	89.7564	87.0697	17.5663	16
CMA-ES	40.8184	56.1699	74.8870	89.7836	84.8240	17.1234	9
LSHADE-SPACMA	39.6517	54.5633	72.7444	87.2177	87.1834	16.8484	6
LSHADE-cnEpSin	39.3144	54.0986	72.1256	86.4758	87.9451	16.4696	3
CSOAOA	40.8998	56.2820	75.0364	89.9626	86.4063	17.5125	15
IYDSE	39.0243	53.6992	71.5932	85.8381	89.9938	16.6047	5
IHAOAVOA	40.6164	55.8917	74.5161	89.3394	90.0000	17.9888	17
ESO	38.4142	52.8589	70.4731	84.4961	90.0000	16.0913	2
EESHHO	39.5666	54.4459	72.5887	87.0306	87.3717	16.5721	4
RCLSMOA	40.9168	56.3054	75.0676	90.0000	90.0000	18.2560	19

$$\left\{ \begin{array}{l} f_i(\mathbf{x}, I_L, V_L) = I_{ph} - I_{sd} \cdot \left[\exp\left(\frac{q \cdot (V_L + R_s \cdot I_L)}{n \cdot k \cdot T}\right) - 1 \right] - \frac{V_L + I_L \cdot R_s}{R_{sh}} - I_L, \\ \mathbf{x} = \{I_{ph}, I_{sd}, R_s, R_{sh}, n\}, \\ 0 \leq I_{ph} \leq 1, 0 \leq I_{sd} \leq 1, 0 \leq R_s \leq 0.5, 0 \leq R_{sh} \leq 100, 1 \leq n \leq 2. \end{array} \right. \quad (43)$$

The benchmark current–voltage data are obtained from [104], in which a commercial R.T.C French silicon solar cell with 57 mm diameter was tested under 1000W/m^2 at 33°C . Table 20 summarizes the optimal results obtained by MSAO and other meta-heuristics after 30 independent runs. From this table, the search performance of our proposed MSAO ranks first in all algorithms because it reveals the minimum RMSE

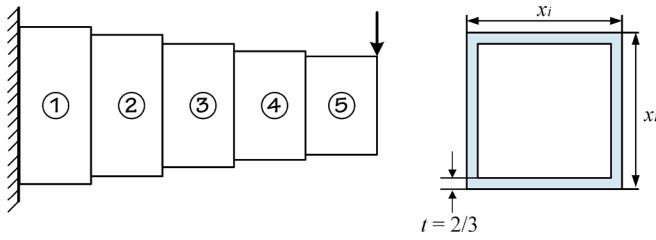


Fig. 20. Graphical representation illustrating the design of a cantilever beam.

value of 7.7299E-04, followed by ARO and SAO. Fig. 25 illustrates the difference between measured data as well as experimental data achieved by MSAO on SDM. It is obvious that the experimental data derived from MSAO perfectly fit the measured data. These results demonstrate that the proposed MSAO can also present great advantages in solving the more practical problem of parameter identification for PV models, which is attributed to its co-enhanced exploration and exploitation capabilities by the multi-strategy.

6. Discussion

The above results show that the MSAO proposed in this paper has better overall optimization performance compared with other state-of-the-art MAs, including AOA, SMA, AO, ARO, GJO, TSA, DO, SO, SAO, CMA-ES, LSHADE-SPACMA, LSHADE-cnEpSin, CSOAOA, IYDSE, IHAAOAVOA, ESO, EESHHO, and RCLSMAOA. As the optimization problem becomes more complex, the effectiveness of MSAO remains

unchanged, which suggests that it is a stable and powerful tool for addressing challenging search domains. The results are summarized as follows:

- (i) IEEE CEC2017 benchmark functions
 - (a) When the crossover probability in the DE strategy is set to 0.8, MSAO performs more consistently in most test cases.
 - (b) Each single improvement strategy introduced in this paper can enhance the search capability of MSAO to a certain extent, but through reasonable combination, they exert a synergistic enhancement effect.
 - (c) MSAO provides optimal solutions on 17, 15, 17, and 14 out of 29 10-dimensional, 30-dimensional, 50-dimensional, and 100-dimensional CEC2017 functions respectively, and achieves the smallest overall ranking of 1 among all comparison algorithms.
 - (d) The convergence speed and stability of MSAO are significantly improved over SAO.
- (ii) IEEE CEC2022 benchmark functions
 - (a) MSAO obtains better fitness and standard deviation results on 10 of 12 CEC2022 functions than other six latest modified optimizers.
- (iii) Real-world engineering optimization issues
 - (a) Pressure vessel design problem: The optimal function value is 5885.3328, obtained using MSAO.
 - (b) Step-cone pulley design problem: The optimal function value is 16.0903, achieved using MSAO.

Table 16
Experimental results of cantilever beam design.

Algorithm	Value of optimal variable					Minimum weight	Ranking
	x_1	x_2	x_3	x_4	x_5		
MSAO	6.01434	5.30596	4.49930	3.50500	2.14909	1.33996	1
AOA	5.25123	5.46693	5.21256	6.90457	2.18219	1.56109	18
SMA	6.00609	5.34656	4.48006	3.49051	2.15118	1.34000	5
AO	6.04026	5.35883	4.56157	3.47027	2.06777	1.34152	13
ARO	6.02650	5.31222	4.49402	3.50627	2.13502	1.33998	4
GJO	5.98313	5.32543	4.48944	3.52973	2.14691	1.34002	6
TSA	6.02926	5.58710	4.24163	3.65129	2.05936	1.34588	15
DO	5.74859	6.66848	6.94635	2.63697	3.39380	1.58460	19
SO	6.06836	5.33920	4.38192	3.54630	2.14856	1.34062	10
SAO	5.92612	5.33074	4.48447	3.54666	2.20026	1.34087	12
CMA-ES	5.90216	5.37114	4.59092	3.47157	2.14829	1.34061	9
LSHADE-SPACMA	6.06203	5.28702	4.48110	3.50860	2.13701	1.34009	7
LSHADE-cnEpSin	5.38869	5.30000	5.13086	3.72694	2.26768	1.36120	16
CSOAOA	5.97666	5.27088	4.46442	3.64126	2.13275	1.34072	11
IYDSE	6.01727	5.31250	4.49458	3.50319	2.14617	1.33997	2
IHAOOAVOA	5.98302	5.31494	4.46297	3.51690	2.19868	1.34013	8
ESO	6.01279	5.31505	4.50442	3.49855	2.14302	1.33997	2
EESHHO	5.98735	5.45708	4.40769	3.60751	2.04376	1.34181	14
RCLSMAOA	5.16473	4.96805	4.85192	4.75384	4.47401	1.51086	17

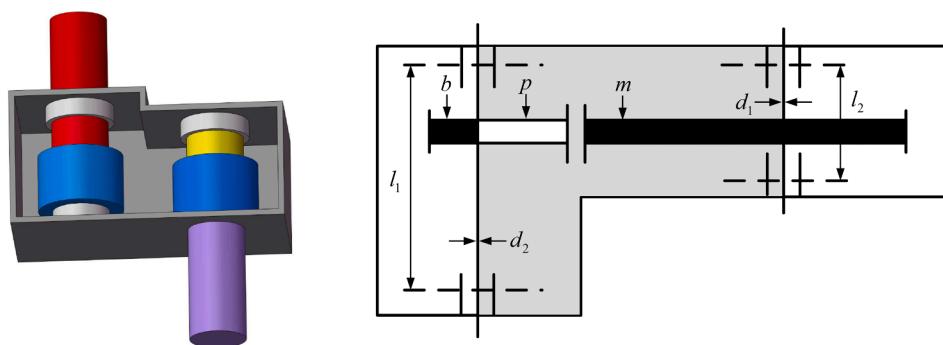
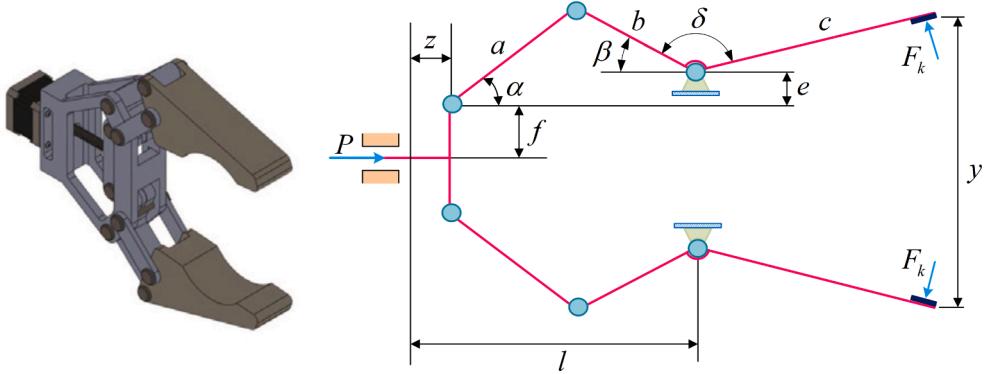


Fig. 21. Graphical representation illustrating the design of a speed reducer.

Table 17

Experimental results of speed reducer design.

Algorithm	Value of optimal variable							Minimum weight	Ranking
	$b(x_1)$	$m(x_2)$	$p(x_3)$	$l_1(x_4)$	$l_2(x_5)$	$d_1(x_6)$	$d_2(x_7)$		
MSAO	3.5000	0.7000	17.0000	7.3000	7.7153	3.3502	5.2867	2994.4711	1
AOA	3.5286	0.7000	17.0000	7.3000	8.3000	3.4355	5.3791	3100.9353	19
SMA	3.5000	0.7000	17.0000	7.3000	7.7153	3.3502	5.2867	2994.4742	5
AO	3.5669	0.7000	17.0000	7.8112	7.7555	3.3964	5.2925	3041.9155	16
ARO	3.5000	0.7000	17.0000	7.3002	7.7161	3.3503	5.2867	2994.5115	9
GJO	3.5035	0.7000	17.0000	7.5624	7.9661	3.3617	5.2908	3009.2790	15
TSA	3.5611	0.7000	17.0000	7.3000	8.3000	3.3623	5.3070	3047.4553	18
DO	3.5000	0.7000	17.0000	7.3093	7.7255	3.3503	5.2867	2994.8348	10
SO	3.5003	0.7000	17.0000	7.3000	7.7439	3.3520	5.2880	2996.5242	12
SAO	3.5344	0.7000	17.0009	7.3000	8.1379	3.3694	5.3207	3044.1968	17
CMA-ES	3.5156	0.7000	17.0000	7.3000	7.7466	3.3502	5.2874	3001.7774	13
LSHADE-SPACMA	3.5000	0.7000	17.0000	7.3000	7.7153	3.3502	5.2867	2994.4711	1
LSHADE-cnEpSin	3.5021	0.7000	17.0023	7.3000	8.2406	3.3576	5.2868	3009.2377	14
CSOAOA	3.5000	0.7000	17.0000	7.3000	7.7153	3.3502	5.2867	2994.4751	6
IVDSE	3.5001	0.7000	17.0001	7.3004	7.7207	3.3509	5.2870	2995.0160	11
IHAOAVOA	3.5000	0.7000	17.0000	7.3002	7.7155	3.3502	5.2867	2994.4762	7
ESO	3.5000	0.7000	17.0000	7.3000	7.7153	3.3502	5.2867	2994.4711	1
EESHHO	3.5000	0.7000	17.0000	7.3000	7.7153	3.3502	5.2867	2994.4711	1
RCLSMOA	3.0579	0.7458	22.0370	7.7579	7.7579	3.3579	5.2290	2994.5045	8

**Fig. 22.** Graphical representation illustrating the design of a robot gripper [103]**Table 18**

Experimental results of robot gripper design.

Algorithm	Value of optimal variable							Minimum cost	Ranking
	$a(x_1)$	$b(x_2)$	$c(x_3)$	$e(x_4)$	$f(x_5)$	$l(x_6)$	$\delta(x_7)$		
MSAO	149.8581	148.4855	199.8686	1.2473	140.7560	101.5692	2.2817	2.5687	1
AOA	150.0000	123.8116	162.1146	18.0495	150.0000	183.0802	2.9354	6.5436	13
SMA	150.0000	139.1227	169.7017	3.9169	10.0000	175.7682	1.8286	4.4483	9
AO	143.8816	112.4018	142.8917	20.3037	86.2696	191.7535	2.6406	10.3308	17
ARO	131.8870	125.7174	198.6461	5.3494	139.8001	121.7154	2.4334	3.4472	4
GJO	59.2115	59.2115	135.1511	17.5756	59.2115	170.3022	1.7522	6.7940	15
TSA	110.7648	19.6127	100.0000	17.1652	71.2075	100.0000	1.6186	17.2967	19
DO	149.5640	109.5397	168.3586	37.6933	82.1315	141.7112	2.4705	4.3916	8
SO	132.1998	98.8255	126.1442	30.4539	74.1473	138.6498	2.6117	6.7828	14
SAO	132.2238	101.9160	134.0481	13.9668	101.2685	195.7287	2.8501	14.8343	18
CMA-ES	150.0000	123.3546	124.5663	6.2477	31.3858	199.6126	2.0406	7.2414	16
LSHADE-SPACMA	106.3406	103.5359	146.5570	0.0182	127.9335	130.5166	2.6864	6.4175	12
LSHADE-cnEpSin	119.6456	119.4266	162.2203	0.0000	123.8360	100.0000	2.4729	4.0009	6
CSOAOA	149.8537	149.3270	192.2982	0.3728	12.1472	104.2083	1.6133	2.7124	2
IVDSE	150.0000	144.1532	130.6429	2.8904	60.8901	157.6817	2.0858	5.5945	10
IHAOAVOA	149.7301	149.6216	124.5629	0.0000	149.6851	100.0000	2.8279	4.0661	7
ESO	98.0985	98.0985	162.9275	31.4637	98.0985	225.8550	2.3467	3.8368	5
EESHHO	147.9991	138.2204	199.1008	9.5030	91.6432	110.2664	2.1067	2.8401	3
RCLSMOA	113.2082	113.2082	173.7202	36.8601	113.2082	247.4403	2.5776	5.6023	11

- (c) Cantilever beam design problem: MSAO reveals the optimal target weight value of 1.33996.
- (d) Speed reducer design problem: The optimal function value is 2994.4711, obtained using the MSAO, LSHADE-SPACMA, ESO, and EESHHO algorithms.

- (e) Robot gripper design problem: MSAO provided a minimum cost of 2.5687, which is 5.30 % less than the second-ranked CSOAOA.
- (f) Rolling element bearing design problem: MSAO attains the maximum objective function value of 85549.2391.

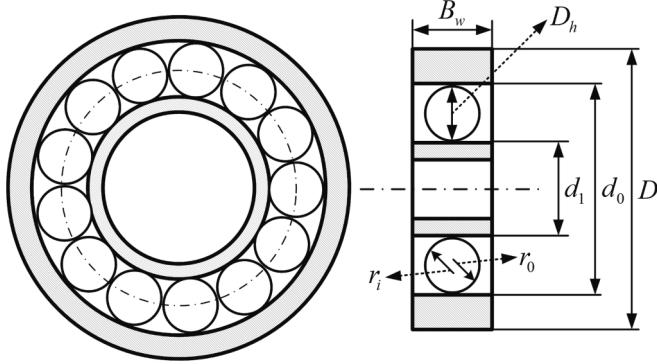


Fig. 23. Graphical representation illustrating the design of a rolling element bearing.

- (g) For the parameter identification problem of single diode PV model, MSAO provides the minimum RMSE value of 7.7299E-04.

The reason for these promising outcomes achieved by MSAO is largely attributable to more robust exploration & exploitation capabilities associated with the four improvement strategies employed in this paper. The good point set initialization strategy generates a uniformly distributed high-quality population in the search space, thus guiding the algorithm to rapidly find the promising region with global optimal solutions. The greedy selection strategy establishes a smooth balance between exploration and exploitation, which preserves the current better candidate solution in both exploration and development phases. This allows MSAO for stable performance in solving simple as well as complex optimization problems. The DE operators strengthen the search process pouring more local exploitation in subsequent iterations to get closer to the global optimum. Moreover, embedding the DLOBL strategy helps to jump out of local optima in later iterations. MSAO is still only a single-objective algorithm. This study is limited to selected cases, which can extend the application of MSAO to solve single-objective optimization problems related to intelligent computing, such as hyper-parameter tuning, feature selection, and so on.

7. Conclusion and future directions

A modified version of snow ablation optimizer named MSAO is proposed in this paper for solving numerical optimization and real-

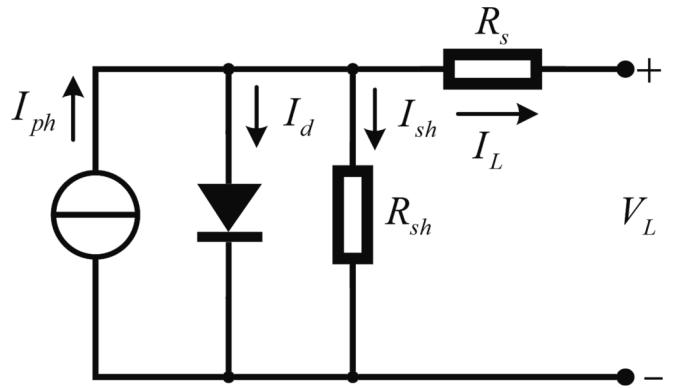


Fig. 24. Equivalent circuit diagram for single diode model (SDM).

world engineering challenges. In MSAO, four improvement strategies are integrated to synergistically elevate the overall optimization performance. First, the good point set strategy is introduced in the initialization stage to generate a uniformly distributed premium population, which increases the convergence speed and global exploration trend to a certain extent. Then, the greedy selection strategy is adopted to establish a stable exploration-exploitation tradeoff in MSAO. Following that, the DE mechanism is leveraged in later stages to update the positions of all individuals to broaden the search domain and raise the solution accuracy. Lastly, to override the weakness of the original algorithm, which is prone to local optimal stagnation, a novel DLOBL strategy is designed to perform a dimension-by-dimension inverse renewal of the current optimal solution. To verify the effectiveness of MSAO, a series of numerical experiments are carried out using 29 CEC2017 and 12 CEC2022 benchmark functions. Compared with various state-of-the-art MAs, the results demonstrate that MSAO can provide better solution accuracy, convergence rate, stability, and dimensional scalability in most test cases. Meanwhile, the performance of MSAO in solving seven engineering optimization problems is fairly competitive, which highlights the practicability of MSAO for real-world applications even more.

Of course, there is still room to improve the optimization performance of MSAO. Due to the introduction of different improvement strategies, MSAO needs to spend more runtime, which is its main limitation to overcome in the future. We are planning to use parallel computing methods to transform the structure of MSAO in a deeper way, such as island model and cell model, so that it can spend less computational cost while ensuring convergence accuracy. Moreover, in the

Table 19
Experimental results of rolling element bearing design.

Algorithm	Value of optimal variable									Maximum value	Ranking
	$D_m(x_1)$	$D_b(x_2)$	$Z(x_3)$	$f_i(x_4)$	$f_o(x_5)$	$K_{D\min}(x_6)$	$K_{D\max}(x_7)$	$\varepsilon(x_8)$	$e(x_9)$		
MSAO	125.7191	21.4256	10.9363	0.5150	0.5150	0.4001	0.6911	0.3000	0.0517	0.6710	85549.2391
AOA	125.0000	21.1603	9.7401	0.5150	0.5150	0.4000	0.7000	0.3000	0.0200	0.6000	78519.1493
SMA	125.7191	21.4256	11.0291	0.5150	0.5150	0.4931	0.7000	0.3000	0.0356	0.6000	85549.2364
AO	128.2637	18.7206	12.4680	0.5150	0.5808	0.4334	0.6677	0.3000	0.1000	0.6000	71160.7676
ARO	125.7151	21.4247	10.8042	0.5150	0.5179	0.4944	0.6245	0.3000	0.0515	0.6059	85542.5219
GJO	125.5971	21.3867	10.7663	0.5150	0.5229	0.4423	0.6893	0.3023	0.0569	0.6198	85270.7986
TSA	125.1625	21.2618	10.8902	0.5151	0.5150	0.5000	0.6262	0.3000	0.0200	0.6000	84160.5302
DO	125.7174	21.4251	10.8044	0.5150	0.5154	0.5000	0.6508	0.3000	0.0979	0.6055	85545.7328
SO	125.0000	21.2730	10.6360	0.5150	0.5150	0.4985	0.6998	0.3201	0.0368	0.6000	84459.7537
SAO	126.1594	21.0306	10.0858	0.5150	0.5812	0.4000	0.6203	0.3000	0.0615	0.6000	77675.3925
CMA-ES	125.0000	20.8811	11.5000	0.5150	0.5150	0.4000	0.6000	0.3000	0.0410	0.6000	81725.0641
LSHADE-SPACMA	125.7121	21.4241	11.3066	0.5150	0.5150	0.4573	0.6690	0.3002	0.0513	0.6173	85538.6659
LSHADE-cnEpSin	125.0472	21.2828	11.0810	0.5150	0.5240	0.4012	0.6087	0.3044	0.0516	0.6005	84526.2493
CSOAOA	125.0013	21.2733	10.5000	0.5150	0.5150	0.4058	0.6385	0.3226	0.0206	0.6143	84461.7335
YDSE	125.7189	21.4253	10.7880	0.5150	0.5250	0.5000	0.6858	0.3000	0.0226	0.6324	85545.3114
IHAOAVOA	125.7191	21.4256	11.2393	0.5150	0.5150	0.4077	0.6423	0.3000	0.0380	0.6000	85549.2378
ESO	125.7186	21.4255	10.5500	0.5150	0.5150	0.5000	0.7000	0.3000	0.0200	0.6000	85548.5580
EESHHO	125.7226	21.4234	11.4955	0.5150	0.5150	0.4507	0.6947	0.3000	0.0633	0.7105	85533.9184
RCLSMOA	125.7190	21.4256	11.1886	0.5150	0.5173	0.4838	0.6998	0.3000	0.1000	0.6000	85548.6648

Table 20

Optimal results of MSAO and competitive methods for the parameter identification problem of SDM.

Algorithm	Value of optimal variable					RMSE	Ranking
	I_{ph} (A)	I_{sd} (μA)	R_s (Ω)	R_{sh} (Ω)	n		
MSAO	0.76079	0.31069	0.03655	52.88991	1.47727	7.7299E-04	1
AOA	0.76145	0.42362	0.03518	50.74343	1.50921	1.0687E-03	10
SMA	0.76159	0.37053	0.03513	44.18683	1.49547	1.1905E-03	13
AO	0.74724	0.44217	0.03192	38.81282	1.51973	1.3324E-02	19
ARO	0.76079	0.31821	0.03644	53.39659	1.47967	7.7399E-04	2
GJO	0.76021	0.55716	0.03218	61.08891	1.53808	2.4271E-03	16
TSA	0.76148	0.29137	0.04162	56.80689	1.47222	6.0883E-03	18
DO	0.75959	0.31388	0.03685	77.45247	1.47796	1.1720E-03	12
SO	0.76078	0.29607	0.03673	51.72812	1.47246	7.7728E-04	4
SAO	0.76077	0.32492	0.03635	53.89089	1.48178	7.7614E-04	3
CMA-ES	0.75730	0.01771	0.04951	70.16915	1.23579	1.8557E-03	15
LSHADE-SPACMA	0.76077	0.32940	0.03629	54.21513	1.48316	7.7836E-04	5
LSHADE-cnEpSin	0.72997	0.01105	0.05888	26.69397	1.21299	5.3779E-03	17
CSOOOA	0.76060	0.17922	0.03899	45.50980	1.42405	1.1632E-03	11
IYDSE	0.76075	0.31740	0.03651	53.35019	1.47944	7.7948E-04	6
IHAOAVOA	0.76104	0.45913	0.03470	56.47131	1.51760	1.0321E-03	9
ESO	0.76054	0.42532	0.03513	62.30928	1.50940	9.2416E-04	8
EESHHO	0.76083	0.28029	0.03699	50.71575	1.46702	7.8927E-04	7
RCLMSAOA	0.75927	0.48924	0.03450	86.22398	1.52392	1.3459E-03	14

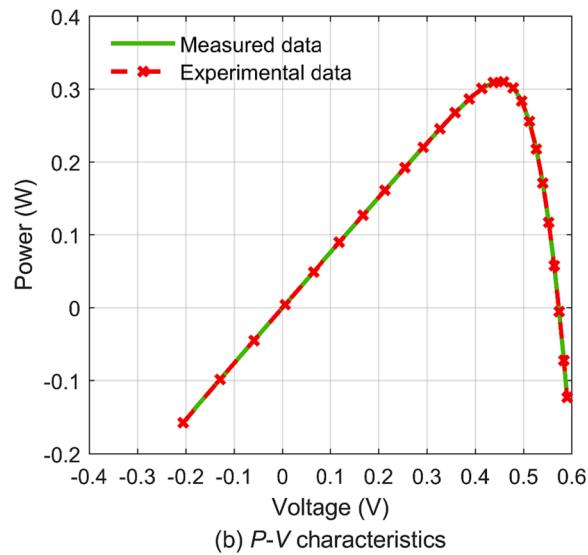
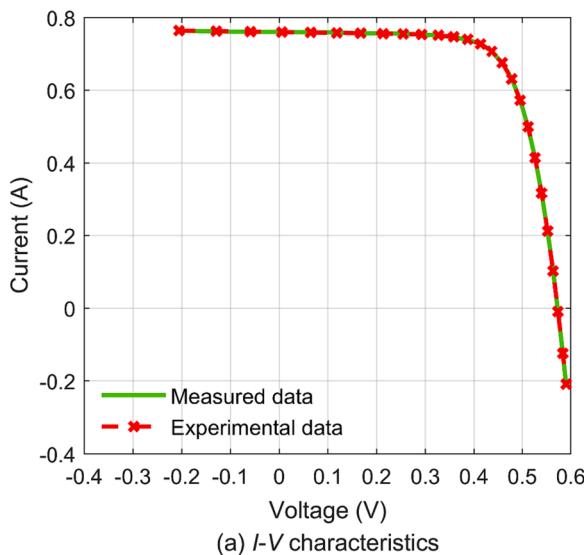


Fig. 25. Curve fitting for the data derived from measurements and experiments identified by MSAO.

crossover operator of DE, a fixed crossover probability is used. Although this widens the search space and increases population diversity, it is not guaranteed that this value will lead to the algorithm's optimal performance in all optimization problems. Consequently, we will endeavor to propose an adaptive crossover probability adjustment mechanism based on individual fitness to further augment the exploration and exploitation abilities of MSAO. To strengthen the algorithm's robustness, some other modification operators such as dynamic population evolution and quantum rotation gate can also be merged into MSAO. Furthermore, in light of the promising consequences achieved by the proposed algorithm, MSAO can be utilized to address real-world optimization projects arising in a broader range of disciplines, such as forecast modeling, image segmentation, feature selection, path planning, cloud job scheduling, and parameter self-tuning for a PID controller. Developing a multi-objective version of MSAO to address more complex multi-objective optimization challenges can also be attempted.

CRediT authorship contribution statement

Yaning Xiao: Conceptualization, Methodology, Writing – original

draft, Writing – review & editing. **Hao Cui:** Formal analysis, Software, Writing – review & editing. **Abdelazim G. Hussien:** Investigation, Validation. **Fatma A. Hashim:** Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data used to support the findings of this study are included within the article.

Acknowledgements

This study is financially supported by the Southern University of Science and Technology Graduate Student Innovation and Practice Fund under Grant Y70300002.

Appendix A

Table A1

Overview of IEEE CEC2017 benchmark suite.

Type	Function	Name	Dimension	Range	F_{\min}
Unimodal functions	F_1	Shifted and Rotated Bent Cigar Function	10	[−100, 100]	100
	F_2	Shifted and Rotated Sum of Different Power Function*	10	[−100, 100]	200
	F_3	Shifted and Rotated Zakharov Function	10	[−100, 100]	300
Multimodal functions	F_4	Shifted and Rotated Rosenbrock's Function	10	[−100, 100]	400
	F_5	Shifted and Rotated Rastrigin's Function	10	[−100, 100]	500
	F_6	Shifted and Rotated Expanded Scaffer's F6 Function	10	[−100, 100]	600
	F_7	Shifted and Rotated Lunacek Bi-Rastrigin Function	10	[−100, 100]	700
	F_8	Shifted and Rotated Non-Continuous Rastrigin's Function	10	[−100, 100]	800
	F_9	Shifted and Rotated Levy Function	10	[−100, 100]	900
	F_{10}	Shifted and Rotated Schwefel's Function	10	[−100, 100]	1000
	F_{11}	Hybrid Function 1 ($N=3$)	10	[−100, 100]	1100
	F_{12}	Hybrid Function 2 ($N=3$)	10	[−100, 100]	1200
	F_{13}	Hybrid Function 3 ($N=3$)	10	[−100, 100]	1300
Hybrid functions	F_{14}	Hybrid Function 4 ($N=4$)	10	[−100, 100]	1400
	F_{15}	Hybrid Function 5 ($N=4$)	10	[−100, 100]	1500
	F_{16}	Hybrid Function 6 ($N=4$)	10	[−100, 100]	1600
	F_{17}	Hybrid Function 6 ($N=5$)	10	[−100, 100]	1700
	F_{18}	Hybrid Function 6 ($N=5$)	10	[−100, 100]	1800
	F_{19}	Hybrid Function 6 ($N=5$)	10	[−100, 100]	1900
	F_{20}	Hybrid Function 6 ($N=6$)	10	[−100, 100]	2000
	F_{21}	Composition Function 1 ($N=3$)	10	[−100, 100]	2100
	F_{22}	Composition Function 2 ($N=3$)	10	[−100, 100]	2200
	F_{23}	Composition Function 3 ($N=4$)	10	[−100, 100]	2300
Composition functions	F_{24}	Composition Function 4 ($N=4$)	10	[−100, 100]	2400
	F_{25}	Composition Function 5 ($N=5$)	10	[−100, 100]	2500
	F_{26}	Composition Function 6 ($N=5$)	10	[−100, 100]	2600
	F_{27}	Composition Function 7 ($N=6$)	10	[−100, 100]	2700
	F_{28}	Composition Function 8 ($N=6$)	10	[−100, 100]	2800
	F_{29}	Composition Function 9 ($N=3$)	10	[−100, 100]	2900
	F_{30}	Composition Function 10 ($N=3$)	10	[−100, 100]	3000

* F_2 has been officially removed due to its unstable behavior on high-dimensional issues.

Table A2

Overview of IEEE CEC2022 benchmark suite.

Type	Function	Name	Dimensions	Range	F_{\min}
Unimodal function	F_{31}	Shifted and full Rotated Zakharov Function	10	[−100, 100]	300
Multimodal functions	F_{32}	Shifted and full Rotated Rosenbrock's Function	10	[−100, 100]	400
	F_{33}	Shifted and full Rotated Expanded Schaffer's f_6 Function	10	[−100, 100]	600
	F_{34}	Shifted and full Rotated Non-Continuous Rastrigin's Function	10	[−100, 100]	800
	F_{35}	Shifted and full Rotated Levy Function	10	[−100, 100]	900
Hybrid functions	F_{36}	Hybrid Function 1 ($N = 3$)	10	[−100, 100]	1800
	F_{37}	Hybrid Function 2 ($N = 6$)	10	[−100, 100]	2000
	F_{38}	Hybrid Function 3 ($N = 5$)	10	[−100, 100]	2200
Composition functions	F_{39}	Composition Function 1 ($N = 5$)	10	[−100, 100]	2300
	F_{40}	Composition Function 2 ($N = 4$)	10	[−100, 100]	2400
	F_{41}	Composition Function 3 ($N = 5$)	10	[−100, 100]	2600
	F_{42}	Composition Function 4 ($N = 6$)	10	[−100, 100]	2700

References

- [1] Y. Xiao, X. Sun, Y. Guo, S. Li, Y. Zhang, Y. Wang, An improved gorilla troops optimizer based on lens opposition-based learning and adaptive β-Hill climbing for global optimization. CMES-Comp. Model. Eng. Sci. 131 (815–850) (2022), <https://doi.org/10.32604/cmes.2022.019198>.
- [2] J. Yao, Y. Sha, Y. Chen, G. Zhang, X. Hu, G. Bai, J. Liu, IHSSAO: An improved hybrid salp swarm algorithm and aquila optimizer for UAV path planning in complex terrain, Appl. Sci.-Basel 12 (5634) (2022), <https://doi.org/10.3390/app12115634>.
- [3] J. Li, W. Chen, K. Han, Q. Wang, Fault diagnosis of rolling bearing based on GA-VMD and improved WOA-LSSVM, IEEE Access 8 (2020) 166753–166767, <https://doi.org/10.1109/ACCESS.2020.3023306>.
- [4] Q. Liu, N. Li, H. Jia, Q. Qi, L. Abualigah, A chimp-inspired remora optimization algorithm for multilevel thresholding image segmentation using cross entropy, Artif. Intell. Rev. (2023), <https://doi.org/10.1007/s10462-023-10498-0>.
- [5] M. Azizi, S.A. Mousavi Ghasemi, R.G. Ejlali, S. Talatohari, Optimum design of fuzzy controller using hybrid ant lion optimizer and Jaya algorithm, Artif. Intell. Rev. 53 (2020) 1553–1584, <https://doi.org/10.1007/s10462-019-09713-8>.
- [6] J. Shamir, J. Rosen, U. Mahlab, H.J. Caulfield, Optimization methods for pattern recognition, Optical Pattern Recognition: A Critical Review 10262 (1992) 3–25, <https://doi.org/10.1111/12.59846>.
- [7] A.G. Hussien, A.E. Hassanien, E.H. Houssein, Swarming behaviour of salps algorithm for predicting chemical compound activities, in: 2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS), 2017, pp. 315–320, <https://doi.org/10.1109/ICICIS.2017.8260072>.
- [8] F.A. Hashim, N. Neggaz, R.R. Mostafa, L. Abualigah, R. Damasevicius, A. G. Hussien, Dimensionality reduction approach based on modified hunger games search: case study on Parkinson's disease phonation, Neural Comput. Appl. 35 (2023) 21979–22005, <https://doi.org/10.1007/s00521-023-08936-9>.
- [9] J. Tang, X. Chen, X. Zhu, F. Zhu, Dynamic reallocation model of multiple unmanned aerial vehicle tasks in emergent adjustment scenarios, IEEE Trans. Aerosp. Electron. Syst. 1–43 (2022), <https://doi.org/10.1109/taes.2022.3195478>.
- [10] Y. Wang, Y. Xiao, Y. Guo, J. Li, Dynamic chaotic opposition-based learning-driven hybrid aquila optimizer and artificial rabbits optimization algorithm: Framework and applications, Processess 10 (2022) 2703, <https://doi.org/10.3390/pr10122703>.
- [11] K. Wang, M. Guo, C. Dai, Z. Li, Information-decision searching algorithm: Theory and applications for solving engineering optimization problems, Inf. Sci. 607 (2022) 1465–1531, <https://doi.org/10.1016/j.ins.2022.06.008>.

- [12] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp swarm algorithm: A bio-inspired optimizer for engineering design problems, *Adv. Eng. Softw.* 114 (2017) 163–191, <https://doi.org/10.1016/j.advengsoft.2017.07.002>.
- [13] G. Hu, J. Zhong, B. Du, G. Wei, An enhanced hybrid arithmetic optimization algorithm for engineering applications, *Comput. Meth. Appl. Mech. Eng.* 394 (2022) 114901, <https://doi.org/10.1016/j.cma.2022.114901>.
- [14] H. Cui, Y. Guo, Y. Xiao, Y. Wang, J. Li, Y. Zhang, H. Zhang, Enhanced Harris hawks optimization integrated with COOT bird optimization for solving continuous numerical optimization problems, *CMES-Comp. Model. Eng. Sci.* 137 (2023) 1635–1675, <https://doi.org/10.32604/cmes.2023.026019>.
- [15] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, *Knowledge-Based Syst.* 89 (2015) 228–249, <https://doi.org/10.1016/j.knosys.2015.07.006>.
- [16] A.O. Topal, O. Altun, A novel meta-heuristic algorithm: Dynamic virtual bats algorithm, *Inf. Sci.* 354 (2016) 222–235, <https://doi.org/10.1016/j.ins.2016.03.025>.
- [17] J. Zhao, S. Liu, M. Zhou, X. Guo, L. Qi, Modified cuckoo search algorithm to solve economic power dispatch optimization problems, *IEEE/CAA Journal of Automatica Sinica* 5 (2018) 794–806, <https://doi.org/10.1109/JAS.2018.7511138>.
- [18] Y. Xiao, X. Sun, Y. Guo, H. Cui, Y. Wang, J. Li, S. Li, An enhanced honey badger algorithm based on Lévy flight and refraction opposition-based learning for engineering design problems, *J. Intell. Fuzzy Syst.* 43 (2022) 4517–4540, <https://doi.org/10.3233/JIFS-213206>.
- [19] J.H. Holland, Genetic algorithms, *Sci. Am.* 267 (1992) 66–72, <https://doi.org/10.1038/scientificamerican0792-66>.
- [20] R. Storn, K. Price, Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces, *J. Glob. Optim.* 11 (1997) 341–359, <https://doi.org/10.1023/A:1008202821328>.
- [21] D. Simon, Biogeography-based optimization, *IEEE Trans. Evol. Comput.* 12 (2008) 702–713, <https://doi.org/10.1109/TEVC.2008.919004>.
- [22] H. Talbi, A. Draa, A new real-coded quantum-inspired evolutionary algorithm for continuous optimization, *Appl. Soft. Comput.* 61 (2017) 765–791, <https://doi.org/10.1016/j.asoc.2017.07.046>.
- [23] E.H. Houssein, D. Oliva, N.A. Samee, N.F. Mahmoud, M.M. Enam, Liver Cancer Algorithm: A novel bio-inspired optimizer, *Comput. Biol. Med.* 165 (2023) 107389, <https://doi.org/10.1016/j.combiomed.2023.107389>.
- [24] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, *Science* 220 (1983) 671–680, <https://doi.org/10.1126/science.220.4598.671>.
- [25] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: A gravitational search algorithm, *Inf. Sci.* 179 (2009) 2232–2248, <https://doi.org/10.1016/j.ins.2009.03.004>.
- [26] S. Mirjalili, S.M. Mirjalili, A. Hatamlou, Multi-Verse Optimizer: a nature-inspired algorithm for global optimization, *Neural Comput. Appl.* 27 (2015) 495–513, <https://doi.org/10.1007/s00521-015-1870-7>.
- [27] W. Zhao, L. Wang, Z. Zhang, Atom search optimization and its application to solve a hydrogeologic parameter estimation problem, *Knowledge-Based Syst.* 163 (2019) 283–304, <https://doi.org/10.1016/j.knosys.2018.08.030>.
- [28] A. Kaveh, A. Dadras, A novel meta-heuristic optimization algorithm: Thermal exchange optimization, *Adv. Eng. Softw.* 110 (2017) 69–84, <https://doi.org/10.1016/j.advengsoft.2017.03.014>.
- [29] H. Su, D. Zhao, A.A. Heidari, L. Liu, X. Zhang, M. Mafarja, H. Chen, RIME: A physics-based optimization, *Neurocomputing* 532 (2023) 183–214, <https://doi.org/10.1016/j.neucom.2023.02.010>.
- [30] M. Abdel-Basset, R. Mohamed, S.A.A. Azeem, M. Jameel, M. Abouhawwash, Kepler optimization algorithm: A new metaheuristic algorithm inspired by Kepler's laws of planetary motion, *Knowledge-Based Syst.* 268 (2023) 110454, <https://doi.org/10.1016/j.knosys.2023.110454>.
- [31] S. Mirjalili, SCA: A sine cosine algorithm for solving optimization problems, *Knowledge-Based Syst.* 96 (2016) 120–133, <https://doi.org/10.1016/j.knosys.2015.12.022>.
- [32] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A.H. Gandomi, The arithmetic optimization algorithm, *Comput. Meth. Appl. Mech. Eng.* 376 (2021) 113609, <https://doi.org/10.1016/j.cma.2020.113609>.
- [33] I. Ahmadianfar, A.A. Heidari, A.H. Gandomi, X. Chu, H. Chen, RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method, *Expert Syst. Appl.* 181 (2021) 115079, <https://doi.org/10.1016/j.eswa.2021.115079>.
- [34] I. Ahmadianfar, A.A. Heidari, S. Noshadian, H. Chen, A.H. Gandomi, INFO: An efficient optimization algorithm based on weighted mean of vectors, *Expert Syst. Appl.* 195 (2022) 116516, <https://doi.org/10.1016/j.eswa.2022.116516>.
- [35] M. Abdel-Basset, D. El-Shahat, M. Jameel, M. Abouhawwash, Exponential distribution optimizer (EDO): a novel math-inspired algorithm for global optimization and engineering problems, *Artif. Intell. Rev.* 56 (2023) 9329–9400, <https://doi.org/10.1007/s10462-023-10403-9>.
- [36] W. Zhao, L. Wang, Z. Zhang, S. Mirjalili, N. Khodadadi, Q. Ge, Quadratic Interpolation Optimization (QIO): A new optimization algorithm based on generalized quadratic interpolation and its applications to real-world engineering problems, *Comput. Meth. Appl. Mech. Eng.* 417 (2023), <https://doi.org/10.1016/j.cma.2023.116446>.
- [37] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95 - International Conference on Neural Networks 4 (1995) 1942–1948. 10.1109/ICNN.1995.488968.
- [38] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *J. Glob. Optim.* 39 (2007) 459–471, <https://doi.org/10.1007/s10898-007-9149-x>.
- [39] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Softw.* 69 (2014) 46–61, <https://doi.org/10.1016/j.advengsoft.2013.12.007>.
- [40] S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Softw.* 95 (2016) 51–67, <https://doi.org/10.1016/j.advengsoft.2016.01.008>.
- [41] G.-G. Wang, Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems, *Memetic Computing* 10 (2018) 151–164, <https://doi.org/10.1007/s12293-016-0212-3>.
- [42] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, *Futur. Gener. Comp. Syst.* 97 (2019) 849–872, <https://doi.org/10.1016/j.future.2019.02.028>.
- [43] S. Li, H. Chen, M. Wang, A.A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, *Futur. Gener. Comp. Syst.* 111 (2020) 300–323, <https://doi.org/10.1016/j.future.2020.03.055>.
- [44] L. Abualigah, D. Youssi, M. Abd Elaziz, A.A. Ewees, M.A.A. Al-qaness, A. H. Gandomi, Aquila optimizer: A novel meta-heuristic optimization algorithm, *Comput. Ind. Eng.* 157 (2021) 107250, <https://doi.org/10.1016/j.cie.2021.107250>.
- [45] L. Wang, Q. Cao, Z. Zhang, S. Mirjalili, W. Zhao, Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems, *Eng. Appl. Artif. Intell.* 114 (2022) 105082, <https://doi.org/10.1016/j.engappai.2022.105082>.
- [46] N. Chopra, M. Mohsin Ansari, Golden jackal optimization: A novel nature-inspired optimizer for engineering applications, *Expert Syst. Appl.* 198 (2022) 116924, <https://doi.org/10.1016/j.eswa.2022.116924>.
- [47] D. Manjarres, I. Landa-Torres, S. Gil-Lopez, J. Del Ser, M.N. Bilbao, S. Salcedo-Sanz, Z.W. Geem, A survey on applications of the harmony search algorithm, *Eng. Appl. Artif. Intell.* 26 (2013) 1818–1831, <https://doi.org/10.1016/j.engappai.2013.05.008>.
- [48] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems, *Comput.-Aided Des.* 43 (303–315) (2011), <https://doi.org/10.1016/j.cad.2010.12.015>.
- [49] N. Moosavian, B. Kasaee Roodsari, Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks, *Swarm Evol. Comput.* 17 (2014) 14–24, <https://doi.org/10.1016/j.swevo.2014.02.002>.
- [50] Q. Zhang, R. Wang, J. Yang, K. Ding, Y. Li, J. Hu, Collective decision optimization algorithm: A new heuristic optimization method, *Neurocomputing* 221 (2017) 123–137, <https://doi.org/10.1016/j.neucom.2016.09.068>.
- [51] M.A. Al-Betar, Z.A.A. Alyasseri, M.A. Awadallah, I. Abu Doush, Coronavirus herd immunity optimizer (CHIO), *Neural Comput. Appl.* 33 (2021) 5011–5042, <https://doi.org/10.1007/s00521-020-05296-6>.
- [52] G. Hu, Y. Guo, J. Zhong, G. Wei, IYDSE: Ameliorated Young's double-slit experiment optimizer for applied mechanics and engineering, *Comput. Meth. Appl. Mech. Eng.* 412 (2023) 116062, <https://doi.org/10.1016/j.cma.2023.116062>.
- [53] S. Kaur, L.K. Awasthi, A.L. Sangal, G. Dhiman, Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization, *Eng. Appl. Artif. Intell.* 90 (2020) 103541, <https://doi.org/10.1016/j.engappai.2020.103541>.
- [54] J. Tu, H. Chen, M. Wang, A.H. Gandomi, The colony predation algorithm, *J. Bionic Eng.* 18 (2021) 674–710, <https://doi.org/10.1007/s42235-021-0050-y>.
- [55] H. Jia, X. Peng, C. Lang, Remora optimization algorithm, *Expert Syst. Appl.* 185 (2021) 115665, <https://doi.org/10.1016/j.eswa.2021.115665>.
- [56] Y. Yang, H. Chen, A.A. Heidari, A.H. Gandomi, Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts, *Expert Syst. Appl.* 177 (2021) 114864, <https://doi.org/10.1016/j.eswa.2021.114864>.
- [57] S. Zhao, T. Zhang, S. Ma, M. Chen, Dandelion Optimizer: A nature-inspired metaheuristic algorithm for engineering applications, *Eng. Appl. Artif. Intell.* 114 (2022) 105075, <https://doi.org/10.1016/j.engappai.2022.105075>.
- [58] F.A. Hashim, A.G. Hussien, Snake Optimizer: A novel meta-heuristic optimization algorithm, *Knowledge-Based Syst.* 242 (2022) 108320, <https://doi.org/10.1016/j.knosys.2022.108320>.
- [59] H. Jia, H. Rao, C. Wen, S. Mirjalili, Crayfish optimization algorithm, *Artif. Intell. Rev.* (2023), <https://doi.org/10.1007/s10462-023-10567-4>.
- [60] J. Wang, C. Zhan, S. Li, Q. Zhao, J. Liu, Z. Xie, Adaptive variational mode decomposition based on Archimedean optimization algorithm and its application to bearing fault diagnosis, *Measurement* 191 (2022) 110798, <https://doi.org/10.1016/j.measurement.2022.110798>.
- [61] Y. Zhang, Y. Guo, Y. Xiao, W. Tang, H. Zhang, J. Li, A novel hybrid improved hunger games search optimizer with extreme learning machine for predicting shrinkage of SLS parts, *J. Intell. Fuzzy Syst.* 43 (2022) 5643–5659, <https://doi.org/10.3233/JIFS-212799>.
- [62] M. Abdel-Basset, V. Chang, R. Mohamed, HSMA_WOA: A hybrid novel Slime mould algorithm with whale optimization algorithm for tackling the image segmentation problem of chest X-ray images, *Appl. Soft. Comput.* 95 (2020) 106642, <https://doi.org/10.1016/j.asoc.2020.106642>.
- [63] Q. Liang, S.C. Chu, Q. Yang, A. Liang, J.S. Pan, Multi-group gorilla troops optimizer with multi-strategies for 3D node localization of wireless sensor networks, *Sensors (basel)* 22 (2022) 4275, <https://doi.org/10.3390/s22114275>.
- [64] A.M. Nassef, E.H. Houssein, H.B. E-d, H. Rezk, Modified honey badger algorithm based global MPPT for triple-junction solar photovoltaic system under partial

- shading condition and global optimization, Energy 254 (2022) 124363, <https://doi.org/10.1016/j.energy.2022.124363>.
- [65] T.S.L.V. Ayyarao, P.P. Kumar, Parameter estimation of solar PV models with a new proposed war strategy optimization algorithm, International Journal of Energy Research 46 (2022) 7215–7238, <https://doi.org/10.1002/er.7629>.
- [66] K. Jagathesan, D. Boopathi, S. Samanta, B. Anand, N. Dey, Grey wolf optimization algorithm-based PID controller for frequency stabilization of interconnected power generating system, Soft Comput (2023), <https://doi.org/10.1007/s00500-023-09213-6>.
- [67] Y. Xiao, Y. Guo, H. Cui, Y. Wang, J. Li, Y. Zhang, IHAAVOA: An improved hybrid aquila optimizer and African vultures optimization algorithm for global optimization problems, Math. Biosci. Eng. 19 (2022) 10963–11017, <https://doi.org/10.3934/mbe.2022512>.
- [68] R. Zheng, H. Jia, L. Abualigah, Q. Liu, S. Wang, Deep ensemble of slime mold algorithm and arithmetic optimization algorithm for global optimization, Processes 9 (2021) 1774, <https://doi.org/10.3390/pr9101774>.
- [69] H. Jia, Y. Li, D. Wu, H. Rao, C. Wen, L. Abualigah, Multi-strategy remora optimization algorithm for solving multi-extremum problems, J. Comput. Des. Eng. 10 (2023) 1315–1349, <https://doi.org/10.1093/jcde/qwad044>.
- [70] R. Zheng, A.G. Hussien, R. Qaddoura, H. Jia, L. Abualigah, S. Wang, A. Saber, A multi-strategy enhanced African vultures optimization algorithm for global optimization problems, J. Comput. Des. Eng. 10 (2023) 329–356, <https://doi.org/10.1093/jcde/qwac135>.
- [71] G. Hu, Y. Zheng, L. Abualigah, A.G. Hussien, DETDO: An adaptive hybrid dandelion optimizer for engineering optimization, Adv. Eng. Inform. 57 (2023) 102004, <https://doi.org/10.1016/j.aei.2023.102004>.
- [72] L. Yao, P. Yuan, C.-Y. Tsai, T. Zhang, Y. Lu, S. Ding, ESO: An enhanced snake optimizer for real-world engineering problems, Expert Syst. Appl. 230 (2023) 120594, <https://doi.org/10.1016/j.eswa.2023.120594>.
- [73] J.-S. Pan, J.-X. Lv, L.-J. Yan, S.-W. Weng, S.-C. Chu, J.-K. Xue, Golden eagle optimizer with double learning strategies for 3D path planning of UAV in power inspection, Math. Comput. Simul. 193 (2022) 509–532, <https://doi.org/10.1016/j.matcom.2021.10.032>.
- [74] F.A. Hashim, R.R. Mostafa, R.A. Khurma, R. Qaddoura, P.A. Castillo, A new approach for solving global optimization and engineering problems based on modified sea horse optimizer, J. Comput. Des. Eng. 11 (2024) 73–98, <https://doi.org/10.1093/jcde/qwae001>.
- [75] M. Zhang, H. Chen, A.A. Heidari, Z. Cai, N.O. Aljehane, R.F. Mansour, OCRUN: An oppositional Runge Kutta optimizer with cuckoo search for global optimization and feature selection, Appl. Soft. Comput. 146 (2023) 110664, <https://doi.org/10.1016/j.asoc.2023.110664>.
- [76] S. Wang, H. Jia, L. Abualigah, Q. Liu, R. Zheng, An improved hybrid aquila optimizer and harris hawks algorithm for solving industrial engineering optimization problems, Processes 9 (2021) 1551, <https://doi.org/10.3390/pr901551>.
- [77] X. Yuan, G. Hu, J. Zhong, G. Wei, HBWO-JS: jellyfish search boosted hybrid beluga whale optimization algorithm for engineering applications, J. Comput. Des. Eng. 10 (2023) 1615–1656, <https://doi.org/10.1093/jcde/qwad060>.
- [78] E.S. Ghith, F.A.A. Tolba, Tuning PID Controllers Based on Hybrid Arithmetic Optimization Algorithm and Artificial Gorilla Troop Optimization for Micro-Robotics Systems, IEEE Access 11 (2023) 27138–27154, <https://doi.org/10.1109/ACCESS.2023.3258187>.
- [79] Y. Qian, J. Tu, G. Luo, C. Sha, A.A. Heidari, H. Chen, Multi-threshold remote sensing image segmentation with improved ant colony optimizer with salp foraging, J. Comput. Des. Eng. 10 (2023) 2200–2221, <https://doi.org/10.1093/jcde/qwad093>.
- [80] L. Deng, S. Liu, Snow ablation optimizer: A novel metaheuristic technique for numerical optimization and engineering design, Expert Syst. Appl. 225 (2023) 120069, <https://doi.org/10.1016/j.eswa.2023.120069>.
- [81] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1997) 67–82, <https://doi.org/10.1109/4235.585893>.
- [82] Y. Xiao, X. Sun, Y. Zhang, Y. Guo, Y. Wang, J. Li, An improved slime mould algorithm based on Tent chaotic mapping and nonlinear inertia weight, Int. J. Innov. Comp. Inf. Control 17 (2021) 2151–2176, <https://doi.org/10.24507/ijicic.17.06.2151>.
- [83] L.-K. Hua, Y. Wang, Number theory in the application of approximate analysis, Science Press, Beijing, 1978.
- [84] G. He, X.-l. Lu, Good point set and double attractors based-QPSO and application in portfolio with transaction fee and financing cost, Expert Syst. Appl. 209 (2022) 118339, <https://doi.org/10.1016/j.eswa.2022.118339>.
- [85] A.A. Heidari, P. Pahlavani, An efficient modified grey wolf optimizer with Lévy flight for optimization tasks, Appl. Soft. Comput. 60 (2017) 115–134, <https://doi.org/10.1016/j.asoc.2017.06.044>.
- [86] L. Cheng, J.-X. Zhou, X. Hu, A.W. Mohamed, Y. Liu, Adaptive differential evolution with fitness-based crossover rate for global numerical optimization, Complex Intell. Syst. (2023), <https://doi.org/10.1007/s40747-023-01159-4>.
- [87] G. Hu, L. Chen, X. Wang, G. Wei, Differential evolution-boosted sine cosine golden eagle optimizer with lévy flight, J. Bionic Eng. 19 (2022) 1850–1885, <https://doi.org/10.1007/s42235-022-00223-y>.
- [88] E.H. Houssein, M.A. Mahdy, M.J. Blondin, D. Shebl, W.M. Mohamed, Hybrid slime mould algorithm with adaptive guided differential evolution algorithm for combinatorial and global optimization problems, Expert Syst. Appl. 174 (2021) 114689, <https://doi.org/10.1016/j.eswa.2021.114689>.
- [89] H.R. Tizhoosh, Opposition-based learning: a new scheme for machine intelligence, International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce 1 (2005) 695–701, <https://doi.org/10.1109/CIMCA.2005.1631345>.
- [90] T.-T. Nguyen, H.-J. Wang, T.-K. Dao, J.-S. Pan, J.-H. Liu, S. Weng, An improved slime mould algorithm and its application for optimal operation of cascade hydropower stations, IEEE Access 8 (2020) 226754–226772, <https://doi.org/10.1109/access.2020.3045975>.
- [91] Y. Zhao, C. Huang, M. Zhang, C. Lv, COLMA: a chaos-based mayfly algorithm with opposition-based learning and Levy flight for numerical optimization and engineering design, J. Supercomput. 79 (2023) 19699–19745, <https://doi.org/10.1007/s11227-023-05400-2>.
- [92] Q. Fan, Z. Chen, W. Zhang, X. Fang, ESSAWOA: Enhanced whale optimization algorithm integrated with salp swarm algorithm for global optimization, Eng. Comput. 38 (2020) 797–814, <https://doi.org/10.1007/s00366-020-01189-3>.
- [93] G. Wu, R. Mallipeddi, P.N. Suganthan, Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization. National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report. (2017).
- [94] A. Kumar, K.V. Price, A.W. Mohamed, A.A. Hadi, P.N. Suganthan, Problem definitions and evaluation criteria for the CEC 2022 special session and competition on single objective bound constrained numerical optimization, Nanyang Technological University, Singapore, 2022. Technical Report.
- [95] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power, Inf. Sci. 180 (2010) 2044–2064, <https://doi.org/10.1016/j.ins.2009.12.010>.
- [96] E. Theodorsson-Norheim, Friedman and Quade tests: Basic computer program to perform nonparametric two-way analysis of variance and multiple comparisons on ranks of several related samples, Comput. Biol. Med. 17 (1987) 85–99, [https://doi.org/10.1016/0010-4825\(87\)90003-5](https://doi.org/10.1016/0010-4825(87)90003-5).
- [97] N. Hansen, S.D. Müller, P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), Evol. Comput. 11 (2003) 1–18, <https://doi.org/10.1162/106365603321828970>.
- [98] A.W. Mohamed, A.A. Hadi, A.M. Fattouh, K.M. Jambi, LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems, in: 2017 IEEE Congress on Evolutionary Computation (CEC), (2017) 145–152. 10.1109/CEC.2017.7969307.
- [99] N.H. Awad, M.Z. Ali, P.N. Suganthan, Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems. In: 2017 IEEE Congress on Evolutionary Computation (CEC) (2017) 372–379. 10.1109/CEC.2017.7969336.
- [100] K. Hussain, M.N.M. Salleh, S. Cheng, Y. Shi, On the exploration and exploitation in popular swarm-based metaheuristic algorithms, Neural Comput. Appl. 31 (2019) 7665–7683, <https://doi.org/10.1007/s00521-018-3592-0>.
- [101] C. Li, J. Li, H. Chen, A.A. Heidari, Memetic Harris Hawks Optimization: Developments and perspectives on project scheduling and QoS-aware web service composition, Expert Syst. Appl. 171 (2021), <https://doi.org/10.1016/j.eswa.2020.114529>.
- [102] H. Chen, Z. Wang, H. Jia, X. Zhou, L. Abualigah, Hybrid slime mould and arithmetic optimization algorithm with random center learning and restart mutation, Biomimetics 8 (2023) 396, <https://doi.org/10.3390/biomimetics8050396>.
- [103] B.S. Yildiz, N. Pholdee, S. Bureerat, A.R. Yildiz, S.M. Sait, Robust design of a robot gripper mechanism using new hybrid grasshopper optimization algorithm, Expert Systems 38 (2021) e12666.
- [104] T. Easwarakanthan, J. Bottin, I. Bouhouch, C. Boutrit, Nonlinear minimization algorithm for determining the solar cell parameters with microcomputers, International Journal of Solar Energy 4 (1986) 1–12, <https://doi.org/10.1080/01425918608909835>.