IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 13, 2021, accepted November 9, 2021, date of publication November 18, 2021, date of current version December 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3129255

Enhanced Sparrow Search Algorithm With Mutation Strategy for Global Optimization

BING MA[1,2], PENGMIN LU[1], LUFAN ZHANG[3], YONGGANG LIU[2], QIANG ZHOU[2], YIXIN CHEN[1,2], QISONG QI[2,4], AND YONGTAO HU[5]

[1]Key Laboratory of Road Construction Technology & Equipment, Ministry of Education, Chang'an University, Xi'an 710064, China
[2]Henan Weihua Heavy Machinery Company Ltd., Changyuan 453400, China
[3]School of Mechanical and Electrical Engineering, Henan University of Technology, Zhengzhou 450001, China
[4]Mechanical Engineering Department, Taiyuan University of Science and Technology, Taiyuan, Shanxi 030024, China
[5]School of Mechanical Engineering, Henan Institute of Technology, Xinxiang, Henan 453003, China

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

This work was supported in part by the National Natural Science Foundation of China under Grant 51705132, in part by the Training Program for Young Backbone Teachers of Henan University of Technology, in part by the Science and the Education Department of Henan Province Natural Science Project 21A460006, in part by the Innovative Funds Plan of Henan University of Technology, in part by the Science and Technology Research Project of Henan Province under Grant 202102210061, in part by the Key Scientific Research Projects of Universities in Henan under Grant 19B460001, and in part by the Horizontal Project of Chang'an University under Grant 220225200517.

ABSTRACT In order to improve the performance of the sparrow search algorithm (SSA), in this paper, a novel series of SSA variants is proposed by combining SSA with improved Tent chaos mutation (IT), Lévy flights mutation (LF), elite opposition-based learning mutation (EOBL), variable radius mutation (VR) and the combination of IT, LF, EOBL, and VR, namely, ITSSA, LFSSA, EOBLSSA, VRSSA, and CMSSA, respectively. Initially, the performance of these variants is evaluated on a comprehensive set of 31 benchmark test functions. Moreover, the performance of the best algorithm among these variants is compared with 19 state-of-the-art optimization algorithms to validate its performance on 31 benchmark test functions. The convergence and computational complexity of the best variant are also analyzed to test exploration, exploitation, and local optima avoidance. It is then employed on eight real-world constrained engineering problems to further verify its robustness. The experimental results reveal that the best algorithm of SSA variants outperforms other competitors and is highly effective in solving real-life cases.

INDEX TERMS Sparrow search algorithm, improved Tent chaos mutation, Lévy flights mutation, elite opposition-based learning mutation, variable radius mutation.

NOMENCLATURE

GA	Genetic Algorithm
GP	Genetic programming
DE	Differential evolution
FES	Fast evolution strategies
FEP	Evolutionary programming made faster
BBO	Biogeography-based optimization
SA	Simulated annealing
BB–BC	Big Bang–Big Crunch
CFO	Central Force Optimization
GSA	Gravitational search algorithm
GIO	Gravitational Interactions Optimization
WCA	Water cycle algorithm
BH	Black hole
SRA	Specular Reflection Optimization Algorithm
SCA	Sine Cosine Algorithm
MVO	Multi-Verse Optimizer
IPO	Inclined planes system optimization algorithm
IPSO	Improved PSO with Time-Varying Accelerator Coefficients
GOA	Grasshopper Optimisation Algorithm
SHO	Spotted hyena optimizer
SSA [20]	Salp swarm algorithm
EPO	Emperor penguin optimizer
BOA	Butterfly optimization algorithm
SOA	Seagull optimization algorithm
STOA	Sooty Tern Optimization Algorithm
BOSA	Binary Orientation Search Algorithm
HHO	Harris hawks optimization

The associate editor coordinating the review of this manuscript and approving it for publication was Chun-Hao Chen.

SO	Sailfish optimizer
SMA	Slime mould algorithm
TSA	Tunicate Swarm Algorithm
SSA [30]	Sparrow search algorithm
RSO	Rat Swarm Optimizer
BEPO	Binary emperor penguin optimizer
ESA	Emperor Penguin and Salp Swarm Algorithm
JS	Jellyfish Search
TEO	Thermal exchange optimization
AEFA	Artificial electric field algorithm
SSA [47]	Spring Search Algorithm
TLBO	Teaching–Learning-Based Optimization
BSA	Brain storm optimization algorithm
SBA	Social-based algorithm
LCA	League championship algorithm
SGO	Social group optimization
SMO	Social mimic optimization
PRO	Poor and rich optimization
PO	Political optimizer
DGO	Darts Game Optimizer
HBO	Heap-based optimizer
PSO	Particle swarm optimization
ABC	Artificial bee colony algorithm
FA	Firefly algorithm
BA	Bat algorithm
CS	Cuckoo search algorithm
GWO	Grey Wolf Optimizer
SOS	Symbiotic Organisms Search
ALO	Ant lion optimizer
MFO	Moth-flame optimization algorithm
DA	Dragonfly algorithm
WOA	Whale Optimization Algorithm
LFSSA	Lévy flights-SSA
EOBLSSA	Elite opposition-based learning-SSA
VRSSA	Variable radius perturbation-SSA
CMSSA	Combined mutated-SSA
N	The population of sparrows
D	The dimension
k	The current iteration
$max\_iteration$	The maximum number of iterations
α	A uniform random number within (0,1]
Q	A random number obeying normal distribution within [0,1]
L	A matrix of $1 \times$ D for which each element is 1
R_2	A random alarm factor within [0,1]
ST	A safety factor within [0.5,1]
PD	The number of discovers
SD	The number of sparrows who to be on guard
ED	The number of sparrows who updated by the EOBL
LD	The number of sparrows who updated by the LF.
x_{iD}^{k}	The k-th location of sparrows
x_{wrost}^{k}	The k-th worst solution in the whole search space.
x_{B}^{k+1}	The $(k+1)$-th best location of sparrows
A	A matrix of $1\times D$ for which each element is 1 or -1
β	The step size factor randomly within [0,1].
γ	The step size factor randomly within $[-1,1]$
x_{best}^{k}	The k-th best location of sparrows
ε	A minimum constant to avoid zero denominator
f_{best}	The current best solution
f_{worst}	The worst solution
f_i	The fitness value of i-th sparrows
x_{iD}^{best}	The best location of sparrows
$Tent\left(x_{iD}^{best}\right)$	The best location of sparrows mutated IT operator
r	Adaptive attenuation factor
ρ	The inertia weighting factor
ACO	Ant colony optimization
IIR	Infinite impulse response
MMIPO	Multimodal IPO algorithm
MOIPO	Multi-objective inclined planes system optimization algorithm
MIPO	Modified inclined planes system optimization algorithm
MOMIPO	Multi-Objective Modified Inclined Planes System Optimization
SIPO	Simplified and Efficient Version of IPO
VLIPO	Variable Length Inclined Planes System Optimization Algorithm
CSSA	Chaos sparrow search optimization algorithm
ISSA	Improved sparrow algorithm
ASSA	Adaptive sparrow search algorithm
CASSA	Modified sparrow search algorithm
ESSA	Enhanced sparrow search algorithm
OBL	Opposition-based learning mechanism
SCA-CSSA	Improved sparrow search algorithm using SCA algorithm and labor cooperation structure
LLSSA	Lens learning sparrow search algorithm
KLSSA	Multi-strategy improved sparrow search algorithm
SSA-PSO	Hybrid sparrow search optimization algorithm and particle swarm optimization algorithm
SVM	Support Vector Machine
IT	Improved Tent chaos mutation
LF	Lévy flights mutation
EOBL	Elite opposition-based learning mutation
VR	Variable radius mutation
ITSSA	Improve Tent chaos-SSA
$N_{\rho}(t)$	The elite individuals
$x_{i,j}(t)$	The solution by the common individuals
$x_{i,j}^{*}(t)$	The elite opposition-based learning solution

k''	A random number within [0,1]
$a_j(t)$	The lower limiting value of $x_{i,j}(t)$
$b_j(t)$	The upper limiting value of $x_{i,j}(t)$
R	The variable radius factor
ub	The upper bounds of variable
lb	The lower bounds of variable
Avg.	Average value
Med.	Median value
Std.	Standard deviation value
ARV.	Average ranking value
T_A	The number of teeth on gear A
T_B	The number of teeth on gear B
T_D	The number of teeth on gear D
T_F	The number of teeth on gear F
MIBBSQP	Nonlinear Branch and Bound Algorithm
IDCNLP	Mixed Integer-discrete-continuous Programming
MVEP	Mixed-variable evolutionary programming
Gene AS	Genetic Adaptive Search
HSIA	Hybrid swarm intelligence approach
UPSO	Unified Particle Swarm Optimization
CAPSO	Chaos-enhanced accelerated particle swarm optimization
A_1	The cross-sectional area of first bar
A_2	The cross-sectional area of second bar
DEDS	Differential evolution with dynamic stochastic selection
MBA	Mine blast algorithm
PSO-DE	Hybridizing particle swarm optimization with differential evolution
TAS	Taguchi-Aided Search Method
L_i	Side length of the i-th square-shaped cross-section
MMA	Method of Moving Asymptotes
$L(\beta)$	A randomly distributed number drawn from Lévy distribution
x_j	A point in D-dimension space
a_j	The lower limiting value of x_j
b_j	The upper limiting value of x_j
x_j^*	The opposite point of x_j
k'	A random number within [0,1]
$x_i(t)$	A solution of the t-th iteration
$x_i^*(t)$	The opposite solution of $x_i(t)$
$N_i(t)$	The elite individuals
$Q_i(t)$	The common individuals
MCOA	Modified Cuckoo Optimization Algorithm
B_1	Face width
Z_1	Module of teeth
Z_2	Number of teeth in the pinion
B_2	Length of the first shaft between bearings
B_3	Length of the second shaft between bearings
D_1	Diameter of first shafts
D_2	Diameter of second shaft
MPSO	Particle swarm optimization algorithm with asymmetric time varying acceleration coefficients
b_1	Height of the main girder
b_2	Width of the main girder
LP	Linear Programming
GCA(I)	Generalized convex approximation using the simplified form
GCA(II)	Generalized convex approximation using the full form
CPSO	Co-evolutionary particle swarm optimization
ES	Evolution strategies
RO	Ray Optimization
d'	Wire diameter
D'	Mean coil diameter
N'	Number of active coils
T_s	Thickness of spherical shell
T_h	Thickness of ball head
R'	Radius of spherical shell
L'	Length of spherical shell
SBS	Socio-Behavioural Simulation Model
h	Thickness of weld
l	Length of the clamped bar
t	Height of the bar
b	Thickness of the bar
TAPSO	Modified particle swarm optimization with an adaptive acceleration coefficients
GWOCS	Grey wolf optimizer algorithm integrated with Cuckoo Search
b_3	Thickness of the web plate
b_4	Thickness of the flange plate

I. INTRODUCTION

Optimization problem has been a hot research topic, which is widely used in science, engineering, economy, management, and other fields. Over the past few years, deterministic optimization techniques have been widely used to solve these optimization problems. However, these techniques have some limitations to obtain a better solution, especially for multimodal, nonlinear constrained, and complex real-world optimization problems. Therefore, to better deal with these cases, meta-heuristic algorithms have been developed rapidly in recent years. The advantages of these methods are the flexibility, gradient-free mechanism, and avoiding local optimum. In addition, since they belong to a class of random techniques with different random operators to help them can effectively avoid local optimum for solving real-world problems. Meta-heuristic algorithms are usually divided into evolutionary-based algorithms, swarm-based algorithms, physics-based algorithms, and human-based algorithms.

Evolutionary-based algorithms generally are inspired by the process of biological evolution. The individuals of each generation of the population are randomly generated through the previous generation of individuals through selection, reproduction, mutation, etc. The individuals of the population are constantly updated and iterated to achieve global optimization. However, the disadvantages of these algorithms are discarding the information of the previous

TABLE 1. A few famous evolutionary-based algorithms.

Algorithm	Authors	Year	Inspiration
Genetic Algorithm (GA) [1]	Holland John H.	1992	Darwinian theory of evolution
Genetic programming (GP) [2]	Koza J.R.	1994	Darwinian principle of survival of the fittest and genetic crossover
Differential evolution (DE) [3]	R. Storn, K. Price	1997	Like genetic algorithms using similar operators; crossover, mutation and selection
Fast evolution strategies (FES) [4]	Yao X., Liu Y.	1997	Cauchy distribution and evolution strategies
Evolutionary programming made faster (FEP) [5]	Xin Yao, et al.	1999	Gaussian mutation and Evolutionary programming
Biogeography-based optimization (BBO) [6]	D. Simon	2008	Mathematics of biogeography

generations of populations, and the high computational complexity due to a large number of operators. A few famous evolutionary-based algorithms are presented in Table 1.

Swarm-based algorithms are inspired by the collective intelligence behavior of social creatures (i.e., fireflies, ant lions, grey wolfs, seagulls, etc.). These methods are easy to implement since they have fewer parameters and operators. Due to the characteristics of information sharing, coevolution, and learning among population agents, they can efficiently avoid the local optimum. A few well-known swarm-based algorithms are presented in Table 2.

Physics-based algorithms mimic the physical phenomena (such as the black hole, the force of gravity, electricity, etc.) in nature. These algorithms have a high ability to avoid local optima since the information is exchanged between the candidate solutions. Some of the popular algorithms in this category are shown in Table 3.

Human-based algorithms are inspired by different human behaviors (such as brainstorming process, competition behavior, teaching behavior, learning behavior, etc.). They usually have the characteristics of organization, persistence, simplicity, and intelligence. The search strategy of these algorithms is different from that of other types. And the search agents of these algorithms are updated and iterated according to different human behaviors to avoid local optimum. Some of the state-of-art algorithms in this category are shown in Table 4.

Therefore, in recent years, various meta-heuristic algorithms inspired by different concepts are coming out in a rush. However, the metaheuristic frameworks of these algorithms are similar to some extent [58]. The similarity of these meta-heuristic algorithms is that they can achieve better solutions and have two search stages: exploration and exploitation [59]. The exploration stage is to seek the global optimal solution in the search space as much as possible. The exploitation stage is to further search for the global optimal solution to improve the search accuracy based on obtaining the optimal solution in the exploration stage. Only using exploration may decrease the convergence accuracy, but only using exploitation may increase the probability of getting trapped in a local optimum. Therefore, it is still a challenging problem that how to seek the balance between exploration and exploitation. In addition, for many problems, these meta-heuristic algorithms can obtain a better solution. And meta-heuristic algorithms have some limits, likely the parameters of algorithms being set manually [60]. Based on the No Free Lunch theory [61], these popular meta-heuristic algorithms are not fully guaranteed to seek the global optimum for all optimization problems, especially NP-hard optimization problems.

At present, to better handle these NP-hard optimization problems, the performance of meta-heuristic algorithms is improved in different ways. A hybrid method is an efficient way to improve the algorithms [62], such as PSO with non-smooth penalty reformulation [63], a randomized fixing strategy inspired by ACO, and an exact large neighborhood search [64], Integrating LP [65], etc. A single algorithm improved with other approaches is another effective way. For instance, combing with Newton's second law and equations of motion, the inclined planes system optimization algorithm (IPO) inspired by the dynamic of tiny ball's sliding motion along frictionless inclined planes was proposed, which effectively handled some single-objective optimization problems [66]. It has the advantages of high stability, robustness, and high convergence efficiency. In recent years, the IPO algorithm was used to solve some various optimization problems, such as optimal design of the level shifter circuit [67], the data clustering problem [68], the unsupervised data and histogram clustering problem [69], the automatic design of a neuro-fuzzy classifier [70], the optimal architecture of MLP neural networks [71], the optimal design of IIR digital filters [72], the IIR system identification problem [73], the IIR system modeling [74], the epileptic seizure detection [75], etc. The multimodal IPO (MMIPO) algorithm was efficiently used to solve the multimodal optimization problems [76]. The fourth-order butterworth filter was effectively and efficiently designed based on the multi-objective inclined planes system optimization (MOIPO) algorithm [77]. The problem of IIR model identification was optimized by a modified inclined planes system optimization (MIPO) algorithm using the appropriate mechanism based on the executive steps of the algorithm with the constant damp factors [78]. Compared to other multi-objective methods, the multi-objective modified inclined planes system optimization (MOMIPO) algorithm was better to solve the ring oscillator optimal design [79]. The multi-objective inclined planes system optimization (MOIPO) algorithm was used to handle the optimization of CMOS cross-coupled LC voltage-controlled oscillators [80]. A simplified and efficient version of IPO (SIPO) with high reliability and stability was proposed and superior to IPO and MIPO [81]. An adaptive neuro-fuzzy inference system classifier was effectively designed by the variable-length inclined planes system optimization algorithm (VLIPO) method [82].

TABLE 2. A few well-known swarm-based algorithms.

Algorithm	Authors	Year	Inspiration	Algorithm	Authors	Year	Inspiration
Particle swarm optimization (PSO) [7]	J. Kenned and R. Eberhart	1995	Particle swarm methodology	Emperor penguin optimizer (EPO) [21]	Gaurav Dhiman and Vijay Kumar	2018	Huddling behavior of emperor penguins
Artificial bee colony algorithm (ABC) [8]	D. Karaboga, B. Basturk	2007	Intelligent behaviour of honey bee swarm	Butterfly optimization algorithm (BOA) [22]	Arora S., Singh S.	2019	Food search and mating behavior of butterflies
Firefly algorithm (FA) [9]	Yang X.S.	2009	Foraging behaviours of fireflies	Seagull optimization algorithm (SOA) [23]	Gaurav Dhiman, Vijay Kumar	2019	Migration and attacking behaviors of a seagull
Bat algorithm (BA) [10]	Yang X., Gandomi A. H.,	2012	Echolocation behavior of bats	Sooty Tern Optimization Algorithm (STOA)[24]	Gaurav Dhiman and Amandeep Kaur	2019	Migration and attacking behaviors of sea bird sooty tern
Cuckoo search algorithm (CS) [11]	Gandomi A. H., Yang X., et al.	2013	Cuckoo breeding behavior with Lévy flights	Binary Orientation Search Algorithm (BOSA) [25]	Mohammad Dehghani et al.	2019	Rules of orientation game and binary model
Grey Wolf Optimizer (GWO) [12]	Mirjalili S., S.M. Mirjalili and A. Lewis	2014	Leadership hierarchy and hunting mechanism of grey wolves	Harris hawks optimization (HHO) [26]	Ali Asghar Heidari et al.	2019	Cooperative behavior and chasing style of Harris' hawks
Symbiotic Organisms Search (SOS) [13]	Cheng M-Y, Prayogo D	2014	Symbiotic interaction strategies adopted by organisms to survive and propagate in the ecosystem	Sailfish optimizer (SO) [27]	Shadravan S. et al.	2019	A group of hunting sailfish
Ant lion optimizer (ALO) [14]	Seyedali Mirjalili	2015	Hunting mechanism of antlions	Slime mould algorithm (SMA) [28]	Shimin Li et al.	2020	Oscillation mode of slime mould
Moth-flame optimization algorithm (MFO) [15]	Mirjalili S.	2015	Navigation method of moths	Tunicate Swarm Algorithm (TSA) [29]	Satnam Kaur et al.	2020	Jet propulsion and swarm behaviors of tunicates during the navigation and foraging process.
Dragonfly algorithm (DA) [16]	Mirjalili S.	2016	Static and dynamic swarming behaviours of dragonflies	Sparrow search algorithm (SSA) [30]	Xue J., Shen B.	2020	Group wisdom, foraging and anti-predation behaviours of sparrows
Whale Optimization Algorithm (WOA) [17]	Mirjalili S. and A. Lewis	2016	Social behavior of humpback whales	Rat Swarm Optimizer (RSO) [31]	Dhiman G. et al.	2021	Chasing and attacking behaviors of rats
Grasshopper Optimisation Algorithm (GOA) [18]	Shahrzad, Saremi, Seyedali, et al.	2017	Behaviour of grasshopper swarms	Binary emperor penguin optimizer (BEPO) [32]	Dhiman Gaurav et al.	2021	Emperor penguin's huddling behaviour
Spotted hyena optimizer (SHO) [19]	Gaurav Dhiman and Vijay Kumar	2017	Behavior of spotted hyenas	Emperor Penguin and Salp Swarm Algorithm (ESA) [33]	Dhiman Gaurav	2021	Huddling and swarm behaviors of emperor penguin optimizer and salp swarm algorithm
Salp swarm algorithm (SSA) [20]	Mirjalili S. et al.	2017	Swarming behaviour of salps when navigating and foraging in oceans	Jellyfish Search (JS) [34]	Jui-ShengChou, Dinh-Nhat Truong,	2021	Behavior of jellyfish in the ocean

To better seek the balance between exploration and exploitation and the global optimum, the sparrow search algorithm (SSA) was proposed by Jiankai Xue, which was mainly inspired by the sparrow group wisdom, foraging

TABLE 3. A few popular physics-based algorithms.

Algorithm	Authors	Year	Inspiration
Simulated annealing (SA) [35]	Kirkpatrick S. et al.	1983	Principle of solid annealing
Big Bang–Big Crunch (BB–BC) [36]	Erol O. K., & Eksin I.	2006	Big Bang and Big Crunch Theory
Central Force Optimization (CFO) [37]	Formato R.A.	2008	Metaphor of gravitational kinematics
Gravitational search algorithm (GSA) [38]	Rashedi E., et al.	2009	Law of gravity and mass interactions
Gravitational Interactions Optimization (GIO) [39]	Flores J.J., et al.	2011	Gravitational interactions
Water cycle algorithm (WCA) [40]	Eskandar H, et al.	2012	Observation of water cycle process and how rivers and streams flow to the sea in the real world
Black hole (BH) [41]	Abdolreza Hatamlou	2013	Black hole phenomenon
Specular Reflection Optimization Algorithm (SRA) [42]	Qi Qisong, et al.	2015	Unique optical property of mirror-reflection function
Sine Cosine Algorithm (SCA) [43]	Mirjalili S.	2016	Sine and cosine functions
Multi-Verse Optimizer (MVO)[44]	Mirjalili S.	2016	Three concepts in cosmology: white hole, black hole, and wormhole
Inclined planes system optimization algorithm (IPO) [66]	Mozaffari M.H., et al.	2016	Dynamic of tiny ball's sliding motion along frictionless inclined planes
Thermal exchange optimization (TEO)[45]	Kaveh A., and A. Dadras.	2017	Newton's law of cooling
Artificial electric field algorithm (AEFA) [46]	Yadav A. et al.	2019	Electrostatic Force on charges
Spring Search Algorithm (SSA) [47]	D. Mohammad et al.	2020	Hooke's law

TABLE 4. A few popular human-based algorithms.

Algorithm	Authors	Year	Inspiration
Teaching–Learning-Based Optimization (TLBO) [48]	Rao R., et al.	2011	Effect of influence of a teacher on learners
Brain storm optimization algorithm (BSA) [49]	Xue J., et al.	2012	Brainstorming process
Social-based algorithm (SBA) [50]	Ramezani F., & Lotfi S.	2013	Evolutionary Algorithm (EA) and socio-political process
League championship algorithm (LCA) [51]	Kashan, A. H.	2014	Championship environment wherein artificial teams play
Social group optimization (SGO) [52]	Satapathy S. & Naik A.	2016	Concept of social behavior of human
Social mimic optimization (SMO) [53]	Balochian S. & Baloochian H.	2012	Mimicking behavior
Poor and rich optimization (PRO) [54]	Moosavi S. H. S. & Bardsiri V. K.	2019	Efforts of the two groups of the poor and the rich to achieve wealth and improve their economic situation
Political optimizer (PO) [55]	Askari, Q., et al.	2020	Multi-phased process of politics
Darts Game Optimizer (DGO) [56]	Dehghani, Mohammad, et al.	2020	Darts Game
Heap-based optimizer (HBO) [57]	Qamar Askari, et al.	2020	Corporate rank hierarchy

and anti-predation behaviors of sparrows [30]. Compared with PSO, GWO, and GSA, SSA is a superior meta-heuristic algorithm with the advantages of fast convergence rate, higher stability and strong resistance. However, the basic SSA can easily fall into the local optimum for high multimodal and complex problems. To improve the performance of basic SSA, the chaos sparrow search optimization algorithm (CSSA) was proposed by combining the Tent chaotic sequence and Gaussian mutation [83], and the improved sparrow algorithm (ISSA) by combining Cauchy mutation and opposition-based learning [84] was proposed. The adaptive sparrow search algorithm (ASSA) was proposed by introducing the adaptive learning factor, DE/best/1 mutation strategy, and dynamic scaling factor to deal with the problem of the optimal parameter identification of the PEMFCs [85]. Combining the center of gravity reverse learning, learning coefficient, and Cauchy mutation operators, the improved sparrow search algorithm with good steady-state performance was applied to track the problem of a distributed maximum power point tracking [86]. By introducing the chaotic map, adaptive inertia weight, and Cauchy–Gaussian mutation strategies, the modified sparrow search algorithm (CASSA) was proposed to efficiently solve the UAV route planning problem [87]. The convolutional neural network is optimized by an enhanced sparrow search algorithm (ESSA) classification which is improved by the opposition-based learning (OBL) mechanism and the Merit function mechanism [88]. By using the SCA algorithm and labor cooperation structure, an improved sparrow search algorithm (SCA-CSSA) was proposed to solve the labeled and unlabeled data classification problem [89]. Based on the logistic map, the chaotic sparrow search algorithm was utilized in the stochastic configuration network [90]. A lens learning sparrow search algorithm (LLSSA), which combined the reverse learning strategy, variable spiral search strategy, and the simulated annealing algorithm, was applied to optimize the 3D UAV path planning problem [91]. The sparrow search algorithm (SSA) was introduced to optimize

TABLE 5. Description of unimodal test functions.

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

TABLE 6. Description of multimodal test functions.

Function	Dim	Range	$f_{\min}$
$F_8(x) = \sum_{i=1}^{n} -x_i sin\sqrt{\lvert x_i\rvert}$	n	[-500,500]	-418.9829×n
$F_9(x) = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$	n	[-5.12,5.12]	0
$F_{10}(x) = -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	n	[-32,32]	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	n	[-600,600]	0
$F_{12}(x) = \frac{\pi}{n}\{sin(\pi y_1)\sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10,100,4)$ $y_i = 1 + \frac{x_i+1}{4} \quad u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	n	[-50,50]	0
$F_{13}(x) = \sum_{i=1}^{n} 0.1\{sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5,100,4)$	n	[-50,50]	0

the extreme learning machine model [92], the parameters of the variational mode decomposition method [93], and the penalty factor and kernel function parameter of SVM [94]. By using the K-means clustering, the levy flight mechanism, and the adaptive local search strategy, the multi-strategy improved sparrow search algorithm (KLSSA) is proposed to overcome the shortcomings of SSA [95]. The hybrid SSA-PSO algorithm was proposed to solve the software defect prediction problem [96]. To solve the large error of DV-Hop, ISSADV-Hop algorithm was proposed based on the improved sparrow search algorithm which introduced the levy flight mechanism [97]. An improved sparrow search algorithm was proposed by the adaptive local search strategy, the improved Tent chaotic map and Cauchy mutation [98]. To deal with the economic optimization of the microgrid cluster problem, a chaos sparrow search algorithm was proposed combing the Bernoulli chaotic map, dynamic adaptive weighting, Cauchy mutation, and reverse learning [99].

Although the above methods have improved the search ability and convergence speed of basic SSA, they also have difficulty in avoiding the local optimum to deal with more complex problems. Therefore, it can be seen that the basic SSA is difficult to obtain the global optimal solution, especially for the high-dimensional and complex multimodal problems with a limited number of iterations, and then simple mutation algorithms can still hardly balance the ability of between exploration and exploitation.

In this paper, a novel series of SSA variants (namely, ITSSA, LFSSA, EOBLSSA, VRSSA, and CMSSA, respectively) is proposed through the combining SSA with different mutation operators (i.e., improved Tent chaos mutation (IT), Lévy flights mutation (LF), elite opposition-based learning mutation (EOBL), variable radius mutation (VR)). In addition, the performance of these variants is evaluated on 31 benchmark test functions. Further, the performance of the best SSA variant is comprehensively tested on 31 benchmark test functions and seven constrained problems. The results show that the best SSA variant is very competitive when compared to other existing methods.

The remaining sections are organized as follows: Initially, a brief review of basic SSA is presented in Section II. In addition, to further balance the exploration and exploitation phases effectively, in Section III, a series of mutation-based SSA is proposed, namely ITSSA, LFSSA, EOBLSSA, VRSSA, and CMSSA, respectively. Experimental results and the applicability of mechanical engineering problems are implemented in detail in Section IV. And then the conclusion and future work is summarized in Section V.

II. OVERIEW OF SPARROW SEARCH ALGORITHM (SSA)

SSA is a novel meta-heuristic optimization algorithm proposed by Jiankai Xue in 2020, which is mainly inspired by the foraging behavior and anti-predation behavior of sparrows [30]. The main principle of SSA can be simplified as a discoverer-participant mathematical model in the foraging process, and then the reconnaissance and alarming behavior are introduced into the model. The detailed steps of SSA are as follows:

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

Function	Dim	Range	$f_{\min}$
$F_{14}(x)=\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{i,j})^6}\right)^{-1}$	2	[-65,65]	1
$F_{15}(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}\right]^2$	4	[-5,5]	0.00030
$F_{16}(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	[-5,5]	-1.0316
$F_{17}(x)=(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6)^2+10\left(1-\frac{1}{8\pi}\right)cosx_1+10$	2	[-5,5]	0.398
$F_{18}(x)=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]\times[30+(2x_1-3x_2)^2\times(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$	2	[-2,2]	3
$F_{19}(x)=-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2\right)$	3	[1,3]	-3.86
$F_{20}(x)=-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2\right)$	6	[0,1]	-3.32
$F_{21}(x)=-\sum_{i=1}^{5}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	-10.1532
$F_{22}(x)=-\sum_{i=1}^{7}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	-10.4028
$F_{23}(x)=-\sum_{i=1}^{10}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0,10]	-10.5363

TABLE 8. **Hybrid and composition benchmark functions.**

Function	Function name	characteristic	Dim	Range	f_{min}
F_{24}	Hybrid Function 6 (N=5)	Hybrid	10	[-100,100]	1700
F_{25}	Hybrid Function 6 (N=5)	Hybrid	10	[-100,100]	1800
F_{26}	Hybrid Function 6 (N=5)	Hybrid	10	[-100,100]	1900
F_{27}	Hybrid Function 6 (N=6)	Hybrid	10	[-100,100]	2000
F_{28}	Composition Function 1 (N=3)	Composition	10	[-100,100]	2100
F_{29}	Composition Function 3 (N=4)	Composition	10	[-100,100]	2300
F_{30}	Composition Function 4 (N=4)	Composition	10	[-100,100]	2400
F_{31}	Composition Function 7 (N=6)	Composition	10	[-100,100]	2700

Step 1: Assuming that there are N sparrows in the D-dimensional space, the position X of the i-th sparrow in the D-dimensional space can be expressed by Eq. (1).

$$X=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1D} \\ x_{21} & x_{22} & \cdots & x_{2D} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{ND} \end{bmatrix} \tag{1}$$

Step 2: The fitness value is evaluated by Eq.(2). The sparrow with the best fitness value is selected as the discoverers to lead the whole sparrow population to get closer to the food source. The location of the discoverers is updated by Eq. (3).

$$F=\begin{bmatrix} f(x_{11},x_{12},\cdots,x_{1D}) \\ f(x_{21},x_{22},\cdots,x_{2D}) \\ \vdots \\ f(x_{N1},x_{N2},\cdots,x_{ND}) \end{bmatrix} \tag{2}$$

$$x_{iD}^{k+1} = \begin{cases} x_{iD}^{k}\cdot\exp\left(\dfrac{-i}{\alpha\cdot\max\_iteration}\right) & if\ R_2 < ST\ (3.1) \\ x_{iD}^{k}+Q\cdot L & if\ R_2 \geq ST\ (3.2) \end{cases} \tag{3}$$

where k is the number of current iterations, $max\_iteration$ is the maximum number of iterations, α is a uniform random number within (0,1], Q is a random number obeying normal distribution within [0,1]. L is a matrix of $1\times D$ for which each element is 1. R_2 is a random alarm factor within [0,1] and ST is a safety factor within [0.5,1], respectively.

Step 3: The rest of the sparrows are selected as the participants, except the sparrows selected as the discovers. And then the location of the participants is updated by Eq. (4).

$$x_{iD}^{k+1}=\begin{cases} Q\cdot \mathrm{e}^{\frac{x_{wrost}^{k}-x_{iD}^{k}}{i^2}} & if\ i>\dfrac{N}{2}\ (4.1) \\ x_{B}^{k+1}+\left|x_{iD}^{k}-x_{B}^{k+1}\right|\cdot A^{+}\cdot L & Otherwise\ (4.2) \end{cases} \tag{4}$$

where x_{wrost}^{k} is the worst solution in the whole search space. x_{B}^{k+1} is the current best solution obtained by the discoverers. A is a matrix of $1\times D$ for which each element is 1 or -1, and $A^{+}=A^{T}(AA^{T})^{-1}$.

Step 4: When the sparrows start to forage, 10%-20% of the sparrows are selected to be on guard. When they find the dangers approaching, both the discoverers and participants will give up the current food and fly to another location. Based on the alarming behavior, the location of sparrows is updated by Eq. (5).

$$x_{iD}^{k+1} = \begin{cases} x_{best}^{k} + \beta \cdot \left| x_{iD}^{k} - x_{best}^{k+1} \right| & \text{if } f_i > f_{gbest} \quad (5.1) \\ x_{iD}^{k} + \gamma \cdot \left(\dfrac{\left| x_{iD}^{k} - x_{worst}^{k} \right| \cdot A^{+} \cdot L}{(f_{best} - f_{worst}) + \varepsilon} \right) & \text{if } f_i = f_{gbest} \quad (5.2) \end{cases} \tag{5}$$

where β is the step size factor randomly within [0,1], which obeys normal distribution. γ is also the step size factor randomly within [−1,1], which represents the movement direction of sparrows. ε is a minimum constant to avoid a zero denominator. f_i is the fitness value of i-th sparrows, f_{best} and f_{worst} are the current best solution and worst solution, respectively. The pseudo code of SSA can be expressed in **Algorithm 1**, and the flowchart of the SSA algorithm is shown in Fig. 1, respectively.

Algorithm 1 Pseudo Code of SSA

Set the parameters of SSA, the population of sparrows N, the number of discovers PD, the number of sparrows SD who to be on guard, the safety factor ST, $max\_iteration$, respectively.
Initialize the population X.
While ($k < max\_iteration$)
  Calculate the fitness vaule F by Eq.(2), and the best solution f_{best} and worst solution f_{worst} by sort (F), respectively.
  Update the R_2
  for $i = 1$: PD
    if ($R_2 < ST$)
      The location of sparrows is updated by Eq.(3.1)
    else
      The location of sparrows is updated by Eq.(3.2)
      End if
  End for
  for $i = (PD+1)$: N
    if ($i < N/2$)
    The location of sparrows is updated by Eq.(4.1)
   else
    The location of sparrows is updated by Eq.(4.2)
    End if
  End for
    for $j = 1$: SD
      if ($f_i < f_{gbest}$)
   The location of sparrows is updated by Eq.(5.1)
    else
    The location of sparrows is updated by Eq.(5.2)
   End if
  End for
   The best solution and positon are updated.
   $k = k + 1$;
End while
Return X_{gbest} and f_{gbest}

III. THE PROPOSED METHODOLOGY

For low-dimensional unimodal and multimodal optimization problems, SSA is characterized by a fast convergence speed and better global convergence ability. However, for some complex problems, especially the high dimensional and multimodal problems, SSA is easier to fall into the local optimum. The performance of basic SSA focuses on the interaction between sparrow individuals. In addition, when most sparrows are trapped into the same local optimum, SSA will slow down and eventually stagnate.

In order to effectively overcome the shortcomings of SSA for complex optimization problems, an effective method called mutation strategy is introduced into SSA. In the mutation strategy, there are four innovations embedded into the basic SSA, which are improved Tent chaos map, Lévy flights, elite opposition-based learning, variable radius perturbation, respectively. In a variety of ways, hybrid mutation strategies are introduced to mutate the population of the basic SSA and to enhance the performance of basic SSA. The ability to quickly move the best solution and enrich the high diversity of the population is improved by the hybrid mutation strategies. In this work, several mutation mechanisms are introduced into the basic SSA. The first, second, third, and fourth variants exploit the concept of improved Tent chaos map, Lévy flights, elite opposition-based learning, variable radius distribution, respectively. The fifth variant uses the combination of improved Tent chaos map, Lévy flights, elite opposition-based learning, variable radius perturbation in three different ways. The detailed variants are as follows.

A. IMPROVED TENT CHAOS-SSA (ITSSA)

Like other traditional metaheuristic algorithms, the basic SSA is easier to fall into local optimum by the weak diversity of the population. Therefore, in this section, the population of SSA is initialized by the improved Tent chaos map to enhance the diversity of the population in SSA.

1) INITIALIZED POPULATION BY IMPROVED TENT CHAOS MAP

The Tent chaos map plays a great influence on the performance of the optimization algorithm [100] and has the advantage of uniform ergodicity and faster search speed [101]. However, the Tent chaos map also has disadvantages of small periods and unstable periodic points. Therefore, to avoid falling into a small period or unstable periodic point, the Tent chaos map is improved by the rand(0,1) × 1/N as shown in Eq. (6) [102].

$$x_{i+1} = \begin{cases} 2x_i + \mathrm{rand}(0,1) \times \dfrac{1}{N} & 0 \le x \le 0.5 \\ 2(1 - x_i) + \mathrm{rand}(0,1) \times \dfrac{1}{N} & 0.5 < x \le 1 \end{cases} \tag{6}$$

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

$$x_{i+1} = (2x_i)\, mod\, 1 + \mathrm{rand}(0,1) \times \frac{1}{N} \tag{7}$$

where N is the population number of sparrows.

Based on improved Tent chaos strategy, the detailed steps to initialize the population of SSA are as follows.

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

Step 2: Set the maximum number of iterations is *max_ieration* and according to Eq. (6), loop iteration is calculated for i times, chaotic sequence x_D is obtained.

Step 3: If $i < max\_iteration$, save the x_D.

Therefore, in ITSSA, the Eq.(1) in SSA is replaced by the Eqs. (6) and (7) to increase the population diversity.

2) OPTIMAL INDIVIDUAL PERTURBATION BY IMPROVED TENT CHAOS

When all sparrows find the optimal solution, the optimal sparrow individual is mutated with the improved Tent chaos by the random roulette strategy to improve the global convergence accuracy. Therefore, in ITSSA, the optimal sparrow individual was mutated by Eq. (8) and (9).

$$r = \frac{e^{2\times(1-k/\max\_iteration)} - e^{-2\times(1-k/\max\_iteration)}}{e^{2\times(1-k/\max\_iteration)} + e^{-2\times(1-k/\max\_iteration)}} \tag{8}$$

$$x_{iD}^{best'} = x_{iD}^{best} \times \left(1 + Tent\left(x_{iD}^{best}\right)\right) \; if \; rand < r \tag{9}$$

where $Tent\left(x_{iD}^{best}\right)$ can be calculated by Eq. (6) and (7).

Thus, the pseudo code of ITSSA can be expressed in **Algorithm 2** as follows.

Algorithm 2 Pseudo Code of ITSSA

Set the parameters of ITSSA, the population of sparrows N, the number of discovers PD, the number of sparrows SD who to be on guard, the safety factor ST, $max\_iteration$, respectively.
Initialize the population X by Eq.(6) and (7).
While ($k < max\_iteration$)
  Calculate the fitness vaule F by Eq.(2), and the best solution f_{best} and worst solution f_{worst} by sort (F), respectively.
  Update the R_2
  for $i = 1$: PD
    if ($R_2 < ST$)
      The location of sparrows is updated by Eq.(3.1)
    else
      The location of sparrows is updated by Eq.(3.2)
    End if
  End for
  for $i = (PD+1)$: N
    if($i < N/2$)
      The location of sparrows is updated by Eq.(4.1)
    else
      The location of sparrows is updated by Eq.(4.2)
      End if
  End for
      for $j = 1$: SD
        if($f_i < f_{gbest}$)
      The location of sparrows is updated by Eq.(5.1)
        else
      The location of sparrows is updated by Eq.(5.2)
    End if
  End for
      The best solution and positon are updated.
      Calculate the r by Eq. (8).
      if($rand < r$)
      the optimal sparrow individual is mutated by Eq. (9).
      End if
      The best solution and positon are updated.
      $k = k + 1$;
End while
Return X_{gbest} and f_{gbest}

B. LÉVY FLIGHTS-SSA (LFSSA)

In this section, in order to enhance the expansion of search space, and improve the ability to avoid falling into local optimum. The Lévy flights mutation and the inertia weighting factor are introduced into the basic SSA. In this way, LFSSA can find the global optimum more effectively. The Lévy distribution can be expressed by Eq. (10) [103].

$$Levy\,(\beta) \sim u = t^{-1-\beta}, 0 < \beta \leq 2 \tag{10}$$

where β is the index of stability. The Lévy distribution can also be described by Eq. (11).

$$Levy\,(\beta) \sim \frac{\gamma \times \mu}{|v|^{1/\beta}} \tag{11}$$

where u and v are both standard normal distribution, $\beta = 1.5$, and γ is expressed by Eq. (12).

$$\gamma = \left[\frac{\Gamma(1+\beta)\sin(\pi \times \beta/2)}{\Gamma\left(\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\frac{\beta-1}{2}}\right)}\right]^{1/\beta} \tag{12}$$

The inertia weighting factor ρ is expressed by Eq.(13).

$$\rho = 1 - k/max\_iteration \tag{13}$$

Therefore, based on the random roulette strategy, the individual position of sparrows x_{iD}^k is mutated by the Eq. (14). Then the optimal sparrow individual is also mutated by Eq. (15).

$$x_{iD}^{k'} = x_{iD}^k + L\,(\beta) \times \left(x_{iD}^k - x_{best}^k\right), if \; rand > \rho \tag{14}$$

$$x_{iD}^{best'} = x_{iD}^{best} \times (1 + L\,(\beta)) \; if \; rand < \rho \tag{15}$$

where $L\,(\beta)$ is a randomly distributed number drawn from Lévy distribution by Eqs. (10)~(12). Thus, the pseudo code of LFSSA can be expressed in **Algorithm 3**.

C. ELITE OPPOSITION-BASED LEARNING-SSA (EOBLSSA)

In this section, opposition-based learning mutation can enlarge the search space of the population, and improve the ability to avoid falling into local optimum prematurely and the convergence speed. The individuals with the best fitness are regarded as elite individuals who contain more

Algorithm 3 Pseudo Code of LFSSA

Set the parameters of LFSSA, the population of sparrows N, the number of discovers PD, the number of sparrows SD who to be on guard, the safety factor ST, $max\_iteration$, respectively.
Initialize the population X by Eq.(1).
While ($k < max\_iteration$)
  Calculate the fitness vaule F by Eq.(2), and the best solution f_{best} and worst solution f_{worst} by sort (F), respectively.
  Update the R_2, and ρ is obtainde by Eq. (13), repsectively.
  for $i = 1$: PD
    if ($R_2 < ST$)
      The location of sparrows is updated by Eq.(3.1)
    else
      The location of sparrows is updated by Eq.(3.2)
    End if
  End for
  for $i = (PD+1)$: N
    if($i < N/2$)
      The location of sparrows is updated by Eq.(4.1)
    else
      The location of sparrows is updated by Eq.(4.2)
    End if
  End for
    for $j = 1$: SD
    if($f_i < f_{gbest}$)
    The location of sparrows is updated by Eq.(5.1)
      else
      The location of sparrows is updated by Eq.(5.2)
    End if
  End for
    for $m = 1$: N
      if($rand > \rho$)
      The location of sparrows is updated by Eq.(14).
    End if
    End for
    The best solution and positon are updated.
    if($rand < \rho$)
    the optimal sparrow individual is mutated by Eq. (15).
    End if
    The best solution and positon are updated.
    $k = k + 1$;
End while
Return X_{gbest} and f_{gbest}

useful information to guide the population to converge to the global optimum. If the algorithm can finally achieve global optimum, the search space of the global optimum will inevitably include the search space of the elite individuals. Hence, the way to strengthen the search space neighborhood of elite individuals can improve the convergence speed and convergence accuracy of the algorithm. In addition, to improve the diversity of the population, the population of SSA is initialized by the elite opposition-based learning mutation, and when the SSA falls into local optimum, it is perturbed by the elite reverse learning mutation. The EOBLSSA algorithm is proposed by the elite opposition-based learning mutation. Some definitions of elite opposition-based learning are as follows [104]–[106].

Definition 1: Opposite number. Consider $p = [x_1, x_2, \ldots, x_D]$ is a point in D-dimension space, and $x_1, x_2, \ldots, x_D \in R, x_j \in [a_j, b_j]$, then the opposite point of p denoted by $p^* = \left[x_1^*, x_2^*, \ldots, x_D^*\right]$ can be calculated by Eq. (16).

$$x_j^* = k' \times (a_j + b_j) - x_j \tag{16}$$

where k' is a random number within [0,1].

Definition 2: Opposition-based optimization. For the minimum optimization problem, the fitness function is set as f. If there is a feasible solution X, the reverse solution is X'. If $f(X) < f(X')$, then X' is replaced by X.

Definition 3: Elite individuals. Consider $x_i(t) = [x_{i1}, x_{i2}, \ldots, x_{iD}]$ is a solution of the t-th iteration. And $x_i^*(t)$ is the opposite solution of $x_i(t)$, when $f(x_i(t)) \geq f(x_i^*(t))$, then $x_i(t)$ is denoted as the elite individuals $N_i(t)$. when $f(x_i(t)) < f(x_i^*(t))$, then $x_i(t)$ is denoted as the common individuals $Q_i(t)$. Consider the size of elite individuals is $\rho(0 < \rho \leq n, \rho \in N^+)$, then the elite individuals $N_\rho(t)$ can be denoted as Eq. (17).

$$\begin{aligned} N_\rho(t) = \left[N_1(t), N_2(t), \ldots, N_\rho(t)\right] \\ \in [x_1(t), x_2(t), \ldots, x_n(t)] \end{aligned} \tag{17}$$

where n is the total number of solutions.

Definition 4: Elite opposition-based learning solution. Consider $x_{i,j}(t)$ is the solution by the common individuals, then the elite opposition-based learning solution $x_{i,j}^*(t)$ can be denoted as Eq. (18).

$$x_{i,j}^*(t) = k'' \times \left(a_j(t) + b_j(t)\right) - x_{i,j}(t) \tag{18}$$

where k'' is a random number within [0,1], $a_j(t) = \min(N_{1j}(t), N_{2j}(t), \ldots, N_{\rho j}(t))$, $b_j(t) = max(N_{1j}(t), N_{2j}(t), \ldots, N_{\rho j}(t))$. $N_j(t) \in [a_j(t), b_j(t)]$.

Thus, the pseudo code of EOBLSSA can be expressed in **Algorithm 4**.

D. VARIABLE RADIUS PERTURBATION-SSA (VRSSA)

In this section, to enhance the ability to avoid falling to the local optimum and to improve the global convergence, the variable radius perturbation operator is introduced into the SSA, as shown in Fig. 2. The variable radius can be expressed by Eq.(19).

$$R = 1 - k/\max\_iteration \tag{19}$$

where k is the current iteration, and $max\_iteration$ is the maximum iteration, respectively.

Therefore, based on the random roulette strategy, the individual position of sparrows $x_{iD}^{k'}$ is mutated by the Eq. (20).

$$x_{iD}^{k'} = x_{best}^k + R \times (ub + lb) - lb \tag{20}$$

Algorithm 4 Pseudo Code of EOBLSSA

Set the parameters of EOBLSSA the population of sparrows N, the number of discovers PD, the number of sparrows SD who to be on guard, the safety factor ST, $max\_iteration$, the number ED of the elite individuals respectively.
Initialize the population X by Eq.(16), (17) and (18), respectively.
While ($k < max\_iteration$)
    Calculate the fitness vaule F by Eq.(2), and the best solution f_{best} and worst solution f_{worst} by sort (F), respectively.
    Update the R_2.
    for $i = 1$: PD
        if ($R_2 < ST$)
            The location of sparrows is updated by Eq.(3.1)
        else
            The location of sparrows is updated by Eq.(3.2)
            End if
    End for
    for $i = (PD+1)$: N
        if ($i < N/2$)
        The location of sparrows is updated by Eq.(4.1)
      else
        The location of sparrows is updated by Eq.(4.2)
        End if
    End for
        for $j = 1$: SD
        if($f_i < f_{gbest}$)
        The location of sparrows is updated by Eq.(5.1)
      else
        The location of sparrows is updated by Eq.(5.2)
        End if
    End for
        for $m = 1$: ED
        The location of elite individuals is updated by Eq. (17).
        The elite solution f_n is updated by Eq.(18).
        if ($f_n < f(x_n)$)
        The elite solution and location of elite individuals are updated.
        End if
    End for
        The best solution and positon are updated.
        $k = k + 1$;
End while
Return X_{gbest} and f_{gbest}

where ub and lb are the upper and lower bounds of variables, respectively.

In Fig. 2, with the current optimal solution as the reference point, as the number of iterations increases, R gradually becomes smaller and smaller, so that the search space is constantly shrinking. In the early iteration, both R and the search space are larger, which is conducive to further improving the global search ability. In the latter iteration,

Algorithm 5 Pseudo Code of VRSSA

Set the parameters of VRSSA, the population of sparrows N, the number of discovers PD, the number of sparrows SD who to be on guard, the safety factor ST, $max\_iteration$, respectively.
Initialize the population X by Eq.(1).
While ($k < max\_iteration$)
    Calculate the fitness vaule F by Eq.(2), and the best solution f_{best} and worst solution f_{worst} by sort (F), respectively.
    Update the R_2, and R is obtainde by Eq. (19), repsectively.
    for $i = 1$: PD
        if ($R_2 < ST$)
            The location of sparrows is updated by Eq.(3.1)
      else
            The location of sparrows is updated by Eq.(3.2)
            End if
    End for
    for $i = (PD+1)$: N
        if($i < N/2$)
        The location of sparrows is updated by Eq.(4.1)
      else
        The location of sparrows is updated by Eq.(4.2)
        End if
    End for
        for $j = 1$: SD
            if($f_i < f_{gbest}$)
        The location of sparrows is updated by Eq.(5.1)
            else
            The location of sparrows is updated by Eq.(5.2)
        End if
    End for
        The best solution and positon are updated.
        if ($rand < R$)
        the optimal sparrow individual is mutated by Eq. (20).
        End if
        The best solution and positon are updated.
        $k = k + 1$;
End while
Return X_{gbest} and f_{gbest}

both R and the search space are relatively smaller, which is conducive to improving the ability of the algorithm to jump out of the local optimum.

The VRSSA algorithm is proposed based on the variable radius perturbation operator, and the pseudo code of VRSSA is shown in **Algorithm 5.**

E. COMBINED MUTATED-SSA (CMSSA)

A variety of mutation operators can improve the local and global search capability of the SSA algorithm to some extent. However, a single mutation operator is often unable to balance between exploration and exploitation comprehensively. In order to further improve the performance of SSA, CMSSA is proposed by the combination of the improved Tent chaos map mutation operator, Lévy flights mutation operator, elite opposition-based learning mutation operator, and variable radius perturbation mutation operator. Initially, the population of SSA is initialized by the improved Tent chaos map mutation operator to enrich the diversity of the population. In addition, after the position of all sparrows is updated for the first time, the updating position mode with the combination of Lévy flights and elite opposition-based learning is introduced into the SSA to improve the global search ability of the algorithm. Finally, the optimal sparrow individual is mutated with the combination of variable radius perturbation operator and improved Tent chaos perturbation operator by the random roulette strategy to improve the local search ability of the algorithm. In the CMSSA, especially, the updating position of sparrows is different from that in the LFSSA, which is shown in Eq. (21).

$$x_{iD}^{k'} = x_{iD}^{k} + L(\beta) \times \left(x_{iD}^{k} - x_{worst}^{k}\right) \tag{21}$$

Thus, the pseudo code and flowchart of CMSSA can be described in **Algorithm 6** and shown in Fig. 3.

Algorithm 6 Pseudo Code of CMSSA

Set the parameters of CMSSA, such as the population of sparrows N, the number of discovers PD, the number of sparrows SD who to be on guard, the safety factor ST $max\_iteration$, the number ED of position updated using Lévy flights, and the number LD of position updated using elite opposition-based learning, respectively.
Initialize the population X by Eq.(6) and (7).
While ($k < max\_iteration$)
    Calculate the fitness vaule F by Eq.(2), and the best solution f_{gbest} and worst solution f_{worst} by sort (F), respectively.
    Update the R_2
    for $i = 1$: PD
        if ($R_2 < ST$)
            The location of sparrows is updated by Eq.(3.1)
        else
            The location of sparrows is updated by Eq.(3.2)
            End if
    End for
    for $i = (PD+1)$: N
            if($i < N/2$)
            The location of sparrows is updated by Eq.(4.1)
        else
            The location of sparrows is updated by Eq.(4.2)
            End if
    End for
            for $j = 1$: SD
            if($f_i < f_{gbest}$)
            The location of sparrows is updated by Eq.(5.1)
        else
            The location of sparrows is updated by Eq.(5.2)
            End if
    End for
            The best solution and position are updated.
        for $m = 1$: ED
        The location of elite individuals is updated by Eq. (17).
        The elite solution f_n is updated by Eq.(18).
        if ($f_{ED} < f(x_n)$)
        The elite solution and location of elite individuals are updated.
            End if
    End for
            for $l = 1$: LD
            The location of the rest sparrows is updated by Eq. (21).
            if($f_{LD} < f(x_n)$)
            The elite solution and location of sparrows are updated.
            End if
    End for
            The fitness value and position are updated.
            R is obtainde by Eq. (16).
            if ($rand < R$)
            the optimal sparrow individual is mutated by Eq. (20).
            else
            the optimal sparrow individual is mutated by Eq. (9).
            End if
            The best solution and position are updated.
            $k = k + 1$;
End while
Return X_{gbest} and f_{gbest}

F. COMPUTATIONAL COMPLEXITY ANALYSIS

In this part, all the proposed SSA variants mainly consist of the following phases: initialization, fitness evaluation, and sorting, and the sparrow's location update. Among them, N denotes the number of sparrows, D denotes the dimension of functions, T denotes the maximum number of iterations, PD denotes the number of discoverers, SD denotes the number of sparrows who to be on guard, ED denotes the number of sparrows who updated by the EOBL, and LD denotes the number of sparrows who updated by the LF. In SSA, the computational complexity of initialization is $O(N \times D)$, the computational complexity of fitness evaluation and sorting is $O(N \times D)$, the computational complexity of the sparrow's location update is $O(T \times (PD \times D + SD \times D + (N\text{-}SD\text{-}PD) \times D)$, namely $O(T \times N \times D)$. Therefore, the computational complexity of SSA is $O(N)$. Compared with SSA, the computational complexity of all the proposed SSA variants is mainly different from that of the sparrow's location update phase. In terms of the sparrow's location update phase, the computational complexity of both ITSSA and VRSSA is equal to the SSA, namely $O(T \times N \times D)$. The computational complexity of EOBLSSA is $O(T \times (N+ED) \times D)$, namely $O(T \times N \times D)$. The computational complexity of LFSSA is $O(T \times 2N \times D)$. The computational complexity of CMSSA is $O(T \times (N + (ED + LD)) \times D)$, namely $O(T \times 2N \times D)$. To sum up, the computational complexity of all the proposed SSA variants and the basic SSA is $O(N)$.

IV. EXPERIMENTAL RESULTS

In this section, the proposed algorithms are tested on 31 well-known benchmark test functions [107]–[109] and the results are compared to other state-of-the-art algorithms. There are four groups of these benchmark test functions in Appendix A including the unimodal functions (see Table 5),

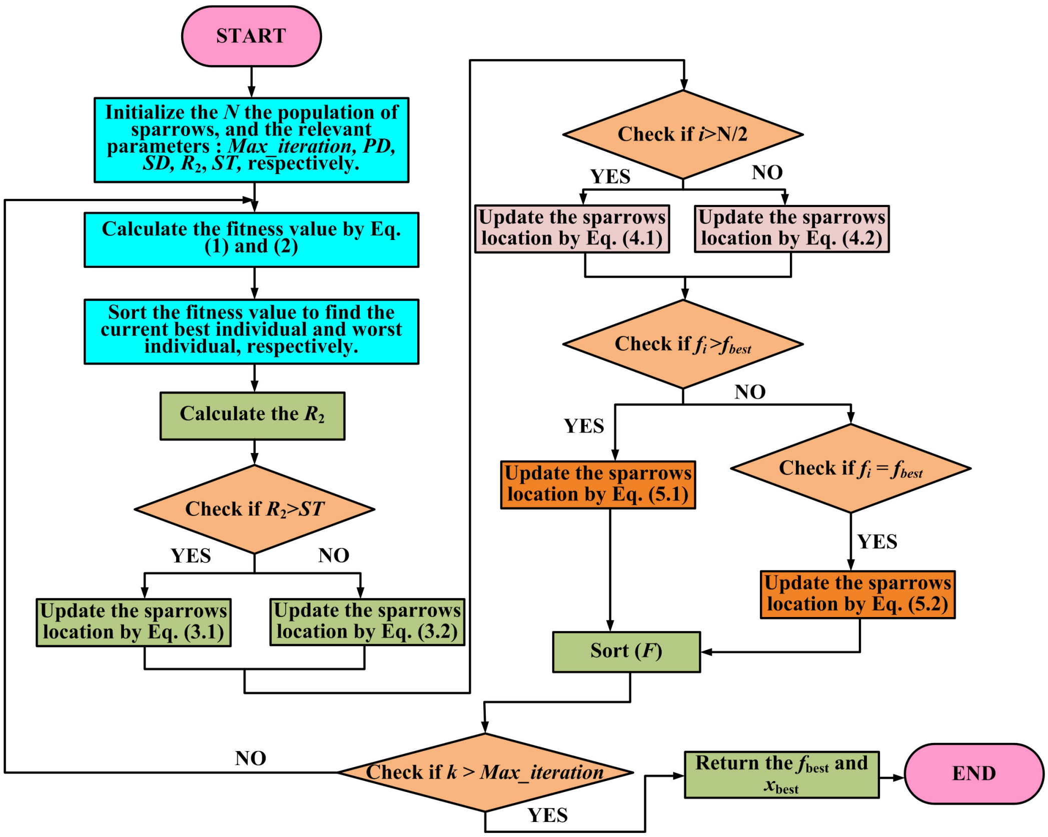

FIGURE 1. The flowchart of SSA algorithm.

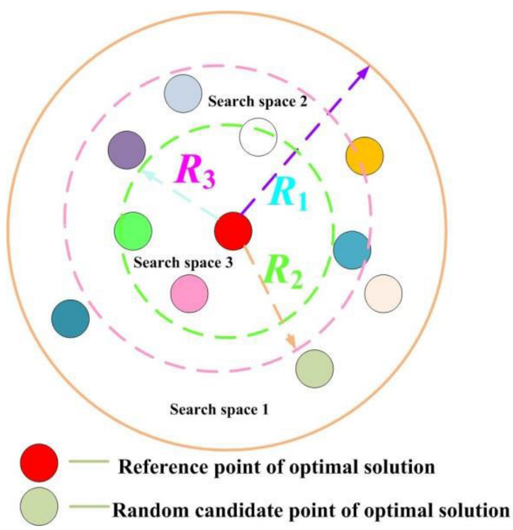

FIGURE 2. The variable radius perturbation.

the multimodal functions (see Table 6), the fixed-dimension multimodal functions (see Table 7), and 8 hybrid and composition functions on CEC2017 (see Table 8). The 2D landscapes of a few benchmark functions are shown in Fig. 4. The experiments are mainly divided into four parts. In the first part, the performance of the proposed algorithms is tested on 31 benchmark test functions compared to the basic SSA. In the second part, the scalability test of the best of these proposed algorithms. In the third part, the performance of the best-proposed algorithm is evaluated on 31 benchmark test functions compared to other algorithms. In the fourth part, to demonstrate the efficiency of the best-proposed algorithm, the best-proposed algorithm is also employed on eight constrained real-world optimization problems.

A. EXPERIMENTAL SETUP

All experiments and algorithms are carried out in MATLAB R2019a version software. The simulation environments are performed on Windows 8.1 (64 bit) systems with a Core i7 processor with 2.3 GHz and 64 GB memory. All the experimental results are obtained by all the algorithms running 30 times independently in terms of Average (Avg.), Median (Med.), and Standard deviation (Std.) values. The Wilcoxon rank-sum test at a 5% level of significance [110] and the Friedman test [111] are performed in a statistically significant way. By the results of these algorithms running 30 times independently, the p-values of the Wilcoxon statistical test and the average ranking value (ARV.) of the Friedman test were obtained.

B. THE PERFORMANCE EVALUATION OF ALL THE PROPOSED VARIANTS COMPARED TO BASIC SSA

In this section, initially, the parameter tuning on the proposed SSA variants is performed. In addition, the performance of

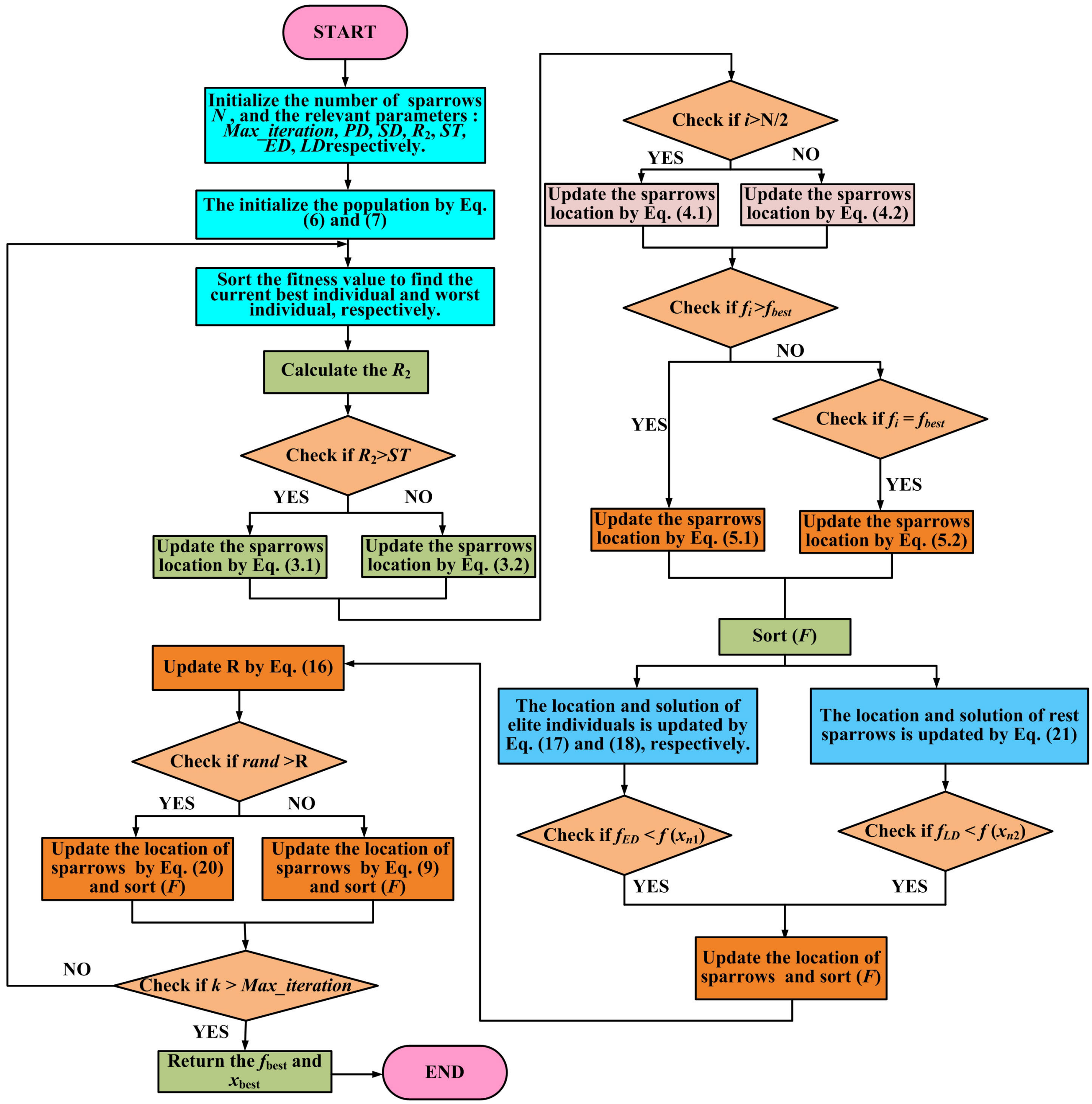

FIGURE 3. The flowchart of CMSSA algorithm.

the proposed SSA variants is evaluated on 31 benchmark test functions.

1) INVESTIGATING THE INFLUENCE OF PARAMETERS ON SSA VARIANTS

The parameter tuning plays an essential role in the performance evaluation of metaheuristics. The SSA variants mainly involve five parameters namely the maximum number of iterations, number of search agents, *PD*, *SD*, and *ST*. The sensitivity analysis of these parameters has been discussed by varying their values on F_1, F_5, F_{12}, F_{13}, and F_{22} functions. All results in this section are obtained under the 30 times independent experiments.

a: MAXIMUM NUMBER OF ITERATIONS

All proposed algorithms were run by the different maximum number of iterations. The maximum number of iterations was set to 100, 500, 1000, respectively. The obtained Avg. values were shown in Table 9. For all proposed algorithms, the results reveal that the Avg. values become better when the maximum number of iterations increases.

b: NUMBER OF SEARCH AGENTS

In order to evaluate the effect of the number of search agents on all test functions, for all proposed algorithms, the number of search agents was set to 10, 30, 50, respectively. All proposed algorithms were run by different search agents.

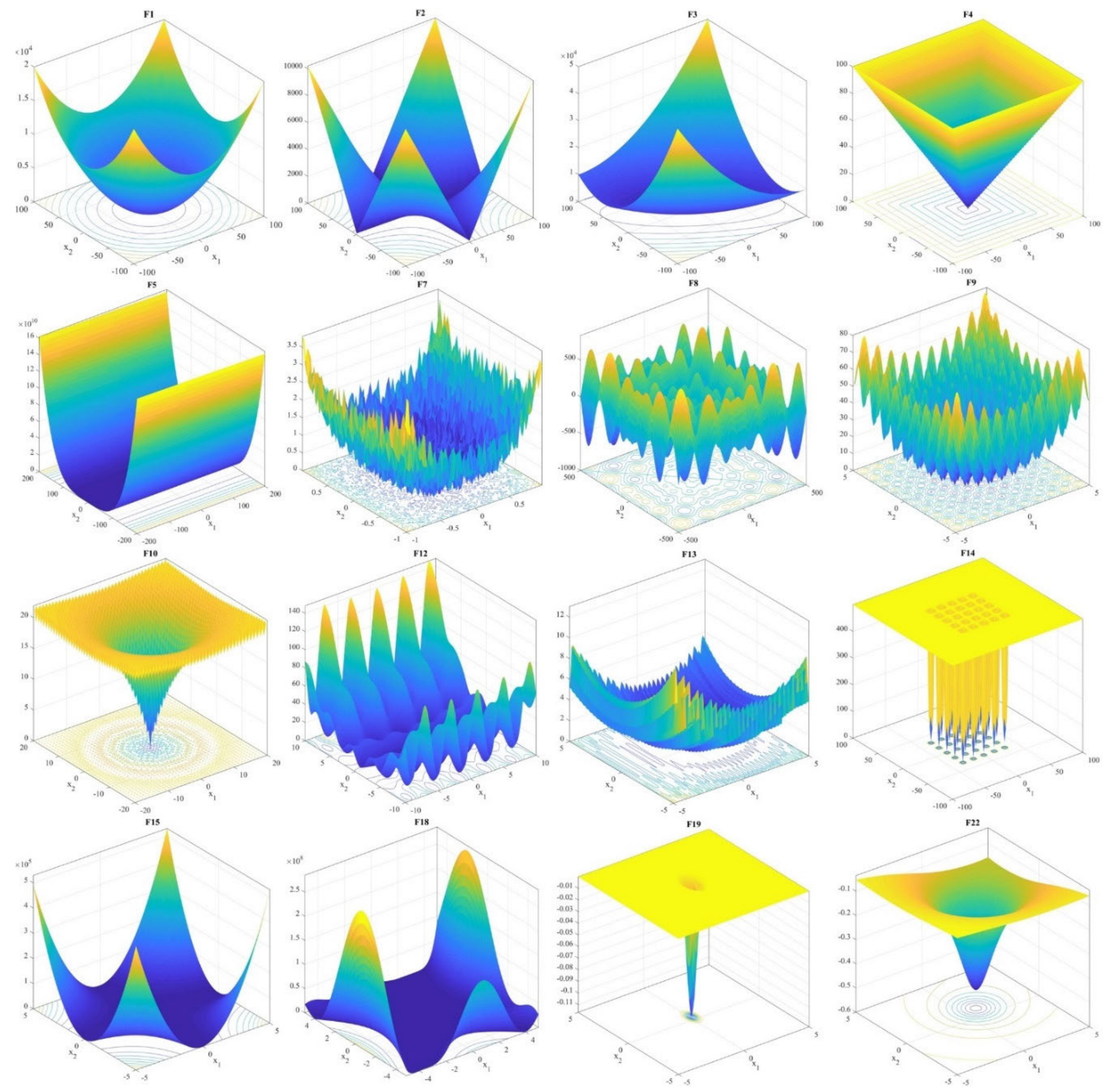

FIGURE 4. 2D landscapes of some benchmark functions.

Table 10 shows the Avg. values obtained by different search agents. It was found that all proposed algorithms could provide better results on most of the test functions with the increase of search agents.

c: VARIATION IN PARAMETER PD

All proposed algorithms were run for different values of parameter *PD* keeping other parameters fixed (i.e., the maximum number of iterations, number of search agents, *SD* and *ST*). The values of *PD* used in experiments were set to 0.1, 0.2, and 0.3, respectively. Table 11 represents the results of Avg. values for all proposed algorithms with different *PD* values. As it can be seen in Table 11, for ITSSA, LFSSA, EOBLSSA, and VRSSA algorithms, it is observed that $PD = 0.3$ is a reasonable value for most of the test functions. However, it shows that $PD = 0.2$ is an appropriate value for the CMSSA algorithm on most of the test functions.

d: VARIATION IN PARAMETER SD

All proposed algorithms were run for different values of parameter *SD* by fixing other parameters (i.e., the maximum number of iterations, number of search agents, *PD* and *ST*). The values of *SD* used in experiments were set to 0.1 and

TABLE 9. The obtained average values under different iterations.

Functions	Iterations	ITSSA	LFSSA	EOBLSSA	VRSSA	CMSSA
	100	1.09E-32	3.96E-50	7.23E-117	5.52E-31	**0.00E+00**
F1	500	7.30E-64	1.46E-148	**0.00E+00**	7.16E-49	**0.00E+00**
	1000	**4.48E-83**	**9.75E-273**	**0.00E+00**	**5.38E-68**	**0.00E+00**
	100	2.26E+00	2.51E+00	8.35E-01	1.93E+00	6.16E-03
F5	500	7.45E-01	4.29E-01	2.73E-01	8.90E-01	6.73E-05
	1000	**4.09E-01**	**1.44E-01**	**6.25E-02**	**2.76E-01**	**1.32E-05**
	100	2.95E-02	2.62E-02	2.18E-02	2.78E-02	1.17E-06
F12	500	4.73E-03	4.62E-03	3.33E-03	8.48E-03	2.42E-08
	1000	**9.85E-04**	**8.67E-04**	**6.72E-04**	**8.95E-04**	**3.16E-09**
	100	4.79E-01	3.83E-01	2.96E-01	4.21E-01	9.98E-06
F13	500	3.01E-01	2.95E-01	2.05E-01	2.73E-01	3.00E-07
	1000	**2.43E-01**	**2.24E-01**	**1.03E-01**	**2.12E-01**	**3.24E-08**
	100	-6.6195	-6.4497	-6.7363	-5.7414	-6.4889
F22	500	-8.4212	-8.2313	-8.3895	-8.6752	-8.9354
	1000	**-8.9852**	**-9.4716**	**-8.9400**	**-9.3324**	**-10.2257**

TABLE 10. The obtained average values under different search agents.

Functions	Search agents	ITSSA	LFSSA	EOBLSSA	VRSSA	CMSSA
	10	1.99E-33	2.40E-95	2.05E-10	2.47E-10	**0.00E+00**
F1	30	2.54E-91	1.40E-196	**0.00E+00**	6.86E-64	**0.00E+00**
	50	**1.96E-185**	**0.00E+00**	**0.00E+00**	**1.14E-153**	**0.00E+00**
	10	4.39E+00	1.83E+00	3.75E+00	5.34E+00	1.01E-04
F5	30	4.02E-01	3.06E-01	1.13E-01	2.96E-01	1.04E-05
	50	**3.85E-02**	**3.72E-02**	**2.01E-02**	**2.61E-02**	**3.18E-06**
	10	7.77E-03	6.29E-03	9.13E-03	7.23E-03	7.24E-08
F12	30	9.43E-04	7.58E-04	7.99E-04	9.37E-04	1.39E-09
	50	**1.45E-04**	**1.38E-04**	**1.14E-04**	**1.44E-04**	**3.29E-10**
	10	4.43E-01	3.10E-01	4.79E-01	3.59E-01	1.28E-06
F13	30	1.92E-01	2.13E-01	1.21E-01	2.46E-01	3.63E-08
	50	**9.64E-01**	**1.25E-01**	**5.95E-02**	**1.56E-01**	**3.48E-09**
	10	-6.8855	-7.4132	-7.7528	-6.0006	-7.5308
F22	30	-9.0035	-9.3398	-9.4028	-9.5567	-9.6036
	50	**-10.3893**	**-9.6942**	**-10.0485**	**-10.1236**	**-10.0485**

TABLE 11. The obtained average values under different *PD*.

Functions	*PD*	ITSSA	LFSSA	EOBLSSA	VRSSA	CMSSA
	0.1	1.46E-41	1.87E-227	**0.00E+00**	8.78E-19	**0.00E+00**
F1	0.2	2.22E-76	**2.03E-267**	**0.00E+00**	2.15E-57	**0.00E+00**
	0.3	**2.40E-84**	8.54E-223	**0.00E+00**	**1.78E-67**	**0.00E+00**
	0.1	1.11E+00	3.14E-01	1.61E-01	1.36E+00	1.13E-05
F5	0.2	4.15E-01	2.47E-01	7.96E-02	**1.33E-01**	**1.06E-05**
	0.3	**1.21E-01**	**1.15E-01**	**3.94E-02**	1.34E-01	1.33E-05
	0.1	3.20E-03	2.08E-03	2.18E-03	2.93E-03	4.86E-09
F12	0.2	8.57E-04	7.57E-04	7.34E-04	1.07E-03	**1.58E-09**
	0.3	**4.63E-04**	**4.24E-04**	**3.72E-04**	**5.20E-04**	3.51E-09
	0.1	3.30E-01	3.02E-01	1.75E-01	3.74E-01	4.15E-08
F13	0.2	2.29E-01	1.69E-01	8.84E-02	2.53E-01	**3.65E-08**
	0.3	**1.70E-01**	**1.14E-01**	**7.20E-02**	**1.06E-01**	1.14E-07
	0.1	-7.9179	-8.8073	-8.9855	-8.8911	-9.4713
F22	0.2	-9.0158	-9.0275	-9.8714	**-9.4691**	**-10.0840**
	0.3	**-10.0485**	**-9.8570**	**-10.4029**	-9.2769	-10.0573

0.2. For all proposed algorithms, the Avg. values are attainted by different SD which is shown in Table 12. The results demonstrate that when SD is set to 0.1, the CMSSA and ITSSA algorithms provide better results on most of the test functions. And when SD is set to 0.2, EOBLSSA, LFSSA, and VRSSA algorithms attain better results on most of the test functions. Therefore, $SD = 0.1$ is a suitable value for CMSSA and ITSSA algorithms, and $SD = 0.2$ is a reasonable value for EOBLSSA, LFSSA, and VRSSA algorithms.

e: VARIATION IN PARAMETER ST

All proposed algorithms were run for various values of parameter ST when other parameters (i.e., the maximum number of iterations, number of search agents, PD and SD)

TABLE 12. The obtained average values under different SD.

Functions	*SD*	ITSSA	LFSSA	EOBLSSA	VRSSA	CMSSA
F1	0.1	**8.22E-93**	6.62E-229	**0.00E+00**	4.80E-42	**0.00E+00**
	0.2	2.33E-91	**3.47E-303**	**0.00E+00**	**2.09E-55**	**0.00E+00**
F5	0.1	2.33E-01	4.06E-01	5.73E-02	3.70E-01	**7.16E-06**
	0.2	**9.25E-02**	**1.83E-01**	**5.44E-02**	**1.31E-01**	3.95E-05
F12	0.1	**7.68E-04**	7.70E-04	6.79E-04	**8.79E-04**	4.62E-09
	0.2	5.52E-03	**5.93E-04**	**4.50E-04**	9.94E-03	**2.57E-09**
F13	0.1	**2.97E-01**	**1.89E-01**	1.01E-01	**2.44E-01**	**3.73E-08**
	0.2	3.00E-01	2.03E-01	**8.07E-02**	2.71E-01	4.37E-08
F22	0.1	**-9.5305**	**-9.8484**	**-10.2230**	-8.6732	**-9.4484**
	0.2	-8.4648	-9.1626	-9.5078	**-9.4546**	-9.3398

TABLE 13. The obtained average values under different *ST*.

Functions	*ST*	ITSSA	LFSSA	EOBLSSA	VRSSA	CMSSA
F1	0.5	4.32E-46	2.00E-91	7.33E-212	4.83E-22	**0.00E+00**
	0.6	1.32E-51	1.68E-150	**0.00E+00**	2.22E-32	**0.00E+00**
	0.7	8.41E-68	4.42E-185	**0.00E+00**	1.62E-42	**0.00E+00**
	0.8	9.37E-89	**6.61E-286**	**0.00E+00**	**3.30E-65**	**0.00E+00**
	0.9	**2.30E-108**	1.43E-258	**0.00E+00**	6.11E-63	**0.00E+00**
F5	0.5	**1.0E-01**	**3.22E-02**	**1.51E-02**	**1.09E-01**	1.63E-05
	0.6	1.15E-01	7.55E-02	2.83E-02	1.16E-01	1.16E-05
	0.7	1.79E-01	2.21E-01	4.56E-02	1.41E-01	7.90E-06
	0.8	4.15E-01	2.47E-01	1.03E-01	4.19E-01	**1.37E-06**
	0.9	4.20E-01	8.83E-01	4.42E-01	5.37E-01	1.69E-05
F12	0.5	**1.37E-04**	**8.10E-05**	**9.05E-05**	**1.17E-04**	3.71E-09
	0.6	4.57E-03	1.97E-04	1.32E-04	2.44E-04	3.04E-09
	0.7	5.85E-04	3.97E-04	3.30E-04	5.26E-03	2.16E-09
	0.8	1.11E-03	8.59E-04	7.56E-04	1.04E-03	**2.05E-09**
	0.9	2.53E-03	2.34E-03	7.17E-03	3.06E-03	6.66E-09
F13	0.5	**1.71E-01**	2.03E-01	5.52E-02	2.08E-01	5.00E-09
	0.6	2.26E-01	2.02E-01	**4.77E-02**	**1.79E-01**	8.76E-09
	0.7	1.64E-01	2.54E-01	7.46E-02	1.84E-01	1.53E-08
	0.8	2.38E-01	**1.94E-01**	1.01E-01	3.01E-01	**3.44E-09**
	0.9	3.03E-01	2.66E-01	1.64E-01	2.83E-01	5.55E-08
F22	0.5	-10.0485	**-10.4029**	**-10.4029**	-9.9045	-10.2243
	0.6	**-10.2257**	**-10.4029**	**-10.4029**	**-10.0589**	**-10.2257**
	0.7	-9.6942	-10.0432	-9.3801	-9.1806	**-10.2257**
	0.8	-8.9892	-10.0520	-10.0417	-9.5497	-9.6799
	0.9	-8.2527	-8.2768	-7.7437	-9.3074	-9.6571

TABLE 14. Parameter setting of all proposed algorithms and SSA.

Algorithms	Population size	Maximum number of iterarions	Parameters
CMSSA	30	500	*PD*=0.2; *SD*=0.1; *ST*=0.8; *ED*=0.5; *LD*=0.5
ITSSA	30	500	*PD*=0.3; *SD*=0.1; *ST*=0.5
SSA	30	500	*PD*=0.2, *SD*=0.1, *ST*=0.8.
VRSSA	30	500	*PD*=0.3; *SD*=0.2; *ST*=0.6
LFSSA	30	500	*PD*=0.3; *SD*=0.2; *ST*=0.5
EOBLSSA	30	500	*PD*=0.3; *SD*=0.2; *ST*=0.5

were fixed. In these experiments, the values of ST were set 0.1, 0.2, 0.3, 0.4 and 0.5, respectively. The Avg. values of all proposed algorithms are shown in Table 13. It is observed that $ST = 0.5$ is a proper value for ITSSA, LFSSA, and EOBLSSA algorithms in many cases, and $ST = 0.6$ is suitable for the VRSSA algorithm, and $ST = 0.8$ is an advisable value for the CMSSA algorithm on most of the test functions.

2) THE PERFORMANCE OF ALL THE PROPOSED ALGORITHMS

In this section, to evaluate the performance of the proposed SSA variants more comprehensively, compared to the basic SSA algorithm, the performance of all proposed algorithms (i.e. ITSSA, LFSSA, EOBLSSA, VRSSA and CMSSA) is tested on 31 benchmark test functions. According to the results of the parameter tuning of all proposed algorithms, the parameter settings of all proposed algorithms and SSA are given in Table 14. Table 15 reveals the optimal values obtained by all proposed algorithms compared to the basic SSA algorithm. As it can be seen from Table 15, the CMSSA algorithm can attain the global optimum value on 11 out of 31 benchmark functions (i.e., F_1, F_2, F_3, F_4, F_9, F_{11}, F_{16}, F_{17}, F_{18}, F_{19}, and F_{22}) with the smallest Std. value. For F_1, F_2, F_3, and F_4, CMSSA provides the best results in terms of Avg., Std. and Med. values. EOBLSSA and LFSSA attain

TABLE 15. The statistical experimental results of 31 benchmark functions.

Function	Index	ITSSA	LFSSA	EOBLSSA	VRSSA	CMSSA	SSA
	Avg.	9.40E-75	5.97E-146	0.00E+00	1.36E-47	**0.00E+00**	1.13E-44
F1	Med.	1.23E-108	9.57E-196	0.00E+00	5.30E-79	**0.00E+00**	1.50E-77
	Std.	5.14E-74	3.22E-145	0.00E+00	7.45E-47	**0.00E+00**	6.21E-44
	Avg.	1.86E-34	1.08E-60	5.49E-307	2.09E-26	**0.00E+00**	2.57E-20
F2	Med.	5.23E-54	6.30E-108	0.00E+00	2.96E-44	**0.00E+00**	9.36E-46
	Std.	9.82E-34	5.91E-60	0.00E+00	1.14E-25	**0.00E+00**	1.40E-19
	Avg.	4.58E-34	7.30E-56	1.68E-125	1.23E-26	**0.00E+00**	3.81E-28
F3	Med.	1.17E-52	7.05E-83	1.04E-192	8.39E-50	**0.00E+00**	3.36E-54
	Std.	2.38E-33	4.00E-55	9.23E-125	6.77E-26	**0.00E+00**	2.08E-27
	Avg.	4.27E-27	3.27E-72	7.15E-302	3.15E-26	**0.00E+00**	2.56E-27
F4	Med.	2.40E-52	3.27E-97	2.50E-319	7.91E-40	**0.00E+00**	1.25E-42
	Std.	2.34E-26	1.45E-71	0.00E+00	1.62E-25	**0.00E+00**	9.82E-27
	Avg.	7.96E-01	9.61E-01	3.50E-01	9.04E-01	**8.34E-05**	1.00E+00
F5	Med.	1.94E-01	2.97E-01	6.51E-02	8.21E-02	**1.24E-05**	9.28E-02
	Std.	1.16E+00	1.43E+00	6.71E-01	1.46E+00	**2.45E-04**	1.43E+00
	Avg.	2.99E-02	2.54E-02	2.84E-02	3.20E-02	**3.35E-07**	3.32E-02
F6	Med.	2.88E-02	2.36E-02	2.74E-02	3.08E-02	**4.69E-08**	3.34E-02
	Std.	1.06E-02	9.90E-03	9.93E-03	1.18E-02	**7.00E-07**	1.13E-02
	Avg.	6.57E-04	2.06E-04	1.20E-04	6.07E-04	**1.85E-04**	6.68E-04
F7	Med.	6.04E-04	1.64E-04	8.12E-05	4.60E-04	**1.29E-04**	4.75E-04
	Std.	5.34E-04	1.81E-04	1.13E-04	5.47E-04	**1.80E-04**	4.39E-04
	Avg.	-7.70E+03	-7.21E+03	-7.80E+03	-7.11E+03	**-7.71E+03**	-8.23E+03
F8	Med.	-8.00E+03	-5.76E+03	-8.13E+03	-6.44E+03	**-8.04E+03**	-8.58E+03
	Std.	2.88E+03	2.77E+03	2.62E+03	1.67E+03	**2.52E+03**	3.12E+03
	Avg.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00
F9	Med.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00
	Avg.	8.88E-16	8.88E-16	8.88E-16	8.88E-16	**8.88E-16**	8.88E-16
F10	Med.	8.88E-16	8.88E-16	8.88E-16	8.88E-16	**8.88E-16**	8.88E-16
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00
	Avg.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00
F11	Med.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00
	Avg.	6.29E-03	6.62E-03	5.52E-03	6.07E-03	**3.53E-08**	8.58E-03
F12	Med.	6.60E-03	6.53E-03	5.71E-03	6.04E-03	**5.89E-09**	6.64E-03
	Std.	2.08E-03	1.40E-03	1.36E-03	1.55E-03	**8.30E-08**	1.00E-02
	Avg.	2.46E-01	2.81E-01	1.59E-01	2.69E-01	**2.60E-07**	2.92E-01
F13	Med.	1.91E-01	1.62E-01	1.55E-01	1.97E-01	**9.36E-08**	2.40E-01
	Std.	2.32E-01	2.37E-01	9.72E-01	2.37E-01	**5.77E-07**	2.01E-01
	Avg.	1.26E+01	9.27E+00	1.09E+01	1.08E+01	**2.31E+00**	1.17E+01
F14	Med.	1.26E+01	1.26E+01	1.26E+01	1.26E+01	**1.99E+00**	1.26E+01
	Std.	1.48E-13	4.75E+00	3.46E+00	3.62E+00	**2.26E+00**	2.76E+00
	Avg.	4.73E-04	5.44E-04	5.10 E-04	4.72E-04	**3.09E-04**	5.17E-04
F15	Med.	4.38E-04	4.71E-04	5.10E-04	4.26E-04	**3.08E-04**	5.09E-04
	Std.	1.94E-04	2.78E-04	1.43E-04	1.86E-04	**3.55E-06**	2.08E-04
F16	Avg.	-1.0316	-1.0316	-1.0316	-1.0316	**-1.0316**	-1.0316
	Med.	-1.0316	-1.0316	-1.0316	-1.0316	**-1.0316**	-1.0316
	Std.	5.53E-16	6.18E-16	5.45E-16	4.68E-16	**5.45E-16**	5.53E-16
	Avg.	3.97E-01	3.97E-01	3.97E-01	3.97E-01	**3.97E-01**	3.97E-01
F17	Med.	3.97E-01	3.97E-01	3.97E-01	3.97E-01	**3.97E-01**	3.97E-01
	Std.	1.57E-11	8.56E-12	1.19E-11	1.79E-14	**0.00E+00**	3.10E-10
	Avg.	3.00E+00	3.00E+00	3.00E+00	3.00E+00	**3.00E+00**	3.00E+00
F18	Med.	3.00E+00	3.00E+00	3.00E+00	3.00E+00	**3.00E+00**	3.00E+00
	Std.	6.15E-15	4.76E-15	6.25E-15	7.48E-15	**1.83E-14**	9.98E-15

TABLE 15. ***(Continued.)*** **The statistical experimental results of 31 benchmark functions.**

	Avg.	-3.8628	-3.8628	-3.8628	-3.8581	**-3.8628**	-3.8628
F19	Med.	-3.8628	-3.8628	-3.8628	-3.8603	**-3.8628**	-3.8628
	Std.	5.01E-06	4.44E-07	1.47E-06	4.65E-03	**2.73E-15**	7.60E-07
	Avg.	-3.17E+00	-3.27E+00	-3.23E+00	-3.24E+00	**-3.14E+00**	-3.24E+00
F20	Med.	-3.17E+00	-3.32E+00	-3.19E+00	-3.26E+00	**-3.14E+00**	-3.20E+00
	Std.	1.64E-02	6.68E-02	7.53E-02	6.79E-02	**2.96E-02**	7.54E-02
	Avg.	-7.74E+00	-9.00E+00	-8.90E+00	-7.32E+00	**-8.45E+00**	-8.28E+00
F21	Med.	-5.05E+00	-1.01E+01	-1.01E+01	-5.68E+00	**-1.01E+01**	-1.01E+01
	Std.	2.56E+00	2.12E+00	2.12E+00	2.01E+00	**2.44E+00**	2.49E+00
	Avg.	-9.47E+00	-9.23E+00	-9.50E+00	-9.51E+00	**-1.04E+01**	-7.39E+00
F22	Med.	-1.04E+01	-1.04E+01	-1.04E+01	-6.96E+00	**-1.04E+01**	-1.04E+01
	Std.	2.00E+00	2.25E+00	2.00E+00	2.09E+00	**0.00E+00**	2.01E+00
	Avg.	-9.09E+00	-9.63E+00	-9.43E+00	-8.26E+00	**-9.81E+00**	-9.63E+00
F23	Med.	-10.5364	-10.5364	-10.5364	-10.5364	**-10.5364**	-8.0462
	Std.	2.42E+00	2.05E+00	2.19E+00	2.04E+00	**1.86E+00**	1.87E+00
	Avg.	1.91E+03	1.90E+03	1.93E+03	1.92E+03	**1.78E+03**	1.93E+03
F24	Med.	1.91E+03	1.85E+03	1.93E+03	1.89E+03	**1.76E+03**	1.91E+03
	Std.	5.24E+01	1.18E+02	1.08E+02	1.10E+02	**4.90E+01**	1.24E+02
	Avg.	3.81E+03	1.63E+04	1.53E+04	1.61E+04	**5.30E+03**	1.93E+04
F25	Med.	3.93E+03	8.21E+03	1.19E+04	1.16E+04	**3.67E+03**	1.64E+04
	Std.	6.01E+02	1.41E+04	1.06E+04	1.24E+04	**5.21E+03**	1.28E+04
	Avg.	1.69E+04	9.59E+05	2.38E+06	1.26E+06	**1.11E+04**	2.08E+06
F26	Med.	6.39E+03	1.01E+05	7.76E+05	1.75E+05	**7.34E+03**	1.45E+05
	Std.	4.43E+04	1.85E+06	2.68E+06	2.16E+06	**9.94E+03**	4.78E+06
	Avg.	2.28E+03	2.29E+03	2.26E+03	2.25E+03	**2.22E+03**	2.31E+03
F27	Med.	2.24E+03	2.30E+03	2.26E+03	2.25E+03	**2.23E+03**	2.30E+03
	Std.	8.28E+01	8.73E+01	9.15E+01	1.02E+02	**8.34E+01**	1.14E+02
	Avg.	2.38E+03	2.37E+03	2.35E+03	2.37E+03	**2.34E+03**	2.38E+03
F28	Med.	2.42E+03	2.37E+03	2.36E+03	2.37E+03	**2.35E+03**	2.38E+03
	Std.	6.47E+01	2.40E+01	4.72E+01	5.06E+01	**5.90E+01**	4.15E+01
	Avg.	2.90E+03	2.70E+03	2.72E+03	2.71E+03	**2.65E+03**	2.72E+03
F29	Med.	2.78E+03	2.70E+03	2.73E+03	2.71E+03	**2.65E+03**	2.71E+03
	Std.	2.44E+02	3.41E+01	4.72E+01	4.39E+01	**2.35E+01**	3.96E+01
	Avg.	2.89E+03	2.86E+03	2.85E+03	2.87E+03	**2.76E+03**	2.87E+03
F30	Med.	2.84E+03	2.88E+03	2.86E+03	2.88E+03	**2.79E+03**	2.87E+03
	Std.	1.92E+02	1.13E+02	9.69E+01	1.07E+02	**9.18E+01**	6.67E+01
	Avg.	3.51E+03	3.27E+03	3.27E+03	3.23E+03	**3.13E+03**	3.30E+03
F31	Med.	3.50E+03	3.24E+03	3.26E+03	3.24E+03	**3.13E+03**	3.29E+03
	Std.	5.72E+01	9.11E+01	8.49E+01	5.85E+01	**3.87E+01**	9.67E+01
Friedman test	**Rank**	5	3	2	4	**1**	6
	AGV.	4.1129	3.4354	3.1129	3.8871	**1.8548**	4.5967

the second and third best results. For F_4, the results obtained by SSA are better than VRSSA and ITSSA in terms of Avg. and Std. values. For F_5, the CMSSA attains the best results. ITSSA and VRSSA obtain better results than LFSSA and SSA in terms of Avg. and Med. values. For F_6, CMSSA obtains the best results. LFSSA and EOBLSSA provide the second and third best results. For F_7, the results obtained by EOBLSSA are better than other algorithms, and CMSSA attains the second-best results. For F_8, SSA provides the best Avg. and Med. values, and VRSSA attains the best Std. value. For F_9, F_{10} and F_{11}, all algorithms provide the same results in terms of Avg., Med. and Std. values. For F_{12}, CMSSA and EOBLSSA obtain the first and second results, VRSSA attains competitively better results than ITSSA, LFSSA, and SSA. For F_{13}, CMSSA and EOBLSSA can obtain the first and second results in terms of Avg. and Med. values. CMSSA and SSA provide the first and second Std. value. For F_{14}, VRSSA and EOBLSSA attain better results of Avg. and Med. values, and LFSSA obtains the best Std. value. For F_{15}, CMSSA provides the best results. VRSSA and EOBLSSA obtain competitive results which are better than LFSSA, ITSSA, and SSA. For F_{16}, VRSSA provides the best results. CMSSA and EOBLSSA obtain competitive results.

For F_{17} and F_{18}, all algorithms attain the same Avg. and Med. values. However, CMSSA and VRSSA provide the

TABLE 16. The scalability results on 13 benchmark functions.

F1	**Avg.**		**Med.**		**Std.**	
	CMSSA	**SSA**	**CMSSA**	**SSA**	**CMSSA**	**SSA**
30	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00
500	**0.00E+00**	8.22E-60	**0.00E+00**	6.41E-93	**0.00E+00**	4.50E-59
1000	**0.00E+00**	2.81E-55	**0.00E+00**	5.41E-91	**0.00E+00**	1.54E-54
F2	**Avg.**		**Med.**		**Std.**	
	CMSSA	**SSA**	**CMSSA**	**SSA**	**CMSSA**	**SSA**
30	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00
500	**0.00E+00**	1.04E-30	**0.00E+00**	6.92E-47	**0.00E+00**	5.69E-30
1000	**0.00E+00**	1.02E-39	**0.00E+00**	4.05E-54	**0.00E+00**	5.54E-39
F3	**Avg.**		**Med.**		**Std.**	
	CMSSA	**SSA**	**CMSSA**	**SSA**	**CMSSA**	**SSA**
30	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00
500	**0.00E+00**	3.83E-36	**0.00E+00**	8.27E-51	**0.00E+00**	2.05E-35
1000	**0.00E+00**	2.00E-37	**0.00E+00**	2.76E-54	**0.00E+00**	1.02E-36
F4	**Avg.**		**Med.**		**Std.**	
	CMSSA	**SSA**	**CMSSA**	**SSA**	**CMSSA**	**SSA**
30	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00
500	**0.00E+00**	6.07E-31	**0.00E+00**	5.08E-57	**0.00E+00**	3.32E-30
1000	**0.00E+00**	2.85E-34	**0.00E+00**	2.62E-43	**0.00E+00**	1.49E-33
F5	**Avg.**		**Med.**		**Std.**	
	CMSSA	**SSA**	**CMSSA**	**SSA**	**CMSSA**	**SSA**
30	**1.39E-05**	2.16E-01	**9.76E-07**	5.96E-02	**3.00E-05**	3.28E-01
500	**1.09E-03**	5.86E+01	**1.93E-04**	2.69E+01	**2.74E-03**	6.37E+01
1000	**1.45E-03**	1.47E+02	**3.93E-04**	4.04E+01	**2.65E-03**	2.18E+02
F6	**Avg.**		**Med.**		**Std.**	
	CMSSA	**SSA**	**CMSSA**	**SSA**	**CMSSA**	**SSA**
30	**3.15E-08**	6.36E-03	**6.12E-09**	6.23E-03	**5.51E-08**	2.63E-03
500	**3.40E-06**	2.10E+00	**2.00E-07**	2.07E+00	**1.46E-05**	2.79E-01
1000	**3.66E-06**	4.64E+00	**1.59E-06**	4.55E+00	**6.14E-06**	6.33E-01
F7	**Avg.**		**Med.**		**Std.**	
	CMSSA	**SSA**	**CMSSA**	**SSA**	**CMSSA**	**SSA**
30	**1.03E-05**	9.14E-04	**7.26E-05**	5.83E-04	**6.76E-05**	9.76E-04
500	**6.37E-05**	2.58E-04	**6.93E-05**	2.33E-04	**4.69E-05**	2.12E-04
1000	**4.69E-05**	1.64E-04	**4.12E-05**	9.74E-05	**3.83E-05**	1.47E-04
F8	**Avg.**		**Med.**		**Std.**	
	CMSSA	**SSA**	**CMSSA**	**SSA**	**CMSSA**	**SSA**
30	**-7.53E+03**	-8.99E+03	**-7.18E+03**	-1.03E+04	**2.24E+03**	2.93E+03
500	**-1.13E+05**	-1.71E+05	**-1.24E+05**	-1.93E+05	**3.95E+04**	4.25E+04
1000	**-2.25E+05**	-3.17E+05	**-2.45E+05**	-3.46E+05	**8.59E+04**	9.43E+04
F9	**Avg.**		**Med.**		**Std.**	
	CMSSA	**SSA**	**CMSSA**	**SSA**	**CMSSA**	**SSA**
30	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00
500	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00
1000	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00
F10	**Avg.**		**Med.**		**Std.**	
	CMSSA	**SSA**	**CMSSA**	**SSA**	**CMSSA**	**SSA**
30	**8.88E-16**	8.88E-16	**8.88E-16**	8.88E-16	**0.00E+00**	0.00E+00
500	**8.88E-16**	8.88E-16	**8.88E-16**	8.88E-16	**0.00E+00**	0.00E+00
1000	**8.88E-16**	8.88E-16	**8.88E-16**	8.88E-16	**0.00E+00**	0.00E+00
F11	**Avg.**		**Med.**		**Std.**	
	CMSSA	**SSA**	**CMSSA**	**SSA**	**CMSSA**	**SSA**
30	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00
500	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00
1000	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00
F12	**Avg.**		**Med.**		**Std.**	
	CMSSA	**SSA**	**CMSSA**	**SSA**	**CMSSA**	**SSA**
30	**6.96E-09**	3.16E-03	**1.72E-09**	3.22E-03	**1.32E-08**	6.91E-04
500	**2.88E-09**	3.87E-03	**7.21E-10**	3.83E-03	**7.05E-09**	3.96E-04
1000	**4.06E-09**	3.76E-03	**1.21E-09**	3.67E-03	**1.04E-08**	5.43E-04
F13	**Avg.**		**Med.**		**Std.**	
	CMSSA	**SSA**	**CMSSA**	**SSA**	**CMSSA**	**SSA**
30	**7.13E-08**	2.44E-01	**7.48E-09**	1.53E-01	**1.67E-07**	2.46E-01
500	**1.25E-06**	1.23E+00	**4.07E-07**	1.23E+00	**2.05E-06**	3.25E-01
1000	**1.63E-06**	2.39E+00	**4.06E-07**	2.38E+00	**2.64E-06**	5.73E-01

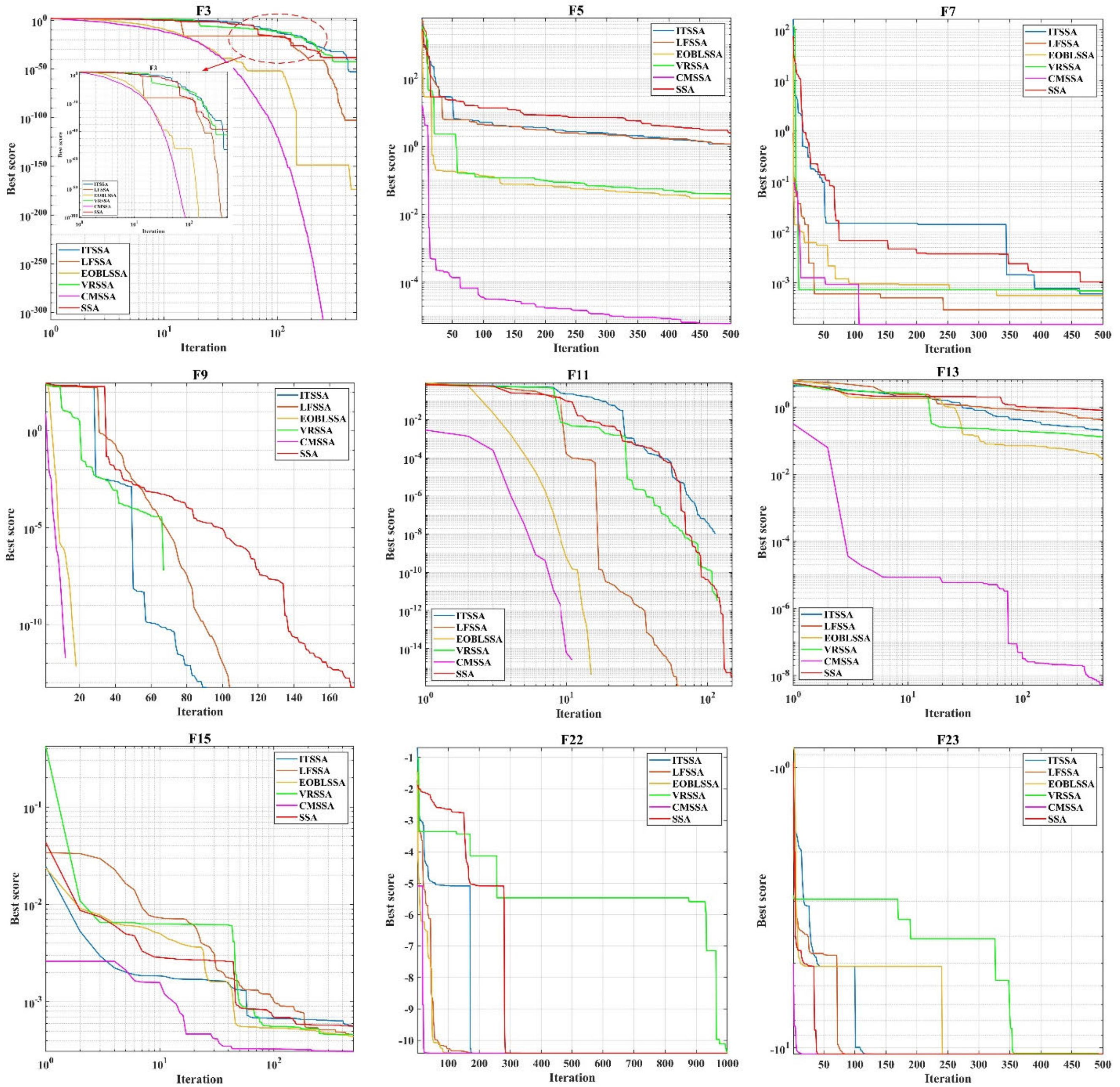

FIGURE 5. Convergence curves on 9 benchmark functions.

first and second Std. value on F_{17}. LFSSA and ITSSA attain the first and second Std. value on F_{18}. For F_{19}, F_{22}, and F_{23}, CMSSA obtains the best results compared to other competitors. For F_{20}, ITSSA provides the best Std. value, and LFSSA attains the best results of Avg. and Med. value. For F_{21}, the Avg. value obtained by LFSSA is better than other algorithms, and VRSSA attains the best Std. value. In addition, LFSSA, EOBLSSA, CMSSA and SSA obtain the same Med. value. For F_{24}, CMSSA obtains the best results compared to other algorithms. In terms of Avg. and Med. values, LFSSA can obtain better results than ITSSA, EOBLSSA, VRSSA and SSA. In terms of Std. value, ITSSA obtains the second-best results. For F_{25} and F_{26}, ITSSA and CMSSA provide better results than other algorithms in terms of Avg., Med. and Std. values. For F_{27}, CMSSA obtains the first-best results than other competitors in terms of Avg. and Med. values. In addition, ITSSA and CMSSA provide the first and second Std. value. For F_{28}, the Std. value obtained by LFSSA and SSA is better than other algorithms. For F_{28} and F_{30}, the Avg. and Med. values obtained by CMSSA are the best among these competitors. For F_{30}, SSA attains the best Std. value. For F_{29} and F_{31}, CMSSA provides the best results compared with other algorithms.

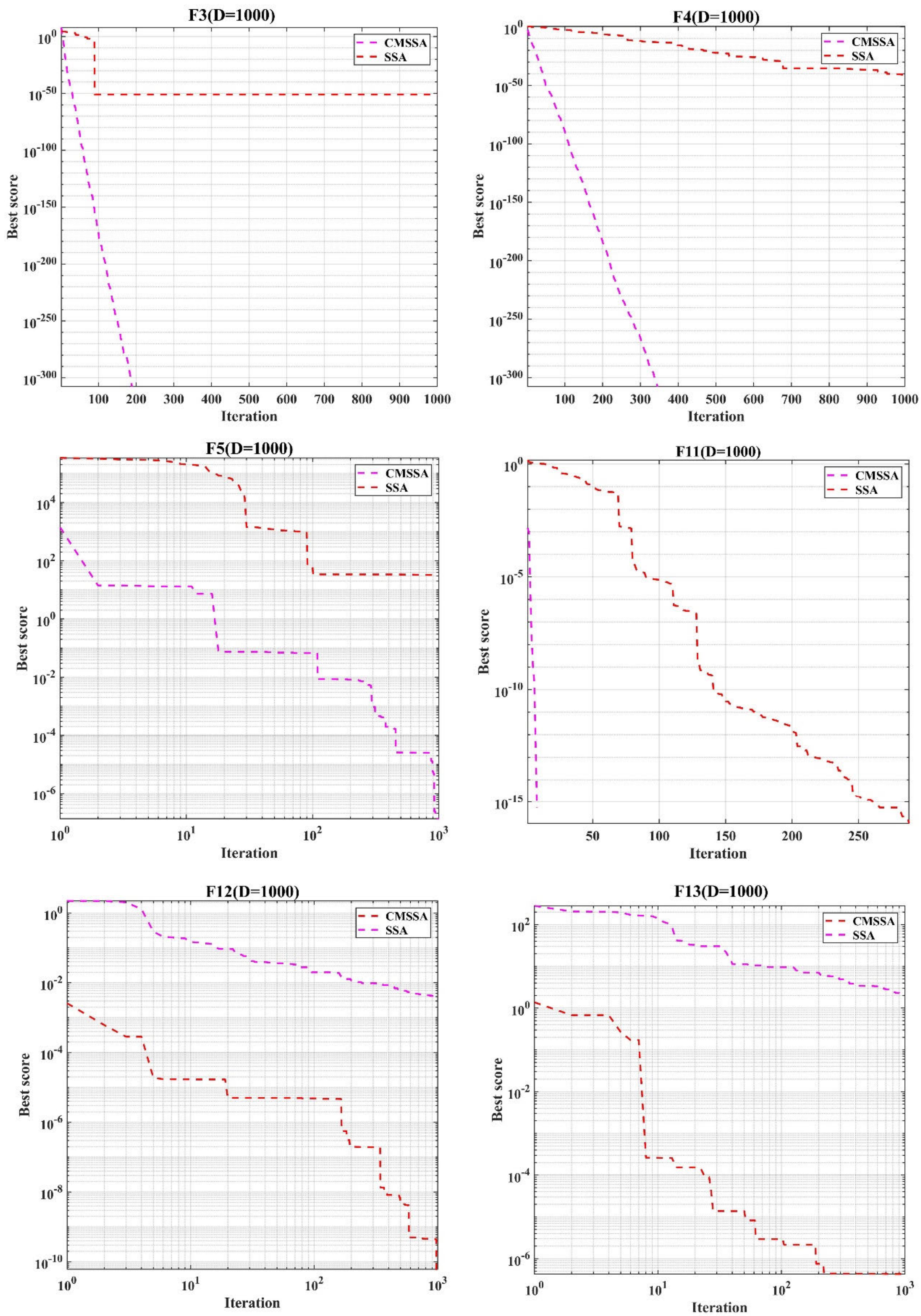

FIGURE 6. Convergence curves on 6 benchmark functions with D = 1000.

Therefore, the CMSSA algorithm is selected as the further research objective. In order to show the advantages of the CMSSA algorithm, some convergence curves of ITSSA, LFSSA, EOBLSSA, VRSSA, CMSSA, and SSA on some

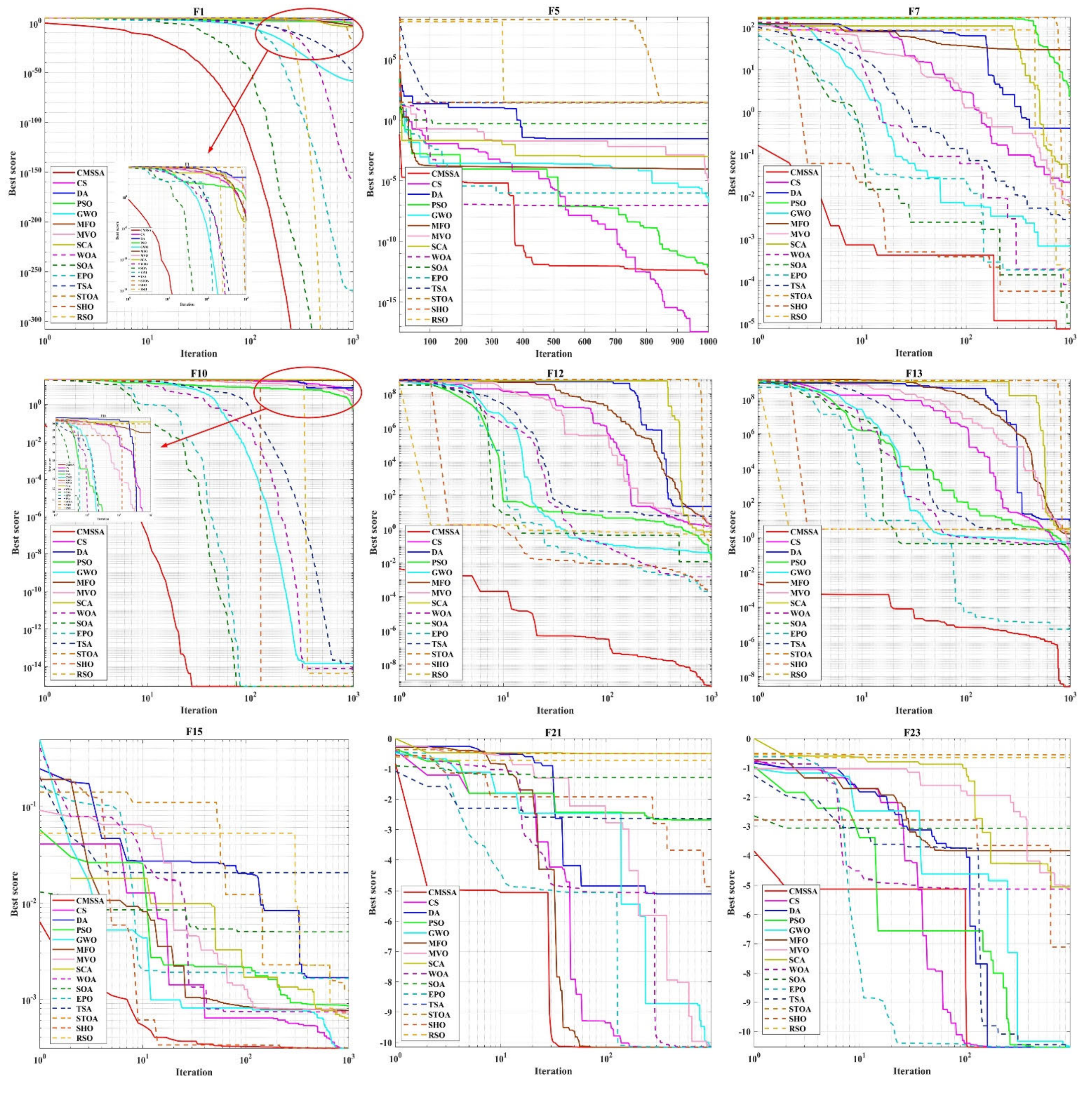

FIGURE 7. Convergence curves on 9 benchmark functions compared to other algorithms.

benchmark functions (i.e., F_3, F_5, F_7, F_9, F_{11}, F_{13}, F_{15}, F_{22}, and F_{23}) are given in Fig. 5. As it can be seen from Fig. 5, in terms of the convergence rate, the CMSSA algorithm ranks the first convergence rate for F_3, F_5, F_7, F_9, F_{11}, F_{13}, F_{15}, F_{22}, and F_{23}. However, the basic SSA algorithm ranks the worst for F_3, F_5, F_7, F_9, F_{11}, F_{13}, and F_{15}. For many benchmark functions, all the proposed SSA variants are superior to the basic SSA algorithm in terms of the convergence rate. This is due to the improvement of the convergence performance of SSA by various mutation operators. To sum up, these results show that the CMSSA algorithm is the best compared to other proposed variants.

In addition, the results of the Friedman test demonstrate that the CMSSA algorithm can obtain the smallest AGV. index, and then in terms of AVG. index, the CMSSA ranks first, followed by EOBLSSA, LFSSA, VRSSA, ITSSA, and the basic SSA. Therefore, it shows that the CMSSA algorithm is still the best method for handling 31 benchmark functions.

C. THE SCALABILITY TEST FOR CMSSA COMPARED TO BASIC SSA

In this section, to further compare the performance of CMSSA and the basic SSA more effectively and comprehensively, the scalability test is carried out. In this test, it focused on the different dimensions of benchmark functions affecting the performance of the CMSSA and basic SSA. Therefore, the dimensions on F_1-F_{13} are set to 30, 500, and 1000, respectively. The population size is set to 30 and the maximum number of iterations is set to 1000. The scalability results on 13 benchmark functions are shown in Table 16.

As the dimensions gradually increase, it becomes more challenging to obtain the global optimal solution for unimodal and multimodal benchmark functions. As it can be seen from Table 16, when the dimension is set to 30, 500, and 1000, SSA is better than CMSSA on F_8 and SSA is equal to CMSSA on F_9, F_{10},and F_{11}, respectively. However, CMSSA outperforms the basic SSA on other benchmark functions when the dimension is set to 500 and 1000, respectively. The convergence curves of some benchmark functions with the dimension of 1000 are described in Fig. 6. As it can be seen in Fig. 6, for different dimension problems, especially for the high dimension problems, the convergence rate of CMSSA is faster than that of the basic SSA, and the ability to jump out of local optimum in CMSSA is better than that of the basic SSA. This may be because the combination of improved Tent chaos, Lévy flights, elite opposition-based learning, and variable radius perturbation mutation operator can increase the population diversity and the search space, and strengthen the ability to jump out of the local optimum and the global search ability. All in all, it can be concluded that the CMSSA is superior to the basic SSA in the different dimensions, especially for the high dimension problems.

D. THE PERFORMANCE EVALUATION OF CMSSA

From the above results, it shows that the CMSSA outperforms other SSA variants. To evaluate the performance of the CMSSA in terms of six aspects (i.e., the exploitation capability, the exploration capability, the ability to escape from local minima, the convergence behavior, the statistical testing, and the wall-clock time cost), in this section, the CMSSA is compared to some state-of-the-art and advanced meta-heuristic algorithms. All the algorithms are carried out in the same experimental environment and all the experiments are carried out by 30 independent runs. The state-of-the-art and advanced meta-heuristic algorithms are as follows: PSO [7], CS [11], DA [16], GWO [12], WOA [17], MFO [15], SOA [23], SCA [43], MVO [44], SSA [30], EPO[21], STOA [24], TSA [29], SHO [19], RSO [31], TAPSO [112], MPSO [113], IPSO [114] and GWOCS [115], respectively. The parameter settings of these algorithms are shown in Table 17.

TABLE 17. The parameter setting values of mentioned algorithms.

Algorithms	Parameters	Values
For all algorithms	Population size	30
	Maximum number of iterarions	1000
CMSSA	Number of discovers *PD*	PD =0.2
	Number of sparrows *SD* who to be on guard	SD =0.1
	Safety factor *ST*	ST=0.8
PSO	Inertia Coefficient	w =0.75
	Cognitive and Social Coefficient	c_1=c_2=2
SSA	Number of discovers *PD*	PD =0.2
	Number of sparrows *SD* who to be on guard	SD =0.1
	Safety factor *ST*	ST=0.8.
CS	Discovery rate of alien eggs p_a	p_a = 0.25
	Lévy parameter β	β=1.5
DA	inertia weight	w=[0.9,0.2]
	separation weight	s = 0.1
	alignment weight	a = 0.1
	cohesion weight	c = 0.7
	food factor	f=1
	enemy factor	e =1
GWO	Control parameter a	a =[2,0]
MFO	Convergence Constant	[-1,-2]
	Logarithmic Spiral	0.75
MVO	Wormhole Existence Prob.	[0.2,1]
	Traveling Distance Rate	[0.6,1]
SCA	Number of Elites A	A=2
SOA	Control parameter A, f_c	A=[2,0], f_c=2
WOA	Control parameter a	a=[2,0]
EPO	Temperature profile T	[1,1000]
	Control parameter	A=[-1.5,1.5], f=[2,3],M=2 l=[1.5,2] $S()$=[0,1.5]
SHO	Control parameter (h)	h=[5,0]
	M constant	M=[0.5,1]
TSA	Parameter P	$Pmin$=1, $Pmax$=4
STOA	Control parameter A, f_c	A=[2,0], f_c=2
RSO	Control parameter (R)	[1,5]
	Constant Parameter C	[0,2]
TAPSO	Inertia weight factor w	w=[0.9, 0.4]
	Cognitive and Social Coeff (c_1, c_2)	[2.5, 0.5]
MPSO	Inertia weight factor w	w=[0.9,0.4]
	Cognitive and Social Coeff (c_1, c_2)	[2.5, 0.5]
IPSO	Inertia weight factor w	w=[0.9,0.4]
	Cognitive and Social Coeff (c_1, c_2)	[2.5, 0.5]
GWOCS	Control parameter a	a=[2,0]
	Lévy parameter β	β=1.5

1) ANALYSIS OF EXPLOITATION CAPABILITY (FUNCTIONS F_1-F_7)

The unimodal benchmark functions (i.e., functions F_1-F_7) have only one global optimal solution in the search space. These functions are usually used to evaluate the exploitation capability of meta-heuristic algorithms. In this experiment, the exploitation capability of the CMSSA was tested by unimodal benchmark functions compared to PSO, CS, DA, GWO, WOA, MFO, SOA, SCA, MVO, EPO, TSA, STOA, SHO, and RSO, respectively. The experimental results were shown in Table 18. As it can be seen from Table 18, for F_1, F_2, F_3, F_4, CMSSA can obtain the global optimum. For F_1, in terms of Avg., Med. and Std. values, CMSSA, SOA, SHO, and RSO can provide the global optimum. In addition, EPO and WOA obtain the second and third best results. For F_2, CMSSA and SHO attain the best results. SOA and RSO obtain second and third best results, which are better than the rest of other algorithms. For F_3, CMSSA, SHO, and RSO attain the best results. The results obtained by GWO and TSA are superior to PSO, CS, DA, MFO, SCA, SOA, STOA, MVO, EPO, and WOA algorithms. For F_4, the results obtained by CMSSA and SHO are better than other algorithms, and RSO and SOA provide the third and fourth best results. For F_5, CS is the best optimizer compared to other algorithms. However, CMSSA can obtain the competitive results which are better than DA, GWO, MFO, MVO, SCA, SOA, EPO, TSA, STOA, SHO, RSO, and WOA algorithms. For F_6, CMSSA obtains the best results compared to other competitors. CS and EPO provide the second and third best results which are superior to PSO, DA, GWO, MFO, MVO, SCA, WOA, STOA, SHO, TSA, RSO, and SOA algorithms. For F_7, CMSSA is inferior to SOA in terms of Avg. and Med. values, but the Std. value of CMSSA is superior to SOA and the results obtained from CMSSA are superior to PSO, CS, DA, GWO, MFO, MVO, SCA, WOA, EPO, TSA, STOA, SHO, and RSO algorithms.

These results show that CMSSA can obtain the best solution or at least the second-best solution on all unimodal benchmark functions compared with other meta-heuristic algorithms. Therefore, CMSSA is significantly competitive and has a very good exploitation capability.

2) ANALYSIS OF EXPLORATION CAPABILITY (FUNCTIONS F_8-F_{23})

Compared with unimodal functions, multimodal functions (i.e., functions F_8-F_{23}) often have multiple local optima, so it is difficult to obtain the global optimum. Therefore, in this experiment, these functions were tested to evaluate the exploration capability of CMSSA compared to PSO, CS, DA, GWO, WOA, MFO, SOA, SCA MVO, EPO, TSA, STOA, SHO, and RSO, respectively. The experimental results are shown in Table 18. For F_9, F_{11}, F_{15}, F_{16}, F_{17}, F_{18}, F_{19}, F_{21}, and F_{23}, CMSSA can attain the global optimum in terms of Avg. and Med. values. For F_8, SOA provides the best results. Whereas, the results obtained by CMSSA are inferior to WOA, SOA, MVO, MFO, EPO and CS, and are superior to PSO, DA, GWO, TSA, STOA, SHO, RSO, and SCA in terms of Avg., Std. and Med. values. For F_9, CMSSA, WOA, SOA, EPO, SHO, and RSO can obtain the global optimum in terms of Avg., Med., and Std. values. The results produced by GWO are better than the rest of other algorithms. For F_{10}, the results obtained by CMSSA, SHO and SOA are the best compared to other approaches. RSO and EPO attain the second and third best results. For F_{11}, the results obtained from CMSSA, SOA, SHO, and RSO are the same which are better than other algorithms. For F_{12}, the results produced by CMSSA are superior to other optimizers. SHO and PSO are the second and third best optimizers.

For F_{13}, CMSSA is the best optimizer. In addition, EPO and MVO obtain the second and third best results in terms of Avg., Med. and Std. values. For F_{14}, CS is the best optimizer. The results obtained by CMSSA are superior to PSO, GWO, MFO, TSA, RSO, and SOA. For F_{15}, CMSSA provides the best Avg. value. CS obtains the best Std. value. In addition, GWO and CS attain the best Med. value. CMSSA can provide competitive results which are better than PSO, DA, MFO, MVO, SCA, SOA, EPO, TSA, STOA, SHO, RSO, and WOA. For F_{16}, CMSSA is the first best optimizer. CS and MFO are the third-best optimizers. For F_{17}, PSO, CS, and MFO are the best optimizers. CMSSA is the second-best optimizer. For F_{18}, the results obtained by MFO are the best. CMSSA obtains better results than DA, GWO, MVO, SCA, SOA, EPO, TSA, STOA, SHO, RSO, and WOA. For F_{19}, the results produced by CMSSA are better than DA, GWO, SCA, SOA, EPO, TSA, STOA, SHO, RSO, and WOA. MFO obtains the best results. For F_{20}, CS is the best optimization approach. CMSSA can obtain better results than SCA, EPO, STOA, SHO, RSO, and SOA. For F_{21}, CMSSA provides the best results compared to other competitors. For F_{22}, CS provides the best results compared to other competitors. The results obtained by CMSSA are better than PSO, DA, MFO, MVO, SCA, SOA WOA, EPO, TSA, STOA, SHO, and RSO. For F_{23}, CS is the best optimizer. CMSAA can obtain better results than other algorithms except for CS.

The results show that CMSSA is better than other competitors for many multimodal functions so that CMSSA has a very good exploration capability. This is due to the mixing of various mutation operators to improve the ability to search for the global optimum in CMSSA.

3) ABILITY TO ESCAPE FROM LOCAL MINIMA (FUNCTIONS F_{24}-F_{31})

To better evaluate the ability to escape from local minima, some hybrid and composition benchmark functions are selected to test the ability to escape from local minima. Four well-known advanced algorithms were added, namely, TAPSO, MPSO, IPSO, and GWOCS. In this experiment, the performance of CMSSA is evaluated compared with SSA, SCA, WOA, TAPSO, MPSO, IPSO, and GWOCS. The experimental results are reported in Table 19. For F_{24} and F_{25}, in terms of Avg., Std. and Med. values, CMSSA provides the best results compared to other algorithms. For F_{26}, TAPSO obtains the best Avg. and Std. values, and GWOCS provides the best Med. value. The results obtained by CMSSA are better than WOA and SSA in terms of Avg., Std. and Med. values, For F_{27}, CMSSA provides the best Std. value. MPSO and TASO can obtain better results than others in terms of Avg. and Med. values. The results produced by CMSSA are superior to WAO, IPSO, SSA, and GWOCS. For F_{28}, CMSSA obtains better results with the second Std. value than other algorithms. For F_{29}, CMSSA is the best optimizer. For F_{30}, GWOCS is the best optimizer in terms of Avg. and Med. values. The results obtained by CMSSA are superior to WOA, TAPSO, IPSO, and SSA. For F_{31}, CMSSA can provide better results than others in terms of Avg. and Std. values. The Med. value of CMSSA is the second-best result compared to other algorithms.

The results demonstrate that CMSSA is better than other algorithms for many functions. Therefore, it shows that the CMSSA algorithm still has a very good ability to escape from local minima. This may be due to the sparrow individual

TABLE 18. The experimental results of 23 benchmark functions.

Algorithm	F1			F2			F3		
	Avg.	Med.	Std.	Avg.	Med.	Std.	Avg.	Med.	Std.
CMSSA	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
PSO	2.884E-01	2.599E-01	2.291E-01	9.712E-01	9.060E-01	4.287E-01	7.742E+01	7.301E+01	1.801E+01
CS	3.209E-03	2.661E-03	2.159E-03	3.312E-01	2.974E-01	1.705E-01	2.738E+02	2.416E+02	1.198E+02
DA	1.149E+03	1.094E+03	5.194E+02	1.488E+01	1.449E+01	5.482E+00	1.353E+04	1.312E+04	8.457E+03
GWO	4.768E-59	1.155E-59	1.023E-58	8.626E-35	5.882E-35	1.008E-34	1.099E-15	4.598E-17	2.750E-15
MFO	6.186E+00	1.000E-04	2.253E+01	3.115E+01	3.017E+01	1.537E+01	1.807E+04	1.557E+04	1.487E+04
MVO	2.809E-01	2.598E-01	8.472E-02	3.806E-01	3.541E-01	1.086E-01	4.871E+01	4.293E+01	2.347E+01
SCA	4.022E-02	1.487E-03	1.088E-01	1.786E-05	4.243E-06	4.495E-05	2.792E+03	1.623E+03	3.041E+03
SOA	0.000	0.000	0.000	4.000E-323	4.400E-323	0.000	1.552E+04	1.218E+04	1.175E+04
WOA	3.284E-152	3.887E-159	1.788E-151	1.182E-104	3.238E-108	4.116E-104	2.073E+04	1.946E+03	1.121E+03
EPO	1.292E-230	4.211E-297	0.000	1.312E-175	7.168E-188	0.000	1.713E+02	6.330E+01	2.235E+02
TSA	1.143E-47	2.285E-48	2.680E-47	7.742E-29	3.872E-29	8.120E-29	1.955E-11	9.271E-15	5.716E-11
STOA	1.233E-17	1.243E-18	2.405E-17	2.364E-12	1.058E-12	3.158E-12	4.658E-08	1.196E-08	8.687E-08
SHO	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RSO	0.000	0.000	0.000	3.067E-275	0.000	0.000	0.000	0.000	0.000

Algorithm	F4			F5			F6		
	Avg.	Med.	Std.	Avg.	Med.	Std.	Avg.	Med.	Std.
CMSSA	**0.000**	**0.000**	**0.000**	**1.483E-08**	**4.714E-11**	**4.069E-08**	**2.175E-08**	**8.556E-09**	**3.295E-08**
PSO	1.553E+00	1.573E+00	2.336E-01	6.251E-12	1.086E-13	2.968E-11	2.904E-01	2.349E-01	1.557E-01
CS	5.268E+00	5.045E+00	1.959E+00	7.459E-16	1.278E-19	3.428E-15	3.081E-03	2.926E-03	1.325E-03
DA	2.197E+01	2.011E+01	6.149E+00	3.816E-02	3.577E-03	1.341E-01	9.741E+02	9.899E+02	5.801E+02
GWO	2.932E-14	7.299E-15	5.426E-14	3.021E-07	1.603E-07	4.813E-07	7.260E-01	7.482E-01	3.145E-01
MFO	6.488E+01	6.416E+01	9.343E+00	2.186E-01	1.647E-02	8.259E-01	3.459E+02	1.857E-04	1.842E+03
MVO	9.611E-01	8.861E-01	4.644E-01	1.443E-05	1.055E-05	1.243E-05	3.007E-01	2.976E-01	9.283E-02
SCA	2.023E+01	1.949E+01	1.173E+01	5.881E-04	4.427E-04	5.163E-04	4.505E+00	4.530E+00	4.556E-01
SOA	4.942E-36	1.058E-70	2.707E-35	1.651E-01	5.352E-02	2.301E-01	2.053E+00	1.695E+00	1.200E+00
WOA	3.771E+01	2.648E+01	3.012E+01	8.151E-07	8.407E-08	3.158E-06	8.737E-02	3.694E-02	8.688E-02
EPO	1.949E-08	6.285E-17	7.016E-08	3.731E-05	1.046E-05	6.610E-05	1.532E-02	6.138E-03	2.301E-02
TSA	6.365E-03	3.686E-03	7.855E-03	2.820E+01	2.865E+01	9.664E-01	3.634E+00	3.575E+00	6.190E-01
STOA	1.453E-05	7.568E-06	2.217E-05	2.789E+01	2.800E+01	6.572 E-01	2.474E+00	2.499E+00	4.714E-01
SHO	0.000	0.000	0.000	2.878E+01	2.871E+01	1.098E-01	2.751E+00	1.843E+00	2.953E+00
RSO	1.668E-237	0.000	0.000	2.865E+01	2.875E+01	3.346E-01	3.328E+00	3.318E+00	5.954E-01

Algorithm	F7			F8			F9		
	Avg.	Med.	Std.	Avg.	Med.	Std.	Avg.	Med.	Std.
CMSSA	**9.180E-05**	**8.187E-05**	**6.030E-05**	**-7.274E+03**	**-6.818E+03**	**2.466E+03**	**0.000**	**0.000**	**0.000**
PSO	2.197E+00	2.106E+00	1.255E+00	-6.905E+03	-6.745E+03	9.024E+02	1.114E+02	1.036E+02	3.1402E+01
CS	4.182E-02	3.911E-02	1.452E-02	-8.652E+03	-8.650E+03	2.363E+02	7.543E+01	7.418E+01	1.156E+01
DA	2.986E-01	2.798E-01	1.670E-01	-5.589E+03	-5.542E+03	6.416E+02	1.579E+02	1.597E+02	1.964E+01
GWO	7.852E-04	7.428E-04	4.061E-04	-6.050E+03	-6.104E+03	7.628E+02	8.618E-01	0.000	2.051E+00
MFO	1.735E+00	1.621E-01	2.769E+00	-9.112E+03	-9.205E+03	6.868E+02	1.555E+02	1.517E+02	3.674E+01
MVO	1.856E-02	1.705E-02	6.884E-03	-7.946E+03	-7.912E+03	6.810E+02	1.11.7E+02	1.097E+02	2.650E+01
SCA	2.239E-02	1.708E-02	1.771E-02	-3.967E+03	-3.997E+03	3.522E+02	1.296E+01	5.013E+00	1.774E+01
SOA	7.772E-05	6.577E-05	8.699E-05	-1.257E+04	-1.257E+04	9.703E-01	0.000	0.000	0.000
WOA	1.670E-03	5.720E-04	2.133E-03	-1.129E+04	-1.219E+04	1.756E+03	0.000	0.000	0.000
EPO	1.621E-04	1.121E-04	1.530E-04	-1.241E+04	-1.249E+04	1.777E+02	0.000	0.000	0.000
TSA	5.572E-03	5.359E-03	2.481E-03	-6.242E+03	-6.263E+03	5.404E+02	1.913E+02	1.791E+02	4.770E+01
STOA	2.523E-03	2.088E-03	1.783E-03	-5.490E+03	-5.357E+03	6.714E+02	2.609E+00	2.273E-13	6.729E+00
SHO	3.407E-04	2.638E-05	1.526E-03	-2.659E+03	-2.718E+03	5.062E+02	0.000	0.000	0.000
RSO	2.371E-04	1.454E-04	2.260E-04	-5.930E+03	-6.167E+03	1.164E+03	0.000	0.000	0.000

Algorithm	F10			F11			F12		
	Avg.	Med.	Std.	Avg.	Med.	Std.	Avg.	Med.	Std.
CMSSA	**8.88E-16**	**8.88E-16**	**0.000**	**0.000**	**0.000**	**0.000**	**2.088E-09**	**2.485E-10**	**3.678E-09**
PSO	1.004E+00	9.655E-01	5.384E-01	1.833E-02	1.744E-02	8.893E-03	2.643E-03	1.533E-03	3.879E-03
CS	2.601E+00	2.564E+00	9.038E-01	6.981E-02	6.130E-02	4.519E-02	1.497E+00	1.485E+00	6.156E-01
DA	8.434E+00	8.349E+00	1.589E+00	1.321E+01	1.386E+01	6.205E+00	4.369E+02	2.259E+01	2.023E+03
GWO	1.628E-14	1.510E-14	3.771E-15	1.933E-03	0.000	5.059E-03	3.880E-02	3.293E-02	2.543E-02
MFO	1.344E+01	1.799E+01	8.053E+00	4.679E-01	1.779E-02	1.173E+00	6.839E-01	1.101E-01	1.218E+00

TABLE 18. *(Continued.)* The experimental results of 23 benchmark functions.

MVO	8.995E-01	7.413E-01	6.675E-01	6.075E-01	6.067E-01	9.172E-02	1.465E+00	1.296E+00	8.380E-01
SCA	1.158E+01	1.860E+01	9.483E+00	0.375722	3.901E-01	2.410E-01	4.645E+00	1.463E+00	1.116E+01
SOA	8.88E-16	8.88E-16	0.000	0.000	0.000	0.000	1.897E-01	1.231E-01	1.925E-01
WOA	3.375E-15	4.441E-15	1.901E-15	1.685E-03	0.000	9.231E-03	5.470E-03	2.944E-03	4.593E-03
EPO	2.191E-15	8.88E-16	1.975E-15	4.903E-01	1.648E-01	5.442E-01	3.942E-03	4.797E-05	1.892E-02
TSA	1.883E+00	2.834E+00	1.584E+00	7.032E-03	9.126E-03	7.724E-03	6.326E+00	6.373E+00	3.378E+00
STOA	1.996E+01	1.995E+01	1.603E-03	5.906E-03	2.220E-16	1.179E-02	2.354E-01	2.006E-01	1.354E-01
SHO	8.88E-16	8.88E-16	0.000	0.000	0.000	0.000	2.032E-04	2.055E-04	2.041E-05
RSO	1.953E-15	8.88E-16	1.656E-15	0.000	0.000	0.000	3.777E-01	3.691E-01	1.317E-01

Algorithm	F13			F14			F15		
	Avg.	Med.	Std.	Avg.	Med.	Std.	Avg.	Med.	Std.
CMSSA	**6.256E-08**	**1.242E-08**	**1.017E-07**	**2.663E+00**	**1.992E+00**	**2.430E+00**	**3.095E-04**	**3.080E-04**	**2.823E-06**
PSO	8.173E-02	6.786E-02	6.091E-02	3.462E+00	1.992E+00	2.632E+00	8.621E-04	8.533E-04	1.457E-04
CS	1.506E-01	6.151E-02	3.489E-01	9.98E-01	9.98E-01	0.000	3.101E-04	3.075E-04	3.927E-08
DA	1.551E+04	4.200E+03	2.470E+04	9.98E-01	9.98E-01	8.057E-10	2.552E-03	1.489E-03	3.926E-03
GWO	5.779E-01	6.159E-01	1.884E-01	3.281E+00	9.98E-01	3.857E+00	3.650E-03	3.075E-04	7.602E-03
MFO	6.857E+00	4.243E-02	3.054E+01	2.803E+00	1.495E+00	2.666E+00	2.132E-03	7.647E-04	4.960E-03
MVO	7.459E-02	6.077E-02	4.302E-02	9.98E-01	9.98E-01	8.212E-12	7.177E-03	7.212E-04	1.255E-02
SCA	6.186E+03	3.566E+00	3.117E+04	1.330E+00	9.981E-01	7.513E-01	9.240E-04	7.798E-04	3.593E-04
SOA	1.073E+00	9.865E-01	6.017E-01	6.364E+00	2.992E+00	4.969E+00	2.314E-03	1.813E-03	2.184E-03
WOA	2.687E-01	1.979E-01	2.016E-01	2.375E+00	9.98E-01	2.563E+00	6.438E-04	5.638E-04	3.397E-04
EPO	1.321E-05	9.382E-06	1.431E-05	1.429E+00	9.980E-01	9.272E-01	1.301E-03	1.192E-03	8.880E-04
TSA	2.936E+00	2.807E+00	5.583E-01	9.897E+00	1.219E+01	4.189E+00	1.211E-02	1.002E-03	2.258E-02
STOA	1.652E+00	1.646E+00	1.892E-01	1.952E+00	9.980E-01	1.898E+00	1.043E-03	1.224E-03	3.470E-04
SHO	2.945E+00	2.944E+00	2.475E-02	1.040E+01	1.155E+01	2.898E+00	3.136E-04	3.127E-04	3.978E-06
RSO	2.843E+00	2.844E+00	4.924E-02	2.839E+00	2.982E+00	2.347E+00	9.520E-04	8.879E-04	3.360E-04

Algorithm	F16			F17			F18		
	Avg.	Med.	Std.	Avg.	Med.	Std.	Avg.	Med.	Std.
CMSSA	**-1.0316**	**-1.0316**	**1.828E-16**	**3.979E-01**	**3.979E-01**	**4.719E-15**	**3.000**	**3.000**	**4.967E-15**
PSO	-1.0316	-1.0316	6.253E-16	3.979E-01	3.979E-01	0.000	3.000	3.000	2.057E-15
CS	-1.0316	-1.0316	6.775E-16	3.979E-01	3.979E-01	0.000	3.000	3.000	1.860E-15
DA	-1.0316	-1.0316	5.735E-06	3.979E-01	3.979E-01	7.015E-06	3.000	3.000	5.240E-05
GWO	-1.0316	-1.0316	3.556E-09	3.979E-01	3.979E-01	6.863E-07	3.000	3.000	1.580E-05
MFO	-1.0316	-1.0316	6.775E-16	3.979E-01	3.979E-01	0.000	3.000	3.000	1.272E-15
MVO	-1.0316	-1.0316	1.273E-07	3.979E-01	3.979E-01	2.080E-07	3.000	3.000	8.309E-07
SCA	-1.0316	-1.0316	2.426E-05	3.989E-01	3.984E-01	1.419E-03	3.000	3.000	3.304E-05
SOA	-1.0262	-1.0316	1.499E-02	3.998E-01	3.980E-01	5.732E-03	1.048E+01	3.000	1.264E+01
WOA	-1.0316	-1.0316	4.779E-11	3.979E-01	3.979E-01	6.473E-07	3.000	3.000	5.414E-05
EPO	-1.0316	-1.0316	5.331E-07	3.979E-01	3.979E-01	1.601E-05	9.300E+00	3.000	1.161E+01
TSA	-1.027E+00	-1.0316	1.093E-02	3.979E-01	3.979E-01	4.301E-05	1.226E+01	3.000	2.261E+01
STOA	-1.0316	-1.0316	3.662E-07	3.979E-01	3.979E-01	8.241E-05	3.000	3.000	1.747E-05
SHO	-9.545E-01	-9.970E-01	1.436E-01	6.579E-01	3.983E-01	9.491E-01	1.805E+01	8.610E+00	1.987E+01
RSO	-1.0315	-1.0316	9.512E-05	1.172E+00	6.361E-01	1.205E+00	3.000	3.000	3.792E-06

Algorithm	F19			F20			F21		
	Avg.	Med.	Std.	Avg.	Med.	Std.	Avg.	Med.	Std.
CMSSA	**-3.8628E+00**	**-3.8628E+00**	**9.136E-10**	**-3.187E+00**	**-3.190E+00**	**1.358E-02**	**-10.1532E+00**	**-10.1532E+00**	**1.698E-08**
PSO	-3.8628E+00	-3.8628E+00	2.597E-15	-3.258E+00	-3.203E+00	6.032E-02	-7.764E+00	-10.1532E+00	3.035E+00
CS	-3.8628E+00	-3.8628E+00	2.710E-15	-3.322E+00	-3.322E+00	5.232E-12	-10.1532E+00	-10.1532E+00	5.205E-15
DA	-3.8625E+00	-3.8628E+00	1.006E-03	-3.263E+00	-3.320E+00	7.397E-02	-7.7009E+00	-10.0988E+00	2.691E+00
GWO	-3.8612E+00	-3.8628E+00	2.871E-03	-3.231E+00	-3.202E+00	7.623E-02	-9.8129E+00	-10.1528E+00	1.293E+00
MFO	-3.8628E+00	-3.8628E+00	2.307E-15	-3.235E+00	-3.203E+00	8.065E-02	-7.8070E+00	-10.1532E+00	3.223E+00
MVO	-3.8628E+00	-3.8628E+00	5.047E-07	-3.250E+00	-3.203E+00	5.952E-02	-7.793E+00	-10.1532E+00	2.808E+00
SCA	-3.8548E+00	-3.8543E+00	2.066E-03	-2.850E+00	-3.016E+00	4.197E-01	-2.1980E+00	-0.8809E+00	1.805E+00
SOA	-3.7355E+00	-3.8234E+00	1.973E-01	-2.723E+00	-2.821E+00	4.183E-01	-4.1955E+00	-3.94962E+00	2.480E+00
WOA	-3.8606E+00	-3.8618E+00	2.500E-03	-3.224E+00	-3.256E+00	1.157E-01	-9.0505E+00	-10.1501E+00	2.274E+00
EPO	-3.8054E+00	-3.8627E+00	1.489E-01	-3.059E+00	-3.085E+00	1.673E-01	-8.5896E+00	-10.1527E+00	3.555E+00
TSA	-3.8627E+00	-3.8627E+00	5.574E-05	-3.234E+00	-3.201E+00	1.061E-01	-6.0337E+00	-5.0526E+00	3.186E+00
STOA	-3.8550E+00	-3.8548E+00	1.115E-03	-3.056E+00	-3.015E+00	5.910E-02	-4.6755E+00	-1.7535E+00	4.627E+00
SHO	-3.6705E+00	-3.7794E+00	2.334E-01	-2.731E+00	-2.801E+00	2.751E-01	-3.7282E+00	-3.301E+00	1.131E+00
RSO	-3.3208E+00	-3.3449E+00	3.635E-01	-1.718E+00	-1.610E+00	4.542E-01	-7.971E-01	-4.811E-01	6.595E-01

TABLE 18. ***(Continued.)*** **The experimental results of 23 benchmark functions.**

Algorithm	F22			F23		
	Avg.	Med.	Std.	Avg.	Med.	Std.
CMSSA	**-9.871E+00**	**-10.4029E+00**	**1.621E+00**	**-10.5364E+00**	**-10.5364E+00**	**1.765E-07**
PSO	-9.871E+00	-10.4029E+00	1.622E+00	-9.9069E+00	-10.5364E+00	1.964E+00
CS	-10.4029E+00	-10.4029E+00	1.261E-13	-10.5364E+00	-10.5364E+00	3.46E-12
DA	-7.5061E+00	-5.1288E+00	2.784E+00	-7.63947E+00	-10.2007E+00	3.217E+00
GWO	-10.2253E+00	-10.4025E+00	9.703E-01	-10.3557E+00	-10.536E+00	9.872E-01
MFO	-7.8537E+00	-10.4029E+00	3.450E+00	-7.3519E+00	-10.5364E+00	3.759E+00
MVO	-7.7194E+00	-10.4027E+00	3.213E+00	-9.9986E+00	-10.5363E+00	1.640E+00
SCA	-3.7914E+00	-4.5946E+00	2.385E+00	-4.43011E+00	-4.81107E+00	1.740E+00
SOA	-3.8804E+00	-3.4837E+00	2.362E+00	-3.6928E+00	-3.6161E+00	1.944E+00
WOA	-8.5588E+00	-10.3973E+00	2.907E+00	-8.7509E+00	-10.5336E+00	3.036E+00
EPO	-9.7688E+00	-10.4020E+00	2.408E+00	-10.2160E+00	-10.5359E+00	1.750E+00
TSA	-7.1911E+00	-10.0383E+00	3.558E+00	-7.4757E+00	-10.2902E+00	3.735E+00
STOA	-4.8795E+00	-4.4040E+00	4.519E+00	-6.6598E+00	-5.1282E+00	3.933E+00
SHO	-3.8892E+00	-4.0835E+00	9.946E-01	-4.5665E+00	-4.6462E+00	1.179E+00
RSO	-1.1171E+00	-7.9170E-01	9.022E-01	-1.3317E+00	-9.792E-01	7.284E-01

TABLE 19. **The experimental results of 8 hybrid and composition benchmark functions.**

Algorithm	F24			F25			F26		
	Avg.	**Med.**	**Std.**	**Avg.**	**Med.**	**Std.**	**Avg.**	**Med.**	**Std.**
CMSSA	**1.740E+03**	**1.741E+03**	**1.059E+01**	**5.305E+03**	**3.674E+03**	**5.215E+03**	**1.118E+04**	**7.344E+03**	**9.938E+03**
SCA	1.784E+03	1.785E+03	1.802E+01	1.782E+05	1.298E+05	1.651E+05	7.519E+03	4.251E+03	5.473E+03
WOA	1.805E+03	1.794E+03	5.222E+01	1.642E+04	1.270E+04	1.032E+04	6.956E+04	2.055E+04	1.328E+05
TAPSO	1.747E+03	1.743E+03	2.181E+01	1.217E+04	9.771E+03	9.814E+03	4.208E+03	2.924E+03	2.678E+03
MPSO	1.763E+03	1.751E+03	4.822E+01	1.158E+04	7.373E+03	1.026E+04	4.526E+03	2.815E+03	3.957E+03
IPSO	1.760E+03	1.757E+03	2.642E+01	9.687E+03	6.398E+03	8.784E+03	5.082E+03	3.996E+03	3.724E+03
SSA	1.974E+03	1.991E+03	1.214E+02	2.037E+04	1.188E+04	3.986E+03	2.795E+06	3.780E+05	4.433E+06
GWOCS	1.777E+03	1.786E+03	2.762E+01	1.143E+04	1.011E+04	6.222E+03	4.885E+03	2.062E+03	4.514E+03
Algorithm	F27			F28			F29		
	Avg.	**Med.**	**Std.**	**Avg.**	**Med.**	**Std.**	**Avg.**	**Med.**	**Std.**
CMSSA	**2.109E+03**	**2.109E+03**	**2.406E+01**	**2.244E+03**	**2.227E+03**	**3.880E+01**	**2.641E+03**	**2.643E+03**	**6.681E+00**
SCA	2.103E+03	2.104E+03	2.566E+01	2.279E+03	2.280E+03	6.785E+01	2.657E+03	2.656E+03	1.060E+01
WOA	2.195E+03	2.210E+03	7.679E+01	2.320E+03	2.346E+03	6.053E+01	2.656E+03	2.653E+03	2.258E+01
TAPSO	2.102E+03	2.091E+03	5.274E+01	2.312E+03	2.337E+03	5.846E+01	2.671E+03	2.665E+03	3.038E+01
MPSO	2.098E+03	2.093E+03	5.094E+01	2.313E+03	2.335E+03	5.844E+01	2.683E+03	2.683E+03	3.277E+01
IPSO	2.112E+03	2.098E+03	6.434E+01	2.307E+03	2.327E+03	6.194E+01	2.685E+03	2.681E+03	2.244E+01
SSA	2.236E+03	2.245E+03	7.316E+01	2.375E+03	2.382E+03	3.514E+01	2.723E+03	2.713E+03	4.345E+01
GWOCS	2.116E+03	2.144E+03	5.253E+01	2.320E+03	2.315E+03	1.175E+01	2.623E+03	2.622E+03	1.281E+01
Algorithm	F30			F31					
	Avg.	**Med.**	**Std.**	**Avg.**	**Med.**	**Std.**			
CMSSA	**2.748E+03**	**2.767E+03**	**4.743E+01**	**3.104E+03**	**3.104E+03**	**2.925E+00**			
SCA	2.784E+03	2.791E+03	3.863E+01	3.107E+03	3.105E+03	1.318E+01			
WOA	2.779E+03	2.787E+03	5.507E+01	3.138E+03	3.121E+03	39.13E+01			
TAPSO	2.749E+03	2.781E+03	1.018E+02	3.151E+03	3.140E+03	48.12E+01			
MPSO	2.739E+03	2.803E+03	1.533E+02	3.184E+03	3.194E+03	4.933E+01			
IPSO	2.750E+03	2.787E+03	1.006E+02	3.176E+03	3.185E+03	4.472E+01			
SSA	2.874E+03	2.878E+03	6.476E+01	3.277E+03	3.261E+03	8.316E+01			
GWOCS	2.637E+03	2.724E+03	1.215E+02	3.112E+03	3.101E+03	2.746E+01			

mutated by the introduction of a variety of mutation operators to the ability to escape from local minima in CMSSA.

4) ANALYSIS OF CONVERGENCE BEHAVIOR

In order to better evaluate the convergence rate of the CMSSA algorithm, some convergence curves of PSO, CS, DA, GWO, WOA, MFO, SOA, SCA, MVO, EPO, TSA, STOA, SHO, RSO, and CMSSA on 9 benchmark functions were provided in Fig. 7, and the convergence curves of SCA, WOA, TAPSO, MPSO, IPSO, SSA GWOCS and CMSSA on 6 hybrid and composition benchmark functions were provided in Fig. 8. As it can be seen from Figs. 7 and 8, the CMSSA has a faster

TABLE 20. Wall-clock time cost for 23 benchmark functions.

Function	CMSSA	SSA	PSO	CS	DA	GWO	MFO	MVO	SCA	SOA	WOA	EPO	TSA	STOA	SHO	RSO
F1	**0.8569**	0.6568	0.0997	0.6125	30.8607	0.2558	0.8945	0.3255	0.2047	0.5004	0.1691	0.4821	0.2133	0.2162	2.3431	0.1046
F2	**0.8899**	0.6256	0.1073	0.6510	41.5853	0.2529	0.9113	0.3156	0.2076	0.4997	0.1671	0.4778	0.2229	0.2159	2.2722	0.1121
F3	**1.3934**	1.0262	0.4233	1.2888	49.0339	0.5676	1.2369	0.6236	0.5413	0.8066	0.5154	0.7939	0.5173	0.5129	2.5830	0.4106
F4	**0.8607**	0.6806	0.1034	0.6346	38.5969	0.2549	0.9216	0.3351	0.2137	0.5021	0.1729	0.4871	0.2146	0.2136	2.2448	0.1109
F5	**0.8724**	0.6610	0.1028	0.7070	53.5105	0.2495	0.9262	0.2994	0.2084	0.4918	0.1603	0.4977	0.2610	0.2791	2.3371	0.1474
F6	**0.9012**	0.6999	0.1054	0.6591	37.6092	0.2627	0.9572	0.3508	0.2158	0.5071	0.1683	0.5071	0.2179	0.2148	1.7779	0.1097
F7	**1.1916**	0.9055	0.2803	1.0478	38.3684	0.4363	1.1493	0.5145	0.3915	0.6779	0.3454	0.6585	0.3876	0.3836	2.4391	0.2795
F8	**0.9836**	0.7364	0.1620	0.841	53.1318	0.3150	0.9948	0.2924	0.2687	0.5558	0.2405	0.5741	0.2987	0.2728	0.6552	0.1915
F9	**0.9114**	0.6794	0.1366	0.7432	44.8611	0.2654	0.9624	0.4212	0.2360	0.5069	0.1789	0.4824	0.2456	0.2446	2.2776	0.1288
F10	**1.5472**	1.1122	0.2747	1.1886	25.0825	0.4537	10.4329	4.0984	0.4176	1.0652	1.7266	0.4891	0.2419	0.2451	2.1732	0.1334
F11	**1.5841**	1.1200	0.2563	1.1218	26.3824	0.4030	11.0151	4.2162	0.3829	1.0909	1.7081	0.5648	0.2608	0.2740	2.3868	0.1598
F12	**2.2358**	1.5720	0.6656	1.9178	26.4126	0.8403	11.3332	4.5162	0.7910	1.4928	2.0932	1.0031	0.6938	0.7563	2.761	0.5926
F13	**2.1928**	1.5399	0.6677	1.8993	24.3023	0.7547	10.513	4.2421	0.7427	1.4409	2.0949	0.9589	0.7009	0.6883	2.7562	0.5856
F14	**1.9832**	1.2397	0.9636	2.2071	20.3806	0.9956	1.0704	1.0613	1.0131	0.9908	1.0934	0.9230	0.9330	0.9298	1.5032	0.9221
F15	**0.7730**	0.3868	0.1364	0.9175	12.5796	0.2509	1.6858	0.8230	0.2372	0.4102	0.5477	0.1219	0.0990	0.0995	0.8228	0.0807
F16	**0.6217**	0.3036	0.0931	0.7748	11.5793	0.1719	0.9199	0.4987	0.17013	0.2830	0.3957	0.0830	0.0851	0.0861	0.6975	0.0778
F17	**0.3134**	0.1525	0.0348	0.3672	20.4582	0.0773	0.1429	0.1412	0.0828	0.0809	0.1478	0.0683	0.0730	0.0716	0.6671	0.0641
F18	**0.6375**	0.3048	0.1055	0.8003	11.4402	0.1782	0.90227	0.5357	0.17467	0.3013	0.4031	0.0666	0.0698	0.0692	0.5732	0.0612
F19	**0.3922**	0.2027	0.0641	0.4574	22.9686	0.1100	0.1975	0.1466	0.1107	0.1191	0.1742	0.1056	0.1011	0.1008	0.7456	0.0894
F20	**1.0601**	0.6425	0.2553	1.0995	13.7942	0.3503	2.4634	1.0465	0.3328	0.5979	0.7184	0.1523	0.1185	0.1169	0.8131	0.0960
F21	**1.1449**	0.6368	0.3525	1.3799	12.8497	0.4940	1.9574	1.0249	0.4651	0.6468	0.7706	0.1686	0.1539	0.1525	0.7449	0.1387
F22	**0.5077**	0.2846	0.1312	0.6062	24.6078	0.1970	0.3337	0.2334	0.1942	0.2106	0.2733	0.1894	0.1789	0.1762	0.7527	0.1603
F23	**1.4689**	0.8806	0.5885	1.7646	13.3666	0.6809	2.1961	1.2510	0.6816	0.8889	0.9559	0.2234	0.2085	0.2086	0.7878	0.1930

TABLE 21. Wall-clock time cost for hybrid and composition benchmark functions.

Function	CMSSA	SSA	SCA	WOA	TAPSO	MPSO	IPSO	GWOCS
F24	**0.5079**	0.3805	0.2204	0.2735	0.1915	0.2159	0.2056	1.021
F25	**0.4172**	0.3253	0.1797	0.2131	0.1567	0.1515	0.1566	0.8713
F26	**0.6348**	0.4771	0.3358	0.3508	0.2841	0.3131	0.3094	1.020
F27	**0.5129**	0.3846	0.2472	0.2815	0.2243	0.2229	0.2027	0.9255
F28	**0.5131**	0.3846	0.2473	0.2816	0.2251	0.2231	0.2028	0.9256
F29	**0.5834**	0.4397	0.2820	0.3282	0.2614	0.2731	0.2734	0.9519
F30	**0.5919**	0.4304	0.2911	0.3080	0.2645	0.2654	0.2637	0.9507
F31	**0.6956**	0.5091	0.3572	0.3866	0.3081	0.3418	0.3139	1.0532

convergence rate than other algorithms in most cases except for F_5 and F_{24}. The convergence behavior of CMSSA can be divided into three types. In the first type, with the increase of the number of iterations, the convergence rate of CMSSA is gradually accelerated, which is evident in F_1, F_{10}, F_{25}, and F_{28}. In the second type, the best solution is attained by the last iteration phase, which is evident in F_7, F_{21}, and F_{23}. In the third type, the convergence is rapid in the early iteration phase, which is evident in F_{27}, F_{30}, and F_{31}.

These results show that the CMSSA's ability to balance exploration and exploration has been improved, especially for hybrid and composition problems.

5) ANALYSIS OF WALL-CLOCK TIME COST

The wall-clock time cost implemented by CMSSA and 19 other competitors on 23 benchmark functions and 8 hybrid and composition benchmark functions were shown in Tables 20 and 21, respectively. As it can be seen in Table 20 and 21, it is obvious that the CMSSA consuming time is slightly longer than the basic SSA. This is due to the introduction of four mutation operators (i.e., the improved Tent chaos map mutation operator, Lévy flights mutation operator, elite opposition-based learning mutation operator, and variable radius perturbation mutation operator) in CMSSA, which enhances the balance between exploitation and exploration of CMSSA algorithm. Therefore, it is reasonable to increase the time-consuming of the CMSSA algorithm. Moreover, it is demonstrated that the wall-clock time cost of DA, SHO, and MFO is longer than CMSSA on most of 23 benchmark functions, and that of GWOCS is also longer than CMSSA on 8 hybrid and composition benchmark functions. In general, CMSSA takes longer than many other algorithms. However, according to all the experimental results, the performance of CMSSA outperforms other algorithms in most cases. Therefore, it is very worthwhile to introduce a variety of mutation operators into the basic SSA to strengthen the performance of the algorithm.

6) STATISTICAL TESTING

Apart from standard statistical analysis such as mean value, median value, and standard deviation value, the Wilcoxon statistical test at 5% level of significance and the Friedman

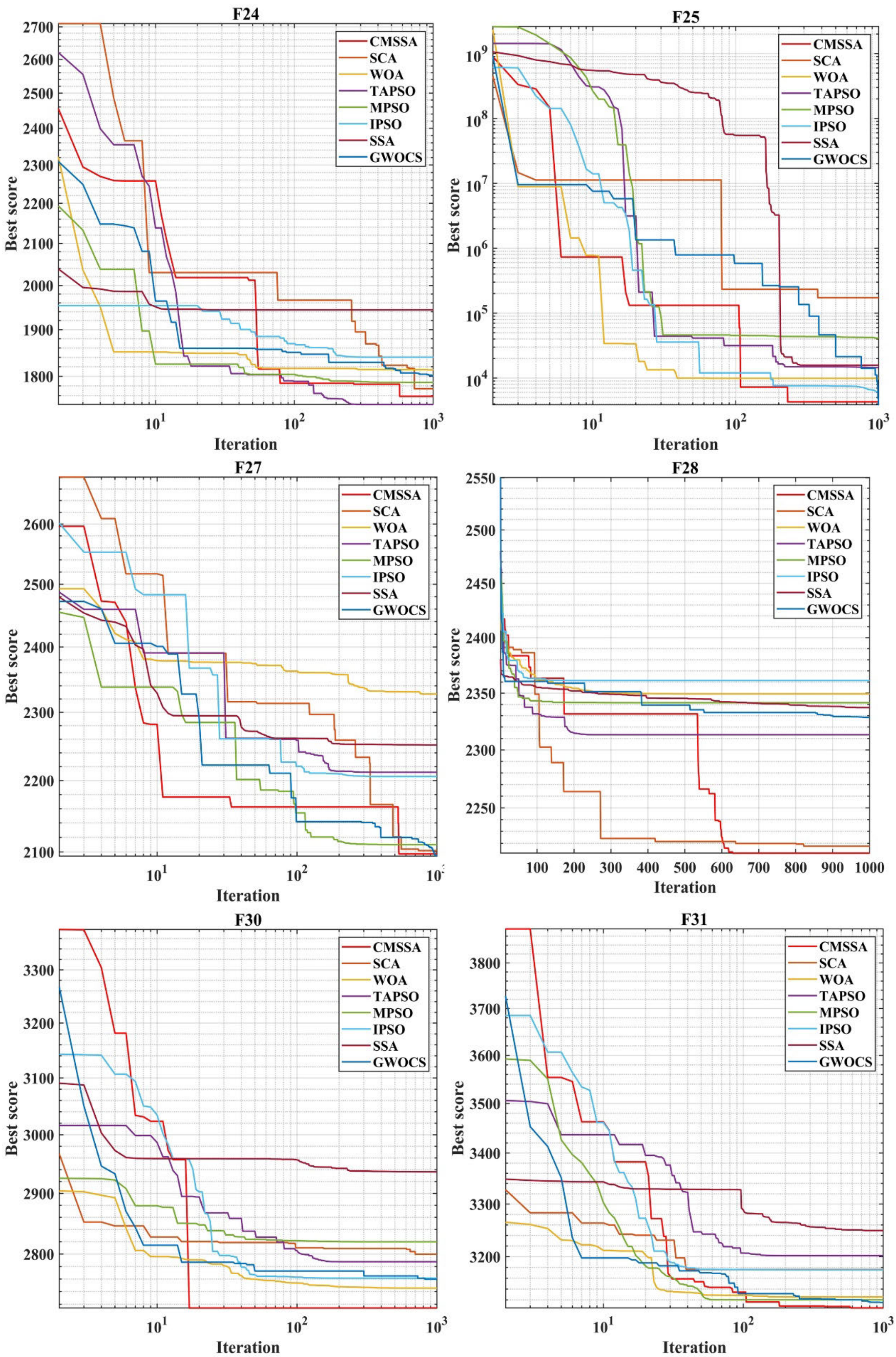

FIGURE 8. Convergence curves on 6 hybrid and composition benchmark functions.

TABLE 22. The *p*-value of Wilcoxon's rank-sum test on 31 benchmark functions.

CMSSA VS.	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12
PSO	1.21E-12	1.21E-12	1.21E-12	1.21E-12	5.61E-05	3.02E-11	3.02E-11	9.35E-01	1.21E-12	1.21E-12	1.21E-12	3.02E-11
CS	1.21E-12	1.21E-12	1.21E-12	1.21E-12	6.07E-11	3.02E-11	3.02E-11	2.41E-02	1.21E-12	1.21E-12	1.21E-12	3.02E-11
DA	1.21E-12	1.21E-12	1.21E-12	1.21E-12	3.02E-11	3.02E-11	3.02E-11	2.71E-02	1.21E-12	1.21E-12	1.21E-12	3.02E-11
GWO	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.86E-09	3.02E-11	4.98E-11	3.25E-01	2.78E-03	5.90E-13	4.19E-02	3.02E-11
MFO	1.21E-12	1.21E-12	1.21E-12	1.21E-12	3.02E-11	3.02E-11	3.02E-11	2.15E-03	1.21E-12	1.21E-12	1.21E-12	3.02E-11
MVO	1.21E-12	1.21E-12	1.21E-12	1.21E-12	6.70E-11	3.02E-11	3.02E-11	1.81E-01	1.21E-12	1.21E-12	1.21E-12	3.02E-11
SCA	1.21E-12	1.21E-12	1.21E-12	1.21E-12	3.02E-11	3.02E-11	3.02E-11	6.72E-10	1.21E-12	1.21E-12	1.21E-12	3.02E-11
SOA	NaN	1.67E-08	1.21E-12	1.21E-12	3.02E-11	3.02E-11	1.41E-01	3.02E-11	NaN	NaN	NaN	3.02E-11
WOA	1.21E-12	1.21E-12	1.21E-12	1.21E-12	2.78E-07	3.02E-11	4.31E-08	9.83E-08	NaN	6.63E-08	3.33E-01	3.02E-11
EPO	1.93E-10	1.21E-12	1.21E-12	1.21E-12	3.02E-11	3.02E-11	1.08E-01	1.09E-10	NaN	6.30E-04	8.87E-07	3.02E-11
TSA	1.21E-12	1.21E-12	1.21E-12	1.21E-12	3.02E-11	3.02E-11	3.02E-11	1.95E-01	1.21E-12	1.12E-12	5.37E-06	3.02E-11
STOA	1.21E-12	1.21E-12	1.21E-12	1.21E-12	3.02E-11	3.02E-11	1.21E-10	6.67E-03	5.63E-11	1.21E-12	1.88E-09	3.02E-11
SHO	NaN	NaN	NaN	NaN	3.02E-11	3.02E-11	5.55E-02	3.02E-11	NaN	NaN	NaN	3.02E-11
RSO	NaN	4.19E-02	NaN	1.10E-02	3.02E-11	3.02E-11	2.05E-03	5.74E-02	NaN	1.31E-03	NaN	3.02E-11
CMSSA VS.	**F13**	**F14**	**F15**	**F16**	**F17**	**F18**	**F19**	**F20**	**F21**	**F22**	**F23**	
PSO	3.02E-11	8.87E-01	3.02E-11	2.48E-06	1.61E-01	2.59E-07	1.15E-10	2.78E-11	5.87E-01	1.94E-07	2.19E-07	
CS	3.02E-11	1.62E-11	7.38E-10	4.19E-11	1.61E-01	1.66E-10	4.57E-12	2.84E-11	1.10E-11	2.63E-10	2.39E-05	
DA	3.02E-11	1.41E-03	3.02E-11	5.47E-09	7.22E-07	4.34E-06	3.02E-11	2.68E-04	1.39E-06	2.88E-06	1.01E-08	
GWO	3.02E-11	4.28E-01	0.00073	7.82E-12	2.37E-12	2.96E-11	3.02E-11	8.12E-04	6.05E-07	8.35E-08	8.35E-08	
MFO	3.02E-11	5.83E-02	3.02E-11	4.19E-11	1.61E-01	1.12E-08	6.19E-10	2.28E-05	5.17E-02	1.66E-01	8.99E-01	
MVO	3.02E-11	7.93E-03	3.02E-11	7.82E-12	2.37E-12	2.96E-11	3.02E-11	1.07E-09	1.36E-07	1.43E-08	8.35E-08	
SCA	3.02E-11	1.53E-01	3.02E-11	7.82E-12	2.37E-12	2.96E-11	3.02E-11	5.57E-10	4.08E-11	2.37E-10	1.78E-10	
SOA	3.02E-11	1.99E-04	3.02E-11	7.82E-12	2.37E-12	2.96E-11	3.02E-11	8.15E-11	2.15E-10	9.92E-11	7.38E-11	
WOA	3.02E-11	7.84E-01	3.69E-11	7.82E-12	2.37E-12	2.96E-11	3.02E-11	1.66E-01	2.03E-07	1.10E-08	1.43E-08	
EPO	3.02E-11	5.49E-01	3.02E-11	9.17E-12	3.16E-12	3.00E-11	3.01E-11	8.84E-07	4.94E-05	6.53E-08	2.78E-07	
TSA	3.02E-11	7.28E-09	8.20E-07	9.17E-12	3.16E-12	3.00E-11	3.01E-11	2.13E-04	9.83E-08	8.10E-10	6.52E-09	
STOA	3.02E-11	6.62E-01	1.21E-10	9.17E-12	3.16E-12	3.00E-11	3.01E-11	2.61E-10	6.53E-08	3.16E-10	1.69E-09	
SHO	3.02E-11	2.79E-10	1.36E-07	9.17E-12	3.16E-12	3.00E-11	3.01E-11	3.02E-11	1.09E-10	4.98E-11	3.16E-10	
RSO	3.02E-11	1.49E-02	3.02E-11	9.17E-12	3.16E-12	3.00E-11	3.01E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	
CMSSA VS.	**F24**	**F25**	**F26**	**F27**	**F28**	**F29**	**F30**	**F31**				
SCA	3.64E-02	3.02E-11	1.26E-01	9.58E-01	1.66E-01	3.35E-08	9.51E-06	1.41E-01				
WOA	6.73E-01	1.15E-01	5.26E-04	8.15E-05	3.09E-06	8.88E-01	4.35E-02	5.08E-03				
TAPSO	5.56E-04	3.11E-01	1.76E-01	6.10E-01	1.02E-01	7.60E-07	3.40E-01	7.66E-05				
MPSO	4.12E-06	2.81E-02	8.76E-01	8.50E-02	2.68E-06	1.55E-09	2.26E-03	1.60E-07				
IPSO	2.78E-07	1.85E-01	3.79E-01	9.47E-01	5.08E-03	9.06E-08	8.49E-02	8.15E-11				
SSA	1.20E-08	6.56E-02	2.23E-09	2.37E-10	2.61E-10	5.07E-10	7.77E-09	3.02E-11				
GWOCS	1.17E-02	1.33E-01	2.34E-01	2.15E-02	4.08E-05	3.81E-07	2.02E-08	9.33E-02				

test are performed in a statistically significant way. The p-values of the Wilcoxon statistical test are shown in Table 22. The statistical results of the Friedman test are tabulated in Tables 23 and 24. Compared with other algorithms, it is observed from Table 22 that the p-value obtained from CMSSA is much smaller than 0.05 for most of the 31 benchmark functions. Tables 23 and 24 show that the AVG. value obtained from CMSSA is the smallest compared with other competitor approaches. Therefore, the results reveal that the proposed CMSSA is statistically different from other competitor algorithms, and outperforms other algorithms.

In conclusion, the discussions and findings in this part illustrate the exploitation and exploration capability, local optimum avoidance, convergence behavior, statistical testing, and wall-clock time cost of the CMSSA algorithm. Compared to other algorithms, the performance of the CMSSA algorithm is better than that of other algorithms in most cases. This is mainly due to the various mutation operators introduced into SSA to balance between the exploitation and exploration capability. In the following section, the applicability of the CMSSA algorithm is evaluated on 8 real-world problems with complex constraints.

E. CMSSA FOR CONSTRAINED REAL-WORLD OPTIMIZATION PROBLEMS

In this section, to effectively evaluate the applicability of CMSSA in terms of constraint handling to optimize constrained problems as well. 8 well-known engineering problems (i.e., gear train design problem, three-bar truss design problem, cantilever beam design problem, tension spring design problem, pressure vessel design problem, speed reducer problem, welded beam design problem, and main girder design problem) were optimized by CMSSA. These problems have various constraints, so some constraint handling methods are used to solve these optimization problems. There are different types of penalty functions to deal with

TABLE 23. The statistical results of Friedman test on 23 benchmark functions.

Algorithm	CMSSA	PSO	CS	DA	GWO	MFO	MVO	SCA
AVG.	**3.130**	6.848	5.543	10.870	6.434	9.586	7.696	11.217
Rank	**1**	6	2	14	4	12	7	15
Algorithm	SOA	WOA	EPO	TSA	STOA	SHO	RSO	
AVG.	8.5	6.5	6.370	10.304	9.348	8.717	8.935	
Rank	8	5	3	13	11	9	10	

TABLE 24. The statistical results of Friedman test on 8 hybrid and composition functions.

Algorithm	CMSSA	SCA	WOA	TAPSO	MPSO	IPSO	SSA	GWOCS
AVG.	**3.125**	4.75	5.875	3.25	3.625	4.125	7.875	3.375
Rank	**1**	6	7	2	4	5	8	3

TABLE 25. Comparison results of the gear train design problem.

Algorithm		Design variables				Gear Ration	f_{min}	error
		T_A	T_B	T_D	T_F			
MIBBSQP [116]		45	22	18	60	0.146666	5.7×10^{-6}	1.65%
IDCNLP [117]		47	29	14	59	0.146411	4.5×10^{-6}	1.47%
SA [118]		52	15	30	60	0.14423	2.36×10^{-9}	0.033%
MVEP [119]		52	15	30	60	0.14423	2.36×10^{-9}	0.033%
Kannan BK [120]		33	15	13	41	0.1441	2.1246×10^{-8}	0.11%
GA [121]		43	16	19	49	0.144281	2.7×10^{-12}	0.00114%
Gene AS [122]		33	14	17	50	0.144242	1.362×10^{-9}	0.026%
HSIA [123]		43	19	16	49	0.144281	2.7×10^{-12}	0.0011%
UPSO [124]		43	19	16	49	0.1442	2.701×10^{-12}	-
CS [11]		43	19	16	49	0.144281	2.7×10^{-12}	-
CAPSO [125]		49	19	16	43	0.1442	2.701×10^{-12}	-
BOA [22]		43	16	19	49	0.1442	2.701×10^{-12}	-
CMSSA	Solution 1	**49**	**16**	**19**	**43**	**0.14428097**	$\mathbf{2.700857148886513\times10^{-12}}$	**0.0011403%**
	Solution 2	**49**	**19**	**16**	**43**	**0.14428097**	$\mathbf{2.700857148886513\times10^{-12}}$	**0.0011403%**
	Solution 3	**43**	**16**	**19**	**49**	**0.14428097**	$\mathbf{2.700857148886513\times10^{-12}}$	**0.0011403%**
	Solution 4	**43**	**19**	**16**	**49**	**0.14428097**	$\mathbf{2.700857148886513\times10^{-12}}$	**0.0011403%**

constraint problems [17] as follows: (a) The static penalty does not rely on the current number of generations and remains constant in the whole computation process. (b) The annealing penalty coefficient changes only once in many iterations and only active constraints that are not trapped in the local optimal value are considered in each iteration. (c) The dynamic penalty in which the current generation is involved in the computation of the equivalent penalty coefficient with the increase of generations. (d) The adaptive penalty attains feedback from the previous search process and only changes when the feasible/infeasible solution is considered to be the best solution in the population. (e) The co-evolutionary penalty is divided into two values (i.e., coeff. and vio.) which are used to find out the constraints that are violated and the corresponding amount of violation. (f) The death penalty deals with a solution that may violate constraints, and its fitness value is zero. It can eliminate infeasible solutions in the optimization process. Compared to other constraint handling techniques, the death penalty approach does not employ the information of infeasible solutions which can better deal with the problems with dominated infeasible regions. Therefore, the CMSSA algorithm is equipped with the death penalty function to solve these constraint problems in this section.

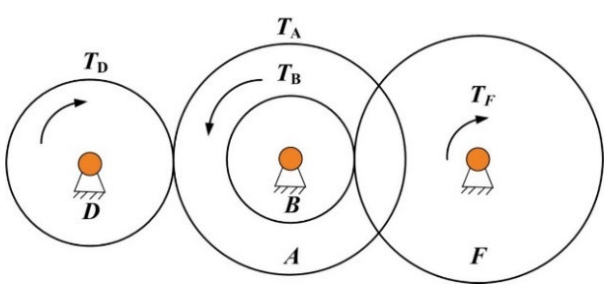

FIGURE 9. Simplified model of the gear train.

TABLE 26. Comparison results of the three-bar truss design problem.

Methods	Optimal solution of design variables		$f_{min}(\boldsymbol{X})$
	x_1	x_2	
CMSSA	**0.788671835599963**	**0.408258294610664**	**263.895910694497**
CS [11]	0.78867	0.40902	263.9716
SSA [20]	0.788665414258065	0.408275784444547	263.8958434
DEDS [126]	0.78867513	0.40824828	263.8958434
MBA [127]	0.7885650	0.4085597	263.8958522
PSO-DE [128]	0.7886751	0.4082482	263.8958433
TAS [136]	0.788	0.408	263.68 (infeasible)

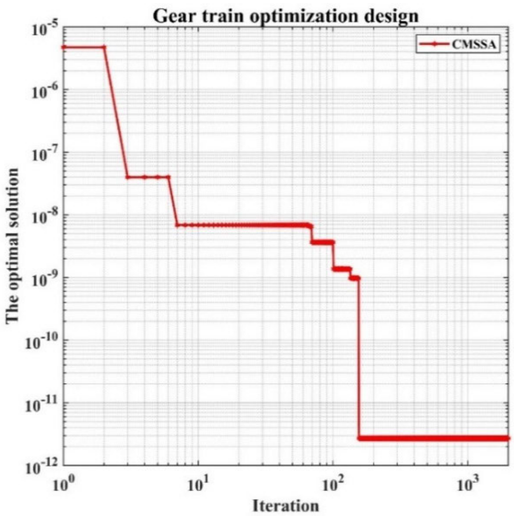

FIGURE 10. The convergence curve of the gear train problem.

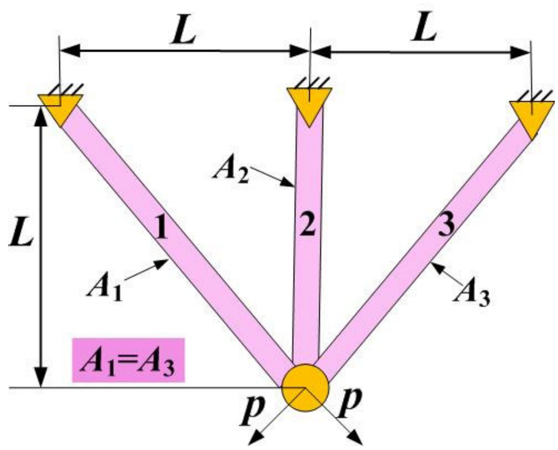

FIGURE 11. Simplified model of the three-bar truss design.

1) GEAR TRAIN OPTIMIZATION DESIGN PROBLEM

According to Ref. [116], the optimal design of the gear train aims to find the minimum gear transmission ratio, making it closer to 1/6.931. The structure of the composite gear train was shown in Fig. 9. T_{A}, T_{B}, T_{D}, and T_{F} are the number of teeth on gears *A*, *B*, *D*, and *F* respectively. Because the number of teeth for each gear must be the integer between 12 and 60, the optimization problem is converted to a constrained optimization problem with discrete variables. The optimization problem is expressed by Eq. (22).

$$\begin{cases} \min f = \left(\dfrac{1}{6.931} - \dfrac{T_D T_B}{T_A T_F}\right)^2 \\ s.t. \quad 12 \leq T_A \leq 60,\ 12 \leq T_B \leq 60,\ 12 \leq T_D \leq 60, \\ \qquad\qquad 12 \leq T_F \leq 60 \end{cases} \tag{22}$$

The convergence curve of the gear train problem is shown in Fig. 10. This problem is solved by the CMSSA and compared to MIBBSQP [116], IDCNLP [117], SA [118], MVEP [119], Kannan BK [120], GA [121], Gene AS [122], HSIA [123], UPSO [124], CS [11], CAPSO [125], and BOA [22]. Comparison results with the reference methods are shown in Table 25.

As it can be seen in Table 25, the CMSSA algorithm obtains the four optimal solutions as follows: the first group of optimal solutions is at $\boldsymbol{X}$ = (49, 16, 19, 43) with a corresponding fitness value equal to f_{min} = 0.14428097. The second group of optimal solutions is at $\boldsymbol{X}$ = (49, 19, 16, 43) with a corresponding fitness value equal to f_{min} = 0.14428097. The third group of optimal solutions is at $\boldsymbol{X}$ = (43, 16, 19, 49) with a corresponding fitness value equal to f_{min} = 0.14428097. The fourth group of optimal solutions is at $\boldsymbol{X}$ = (43, 19, 16, 49) with a corresponding fitness value equal to f_{min} = 0.14428097. By observing Fig. 10, CMSSA converges towards the best solution using low computational efforts.

The results of CMSSA algorithms are superior to MIBB-SQP, IDCNLP, SA, MVEP, Kannan BK, Gene AS, UPSO, CAPSO, and BOA, and slightly inferior to GA, HSIA, and CS. The results reveal that CMSSA can effectively solve the discrete problems.

2) THREE-BAR TRUSS OPTIMIZATION DESIGN PROBLEM

The objective function of the three-bar truss design is to minimize its weight with complex and nonlinear constraints of stress, deformation, and buckling [11]. The structure of the three-bar truss design is depicted in Fig. 11. In this case, the design variables are the cross-sectional area $A_1(x_1)$ and $A_2(x_2)$, and the mathematical optimization model is

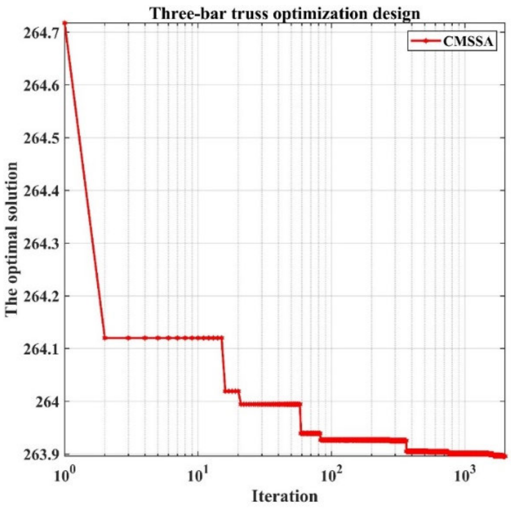

FIGURE 12. The convergence curve of the three-bar truss problem.

illustrated by Eq. (23).

$$\begin{cases} \boldsymbol{X} = [x_1, x_2] = [A_1, A_2] \\ \min f(\boldsymbol{X}) = \left(2\sqrt{2}x_1 + x_2\right) L \\ s.t.\ g_1(\boldsymbol{X}) = \dfrac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} p - \sigma \leq 0 \\ g_2(\boldsymbol{X}) = \dfrac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} p - \sigma \leq 0 \\ g_3(\boldsymbol{X}) = \dfrac{1}{\sqrt{2}x_2 + x_1} p - \sigma \leq 0 \\ 0 \leq x_1, x_2 \leq 1 \end{cases} \tag{23}$$

where $L = 100cm, p = 2KN/cm^2, \sigma = 2KN/cm^2$.

This problem is solved by CMSSA and compared to CS [11], SSA [20], DEDS [126], MBA [127], PSO-DE [128], and TAS [136] in literatures. The convergence curve of the three-bar truss problem is described in Fig. 12. The comparison results of the three-bar truss design problem are illustrated in Table 26. As can be seen in Table 26, the CMSSA algorithm provides the optimal solutions at $\boldsymbol{X} = (0.788671835599963, 0.408258294610664)$ with a corresponding fitness value equal to $f_{min} = 263.895910694497$. And the results obtained by the TAS method are infeasible due to violating one of the constraints.

The results indicate that the CMSSA algorithm attains very competitive results and its best solution obtained is superior to CS, and it also attains the best results close to SSA, DEDS, MBA, and PSO-DE. These results demonstrate that the CMSSA algorithm can also effectively solve non-linear constrained problems.

3) CANTILEVER BEAM OPTIMIZATION DESIGN PROBLEM

As shown in Fig. 13, there are 5 structural parameters (i.e., the side length of the square-shaped cross-section L_i (x_i) ($i = 1, 2, 3, 4, 5$)) in the cantilever beam problem. The objective is to minimize the weight of the cantilever

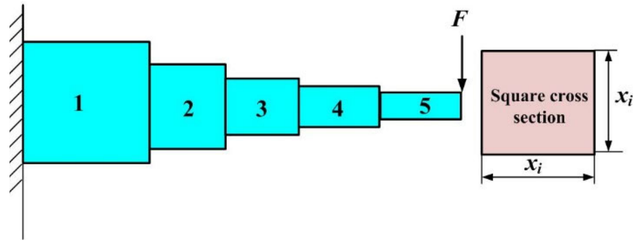

FIGURE 13. Simplified model of the cantilever beam design.

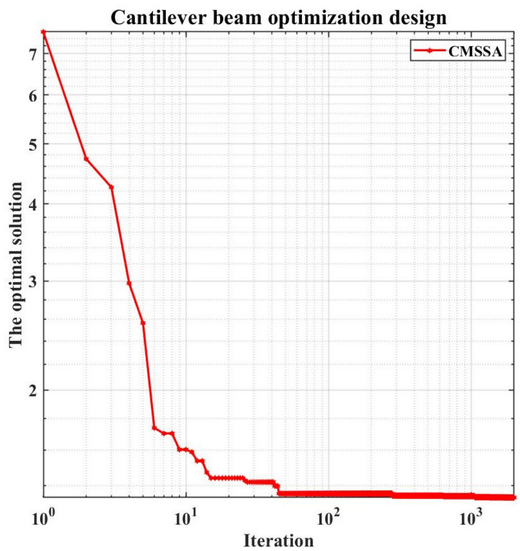

FIGURE 14. The convergence curve of the cantilever beam design problem.

beam with the vertical deformation constraints. The problem formulation is expressed by Eq. (24) [13].

$$\begin{cases} \min f(\boldsymbol{X}) = 0.0624 \times (x_1 + x_2 + x_3 + x_4 + x_5) \\ s.t.\ g(\boldsymbol{X}) = \dfrac{61}{x_1^3} + \dfrac{37}{x_2^2} + \dfrac{19}{x_2^3} + \dfrac{7}{x_4^3} + \dfrac{1}{x_5^3} - 1 \leq 0 \\ 0.01 \leq x_1, x_2, x_3, x_4, x_5 \leq 100 \end{cases} \tag{24}$$

The convergence curve of the cantilever beam design problem is presented in Fig. 14. And this problem has been solved by CMSSA and compared with GOA [18], ALO [14], MMA [129], GCA(I) [129], GCA(II) [129], CS [11], and SOS [13] in Table 27. The CMSSA obtains the optimal solution at $\boldsymbol{X} = (6.010729, 5.318938, 4.499154, 3.494689, 2.150293)$ with the corresponding fitness value equal to $f_{\min} = 1.33996$. The results show that CMSSA outperforms the MMA, GCA(I), GCA(II), and CS. And it is observed the best solution of CMSSA is equal to or close to those of GOA, SOS and ALO with the different optimal design parameters. Therefore, the CMSSA algorithm has some advantages for dealing with such problems.

4) TENSION SPRING OPTIMIZATION DESIGN PROBLEM

This problem is the tension spring design, and the objective is to obtain the minimum fabrication cost [130]. Fig. 15 shows

TABLE 27. Comparison results of the cantilever beam design problem.

Methods	Optimal solution of design variables					$f_{min}(\boldsymbol{X})$
	x_1	x_2	x_3	x_4	x_5	
CMSSA	**6.010729**	**5.318938**	**4.499154**	**3.494689**	**2.150293**	**1.33996**
GOA [18]	6.011674	5.31297	4.48307	3.50279	2.16333	1.33996
ALO [14]	6.01812	5.31142	4.48836	3.49751	2.158329	1.33995
MMA [129]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
GCA(I) [129]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
GCA(II) [129]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
CS [11]	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999
SOS [13]	6.01878	5.30344	4.49587	3.49896	2.15564	1.33996

TABLE 28. Comparison results of the tension spring design problem.

Methods	Optimal solution of design variables			$f_{min}(\boldsymbol{X})$
	x_1	x_2	x_3	
CMSSA	**0.0520769**	**0.3661089**	**10.7606604**	**0.0126699**
SSA [20]	0.051207	0.345215	12.004032	0.0126763
GSA [38]	0.050276	0.323680	13.525410	0.0127022
CPSO [130]	0.051728	0.357644	11.244543	0.0126747
ES [131]	0.051989	0.363965	10.890522	0.0126810
GA [132]	0.051480	0.351661	11.632201	0.0127048
RO [133]	0.051370	0.349096	11.76279	0.0126788

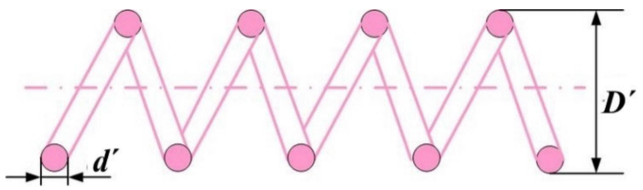

FIGURE 15. Simplified model of the tension spring design.

the structure. There are three parameters: wire diameter d', mean coil diameter D', and the number of active coils N'. Under some complex constraints, the mathematical formulation of this problem is obtained by Eq. (25).

$$\begin{cases} \boldsymbol{X} = [x_1, x_2, x_3] = \left[d', D', N'\right] \\ \min f(\boldsymbol{X}) = (x_3 + 2)\, x_2 x_1^2 \\ s.t.\ g_1(\boldsymbol{X}) = 1 - x_2^3 x_3 / 71785\, x_1^4 \leq 0 \\ g_2(\boldsymbol{X}) = \left(4x_2^2 - x_1 x_2\right) / 12566 \left(x_2 x_1^3 - x_1^4\right) \\ \qquad\qquad + 1/5108\, x_1^2 - 1 \leq 0 \\ g_3(\boldsymbol{X}) = 1 - 140.45 x_1 / x_2^2 x_3 \leq 0 \\ g_4(\boldsymbol{X}) = (x_1 + x_2)/1.5 - 1 \leq 0 \\ 0.05 \leq x_1 \leq 2, 0.25 \leq x_2 \leq 1.3, 2 \leq x_3 \leq 15 \end{cases} \tag{25}$$

There are some solutions obtained by using meta-heuristic algorithms such as SSA [20], GSA [38], CPSO [130], ES [131], GA [132], and RO [133]. The convergence curve

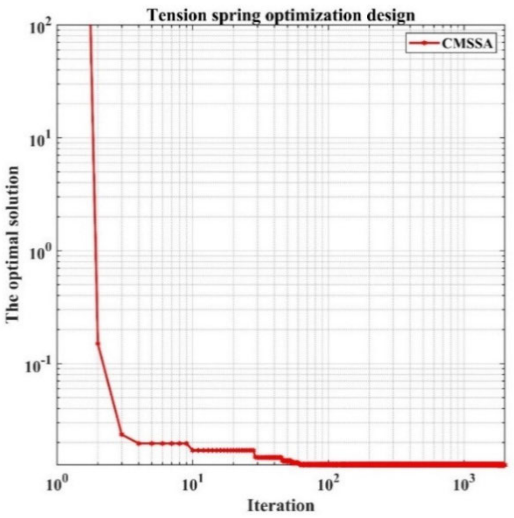

FIGURE 16. The convergence curve of the tension spring design problem.

of the tension spring design problem is shown in Fig. 16. The best solutions of CMSSA are shown in Table 28 compared to those of all the above-mentioned algorithms. The CMSSA attains the optimal solution at $\boldsymbol{X} = (0.0520769, 0.3661089, 10.7606604)$ with a corresponding fitness value equal to $f_{\min} = 0.0126699$.

These results show that CMSSA outperforms other methods when dealing with this problem and attains the best design with the lowest cost.

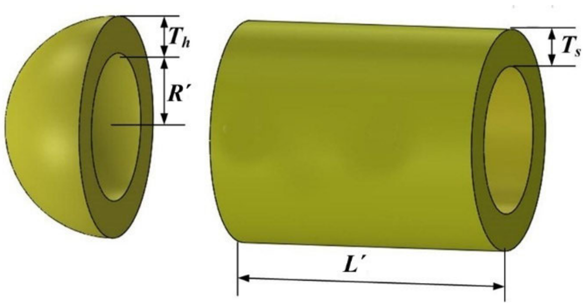

FIGURE 17. Simplified model of the pressure vessel design.

5) PRESSURE VESSEL OPTIMIZATION DESIGN PROBLEM

The pressure vessel design problem has four parameters (i.e., thickness of spherical shell T_S, thickness of ball head T_h, radius of spherical shell R', and length of spherical shell L'). The structure and parameters are shown in Fig. 17. The objective of this problem is to minimize the fabrication cost with some constraints. The mathematical model of this problem is obtained by Eq. (26).

$$\begin{cases} X = [x_1, x_2, x_3, x_4] = [T_s, T_h, R', L'] \\ \min f(X) = 0.6224x_1x_2x_3 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 \\ \qquad + 19.84x_1^2x_3 \\ s.t.\ g_1(X) = -x_1 + 0.0193x_3 \le 0 \\ g_2(X) = -x_2 + 0.0095x_3 \le 0 \\ g_3(X) = -\pi x_3^2 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0 \\ g_4(X) = x_4 - 240 \le 0 \\ 1 \times 0.0625 \le x_1, x_2 \le 99 \times 0.0625,\ 10 \le x_3 \le 200, \\ \qquad 10 \le x_4 \le 240 \end{cases} \tag{26}$$

The pressure vessel design problem is optimized by CMSSA and the results are compared to SMA [28], WOA [17], HHO [26], SHO [19] and MCOA [134]. The convergence curve and comparison results of this problem are shown in Fig. 18 and Table 29, respectively. The CMSSA provides the optimal solution at $X = (0.778216, 0.384684, 40.323097, 199.954457)$ with a corresponding fitness value equal to $f_{\min} = 5885.4120443$.

The results demonstrate that CMSSA can find the first low-cost design compared to other algorithms.

6) SPEED REDUCER OPTIMIZATION DESIGN PROBLEM

The speed reducer design problem has seven design variables as shown in Fig. 19. There are seven design variables (x_1-x_7) which represent the face width B_1 (x_1), module of teeth Z_1 (x_2), a number of teeth in the pinion Z_2 (x_3), length of the first shaft between bearings B_2 (x_4), length of the second shaft between bearings B_3 (x_5), the diameter of first shafts D_1 (x_6), and the diameter of the second shaft D_2 (x_7), respectively. The objective of this problem is to attain the minimum construction cost of the speed reducer with the constraints of bending stress of the gear teeth, surface stress, transverse deflections of the shafts, and stress in the shafts. The mathematical model of this problem is obtained by Eq. (27). The comparison results of the obtained optimal solution with other various competitors (i.e., CS [11], SHO [19], STOA [24], TAS [136], SBS [137], PSO-DE [128], and Ray and Sain [138]) are shown in Table 30.

$$\begin{cases} X = [x_1, x_2, x_3, x_4, x_5, x_6, x_7] \\ \qquad = [B_1, Z_1, Z_2, B_2, B_3, D_1, D_2] \\ f(X) = 0.7854x_1x_2^2\left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) \\ -1.508x_1\left(x_6^2 + x_7^2\right) + 7.4777\left(x_6^3 + x_7^3\right) \\ \qquad + 0.7854(x_4x_6^2 + x_5x_7^2) \\ S.t.g_1(X) = \frac{27}{x_1x_2^2x_3} - 1 \le 0 \\ g_2(X) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \le 0 \\ g_3(X) = \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \le 0 \\ g_4(X) = \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \le 0 \\ g_5(X) = \frac{[(745(x_4/x_2x_3))^2 + 16.9 \times 10^6]^{1/2}}{110x_6^3} - 1 \le 0 \\ g_6(X) = \frac{[(745(x_5/x_2x_3))^2 + 157.5 \times 10^6]^{1/2}}{85x_7^3} - 1 \le 0 \\ g_7(X) = \frac{x_2x_3}{40} - 1 \le 0 \\ g_8(X) = \frac{5x_2}{x_1} - 1 \le 0 \\ g_9(X) = \frac{x_1}{12x_2} - 1 \le 0 \\ g_{10}(X) = \frac{1.5x_6 + 1.9}{x_4} - 1 \le 0 \\ g_{11}(X) = \frac{1.1x_7 + 1.9}{x_5} - 1 \le 0 \\ 2.6 \le x_1 \le 3.6, 0.7 \le x_2 \le 0.8, 17 \le x_3 \le 28 \\ 7.3 \le x_4, x_5 \le 8.3, 2.9 \le x_6 \le 3.9, 5.0 \le x_7 \le 5.5 \end{cases} \tag{27}$$

According to Table 30, the results obtained by the TAS and Ray and Sain methods violate the constraints and are infeasible. It is observed that the CMSSA algorithm can provide an optimal solution at $X = (3.500081, 0.700032, 17, 7.323278, 7.737604, 3.350819, 5.286683)$ with a corresponding fitness value equal to $f_{min} = 2995.564917$. The results indicate that the proposed CMSSA algorithm obtains better results which outperform other competitors (i.e., CS, SHO, STOA, SBS, and PSO-DE). The convergence analysis of the best optimal solution obtained by the CMSSA algorithm is shown in Fig. 20.

TABLE 29. Comparison results of the pressure vessel design problem.

Methods	Optimal solution of design variables				$f_{min}(\boldsymbol{X})$
	x_1	x_2	x_3	x_4	
CMSSA	**0.778216**	**0.384684**	**40.323097**	**199.954457**	**5885.4120443**
SMA [28]	0.7931	0.3932	40.6711	196.2178	5994.1857
WOA [17]	0.812500	0.437500	42.0982699	176.638998	6059.7410
HHO [26]	0.81758383	0.4072927	42.09174576	176.7196352	6000.46259
SHO [19]	0.778210	0.384889	40.315040	200.00000	5885.5773
MCOA [134]	1.1	0.6	56.721834721777	52.5913366743	7037.9770

TABLE 30. Comparison results of the speed reducer design problem.

Methods	Optimal solution of design variables							$f_{min}(\boldsymbol{X})$
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
CMSSA	**3.500081**	**0.700032**	**17**	**7.323278**	**7.737604**	**3.350819**	**5.286683**	**2995.564917**
CS [11]	3.5015	0.7	17	7.6050	7.8181	3.3520	5.2875	3000.9810
SHO [19]	3.50159	0.7	17	7.3	7.8	3.35127	5.28874	2998.5507
STOA [24]	3.50124	0.7	17	7.3	7.8	3.33425	5.26538	2995.9578
SBS [137]	3.506122	0.700006	17	7.549126	7.859330	3.365576	5.289773	3008.08
PSO-DE [128]	3.5	0.7	17	7.3	7.8	3.35021	5.28668	2996.34816
TAS [136]	3.6	0.7	17	7.3	7.8	3.4	5	2876.117623 (infeasible)
Ray and Sain [138]	3.514185	0.700005	17	7.497343	7.834649	2.901786	5.002231	2732.9006 (infeasible)

TABLE 31. Comparison results of the welded beam design problem.

Methods	Optimal solution of design variables				$f_{min}(\boldsymbol{X})$
	x_1	x_2	x_3	x_4	
CMSSA	**0.205410**	**3.258999**	**9.036343**	**0.2057659**	**1.695799**
MFO [15]	0.2057	3.4703	9.0364	0.2057	1.72452
WOA [17]	0 .205396	3 .484293	9 .037426	0 .206276	1 .730499
RO [133]	0 .203687	3 .528467	9 .004233	0 .207241	1 .735344
SHO [19]	0.205563	3.474846	9.035799	0.205811	1.725661
STOA [24]	0.205415	3.472346	9.035220	0.201160	1.723590
SOA [23]	0.205408	3.472316	9.035208	0.201141	1.723485
SSA [20]	0.2057	3.4714	9.0366	0.2057	1.72491
MVO [44]	0.205463	3.473193	9.044502	0.205695	1.72645
GWO [12]	0.205676	3.478377	9.03681	0.205778	1.72624

TABLE 32. Comparison results of main girder design problem.

Methods	Optimal solution of design variables				$f_{min}(\boldsymbol{X})$
	d_1	d_2	d_3	d_4	
CMSSA	**747.6007**	**350**	**5.000**	**5.8333**	**5779.6702**
CGA [139]	799.0784	350.2652	5.021	5.8796	6071.6127
GA-AN2 [140]	791.968	320.440	5.000	6.627	6113.396
Normal way [141]	790	310	5.0	8	6430

7) WELDED BEAM OPTIMIZATION DESIGN PROBLEM

The main objective of the welded beam problem is to minimize the fabrication cost. The simplified model of the welded beam design is described in Fig. 21. There are four design variables of this problem which can be described as the thickness of weld (h), length of the clamped bar (l), the height of the bar (t), and thickness of the bar (b), respectively. This problem is subjected to the constraints of shear stress in the beam, bending stress in the beam, buckling load on the beam, and end deflection of the beam. The mathematical model of this problem is obtained by Eq. (28). The comparison results of the obtained optimal solution

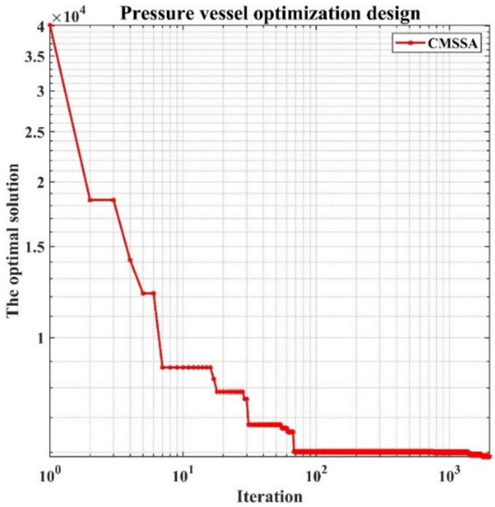

FIGURE 18. The convergence curve of the pressure vessel design problem.

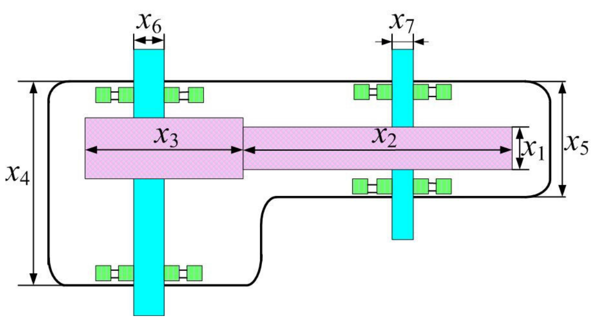

FIGURE 19. Simplified model of the speed reducer design.

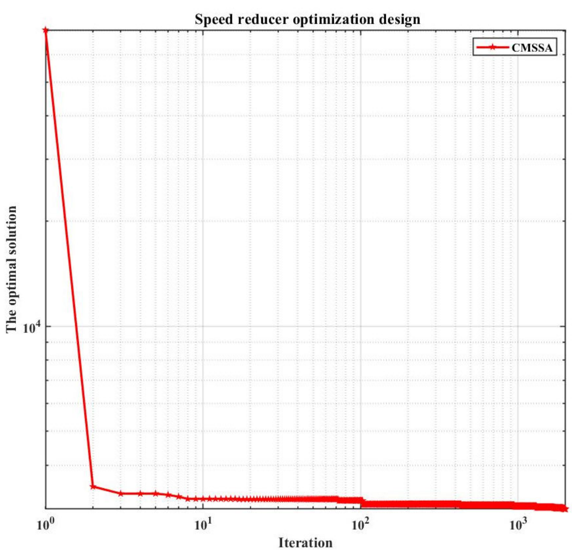

FIGURE 20. The convergence curve of the speed reducer design problem.

with other various competitors (i.e., MFO [15], WOA [17], RO [133], SHO [19], STOA [24], SOA [23] SSA [20], MVO [44], and GWO [12]) are shown in Table 31. As it can be seen in Table 31, the CMSSA algorithm can attain the optimal solution at $\boldsymbol{X} = (0.205410, 3.258999, 9.036343, 0.2057659, 1.695799)$ with a corresponding fitness value

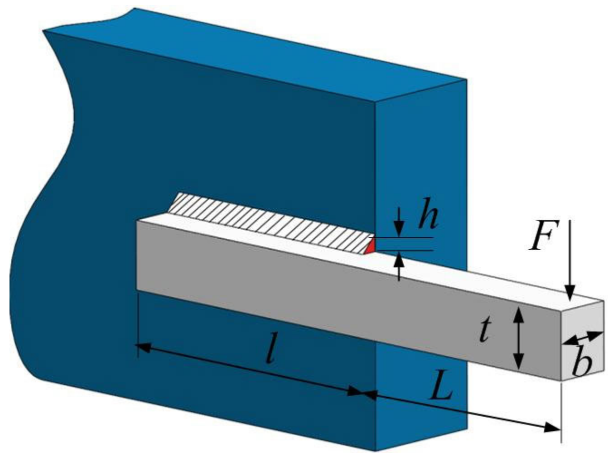

FIGURE 21. Simplified model of the welded beam design.

equal to $f_{min} = 1.695799$. The results show that the CMSSA algorithm can find the best optimal design compared to other algorithms (i.e., MFO, WOA, RO, SHO, STOA, SOA, SSA, MVO, and GWO). By observing Fig. 22, the CMSSA algorithm can obtain the near-optimal solution in the initial iteration process.

$$\begin{cases} \boldsymbol{X} = [x_1, x_2, x_3, x_4] = [h, l, t, b] \\ f(\boldsymbol{X}) = 1.0471x_1^2x_2 + 0.04811x_3x_4(14 + x_2) \\ S.t.\, g_1(\boldsymbol{X}) = \tau(\boldsymbol{X}) - \tau_{max} \le 0 \\ g_2(\boldsymbol{X}) = \sigma(\boldsymbol{X}) - \sigma_{max} \le 0 \\ g_3(\boldsymbol{X}) = x_1 - x_4 \le 0 \\ g_4(\boldsymbol{X}) = \delta(\boldsymbol{X}) - \delta_{max} \le 0 \\ g_5(\boldsymbol{X}) = 1.0471x_1^2x_2 + 0.04811x_3x_4(14 + x_2) - 5.0 \le 0 \\ g_6(\boldsymbol{X}) = 0.125 - x_1 \le 0 \\ g_7(\boldsymbol{X}) = F - F_C(\boldsymbol{X}) \le 0 \\ \tau(\boldsymbol{X}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2} \\ \tau' = \frac{F}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J} \\ M = F\left(L + \frac{x_1}{2}\right), R = \sqrt{\frac{x_2^2}{4} + ((x_1 + x_3)/2)^2} \\ J = 2\left\{\sqrt{2}\,x_1x_2\left[\frac{x_2^2}{4} + ((x_1 + x_3)/2)^2\right]\right\} \\ \sigma(\boldsymbol{X}) = \frac{6FL}{x_4x_3^2}, \delta(\boldsymbol{X}) = \frac{6FL^3}{Ex_3^2x_4} \\ F_c(\boldsymbol{X}) = \frac{4.013E\sqrt{x_3^2x_4^6/36}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \\ F = 6000lb, L = 14in, E = 30 \times 10^6 psi \\ G = 12 \times 10^6 psi, \tau_{max} = 13.6 \times 10^3 psi \\ \sigma_{max} = 30 \times 10^3 psi, \delta_{max} = 0.25in \\ 0.1 \le x_{1,4} \le 2, 0.1 \le x_{2,3} \le 10 \end{cases} \tag{28}$$

8) MAIN GIRDER OPTIMIZATION DESIGN PROBLEM

The lightweight design of the main girder in the bridge crane should meet the requirements of strength, stiffness, and

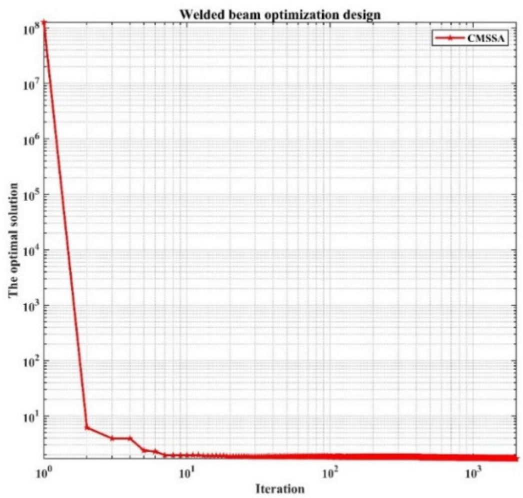

FIGURE 22. The convergence curve of the welded beam design problem.

stability, etc. The simplified structure of the main girder is shown in Fig. 23. There are four design variables of this problem such as the height of the main girder (b_1), the width of the main girder(b_2), the thickness of the web plate(b_3), and thickness of the flange plate(b_4), respectively. The mathematical model of this problem is obtained by Eq. (29).

$$\begin{cases} \boldsymbol{X} = [x_1,\ x_2,\ x_3,\ x_4] = [b_1,\ b_2,\ b_3,\ b_4] \\ \min f(\boldsymbol{X}) = x_1x_3 + x_2x_4 \\ s.t.\ g_1(\boldsymbol{X}) = 0.75 \times 10500 \\ \quad \times \left[\dfrac{120000+7.8\times10^{-5}\times10500\,(x_1x_3+x_2x_4)}{3x_1x_2x_4 + x_1^2x_3} + \dfrac{12000}{3x_1x_2x_3 + x_2^2x_4} \right] - 140 \leq 0 \\ g_2(\boldsymbol{X}) = \dfrac{120000 \times 10500^3}{\left(3x_1^2x_2x_4 + x_1^3x_3\right) \times 1.68 \times 10^6} - \dfrac{10500}{700} \leq 0 \\ g_3(\boldsymbol{X}) = \dfrac{x_2}{x_4} - 60 \leq 0 \\ g_4(\boldsymbol{X}) = \dfrac{x_1}{x_3} - 160 \leq 0 \\ 700 \leq x_1 \leq 800,\ 350 \leq x_2 \leq 400,\ 5 \leq x_3, x_4 \leq 10 \end{cases} \tag{29}$$

The comparison results of the obtained optimal solution with other various competitors (i.e., CGA [139], GA-AN2 [140], and Normal way [141]) are shown in Table 32. As it can be seen in Table 32, the CMSSA algorithm can attain the optimal solution at $\boldsymbol{X} = (747.6007, 350, 5.000, 5.8333)$ with a corresponding fitness value equal to $f_{min} = 5779.6702$. The results show that the CMSSA algorithm can find the best optimal design compared to other algorithms (i.e., CGA, GA-AN2, and Normal way). By observing Fig. 24, the CMSSA algorithm can obtain the near-optimal solution with low computational cost.

To sum up, the results of eight real-world engineering problems indicate that the CMSSA algorithm has high performance in solving various challenging problems. The optimization results show that the CMSSA algorithm has a better capability to handle different combinatorial optimization problems. Thus, the CMSSA algorithm is the best optimizer that can provide better optimization results with low computational cost and a fast convergence rate.

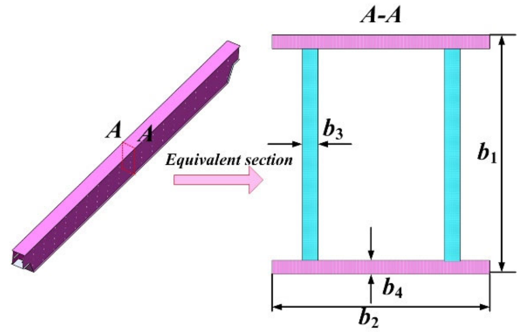

FIGURE 23. Simplified model of the main girder design.

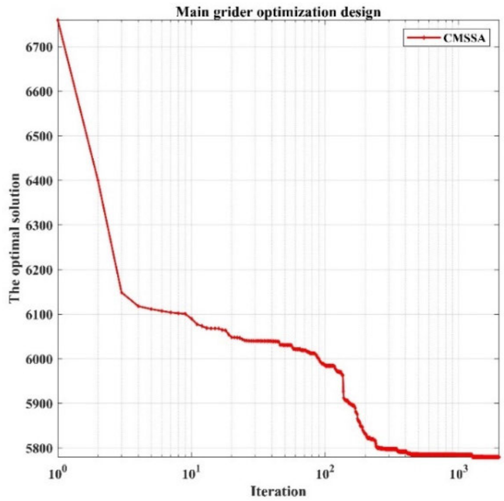

FIGURE 24. The convergence curve of the main girder design problem.

V. CONCLUSION AND FUTURE WORK

To further improve the performance of the SSA algorithm, this paper presents a new series of SSA variants, namely, ITSSA, LFSSA, EOBLSSA, VRSSA, and CMSSA. All the proposed algorithms are tested on a set of thirty-one benchmark functions to evaluate the exploration and exploitation phases for avoiding local optimum. Initially, the results reveal that the CMSSA algorithm is the best among these variants.

Moreover, compared to 19 well-known optimization algorithms, the CMSSA algorithm is tested on thirty-one benchmark functions to analyze the exploration, exploitation, local optima avoidance, convergence behavior, and time-consuming. The results on these test functions show that the CMSSA algorithm is the best optimizer which provides very competitive results as compared to other optimizers. Statistical testing has been carried out to demonstrate the superiority of the CMSSA algorithm compared to other metaheuristics. In addition, the CMSSA algorithm has been employed to eight real-world constrained engineering design problems (i.e., gear train, three-bar truss, cantilever beam, tension spring, pressure vessel, speed reducer, welded beam, and main girder) which demonstrates that the CMSSA algorithm has high-performance capability in unknown search spaces.

This paper puts forward several research directions like the CMSSA algorithm that may be applied to solve multi-objective optimization problems in future work. Also, Binary and multi-objective versions of the CMSSA algorithm can be seen as an interesting direction for future contribution.

APPENDIX A

See Tables 5–8.

REFERENCES

[1] J. H. Holland, "Genetic algorithms," *Sci. Amer.*, vol. 267, no. 1, pp. 66–73, 1992.

[2] J. R. Koza, "Genetic programming as a means for programming computers by natural selection," *Statist. Comput.*, vol. 4, no. 2, pp. 87–112, Jun. 1994.

[3] R. Storn and K. Price, "Differential evolution simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[4] X. Yao and Y. Liu, "Fast evolution strategies," in *Evolutionary Programming* (Lecture Notes in Computer Science), P. J. Angeline, R. G. Reynolds, J. R. McDonnell, and R. Eberhart, Eds), vol. 1213. Berlin, Germany: Springer, 1997, pp. 149–161.

[5] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.

[6] D. Simon, "Biogeography-based optimization," *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 702–713, Dec. 2008.

[7] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE ICNN*, vol. 4, Perth, WA, Australia, Nov./Dec. 1995, pp. 1942–1948.

[8] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Global Optim.*, vol. 39, no. 3, pp. 459–471, Apr. 2007.

[9] X. Yang, "Firefly algorithms for multimodal optimization," in *Stochastic Algorithms: Foundations and Applications* (Lecture Notes in Computer Science), vol. 5792, O. Watanabe and T. Zeugmann, Eds. Berlin, Germany: Springer, 2009, pp. 169–178.

[10] X.-S. Yang and A. H. Gandomi, "Bat algorithm: A novel approach for global engineering optimization," *Eng. Comput.*, vol. 29, no. 5, pp. 464–483, 2012.

[11] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems," *Eng. Comput.*, vol. 29, no. 1, pp. 17–35, 2011.

[12] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[13] M.-Y. Cheng and D. Prayogo, "Symbiotic organisms search: A new metaheuristic optimization algorithm," *Comput. Struct.*, vol. 139, pp. 98–112, Jul. 2014.

[14] S. Mirjalili, "The ant lion optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, May 2015.

[15] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.

[16] S. Mirjalili, "Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Comput. Appl.*, vol. 27, no. 4, pp. 1053–1073, 2016.

[17] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.

[18] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: Theory and application," *Adv. Eng. Softw.*, vol. 105, pp. 30–47, Mar. 2017.

[19] G. Dhiman and V. Kumar, "Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications," *Adv. Eng. Softw.*, vol. 114, pp. 48–70, Dec. 2017.

[20] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017.

[21] G. Dhiman and V. Kumar, "Emperor penguin optimizer: A bio-inspired algorithm for engineering problems," *Knowl. Based Syst.*, vol. 159, pp. 20–50, Nov. 2018.

[22] S. Arora and S. Singh, "Butterfly optimization algorithm: A novel approach for global optimization," *Soft Comput.*, vol. 23, no. 3, pp. 715–734, 2018.

[23] G. Dhiman and V. Kumar, "Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems," *Knowl.-Based Syst.*, vol. 165, pp. 169–196, Feb. 2018.

[24] G. Dhiman and A. Kaur, "STOA: A bio-inspired based optimization algorithm for industrial engineering problems," *Eng. Appl. Artif. Intell.*, vol. 82, pp. 148–174, Jun. 2019.

[25] M. Dehghani, Z. Montazeri, O. P. Malik, G. Dhiman, and V. Kumar, "BOSA: Binary orientation search algorithm," *Int. J. Innov. Technol. Exploring Eng.*, vol. 9, no. 1, pp. 5306–5310, 2019.

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

[27] S. Shadravan, H. R. Naji, and V. K. Bardsiri, "The sailfish optimizer: A novel nature-inspired Metaheuristic algorithm for solving constrained engineering optimization problems," *Eng. Appl. Artif. Intell.*, vol. 80, pp. 20–34, Apr. 2019.

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

[29] S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, "Tunicate swarm algorithm: A new bio-inspired based Metaheuristic paradigm for global optimization," *Eng. Appl. Artif. Intell.*, vol. 90, Apr. 2020, Art. no. 103541.

[30] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: Sparrow search algorithm," *Syst. Sci. Control Eng.*, vol. 8, no. 1, pp. 22–34, Jan. 2020.

[31] G. Dhiman, M. Garg, A. Nagar, V. Kumar, and M. Dehghani, "A novel algorithm for global optimization: Rat swarm optimizer," *J. Ambient Intell. Hum. Comput.*, vol. 12, no. 8, pp. 8457–8482, Aug. 2021.

[32] G. Dhiman, D. Oliva, A. Kaur, K. K. Singh, S. Vimal, A. Sharma, and K. Cengiz, "BEPO: A novel binary emperor penguin optimizer for automatic feature selection," *Knowl.-Based Syst.*, vol. 211, Jan. 2021, Art. no. 106560.

[33] G. Dhiman, "ESA: A hybrid bio-inspired Metaheuristic optimization approach for engineering problems," *Eng. With Comput.*, vol. 37, no. 1, pp. 323–353, Jan. 2021.

[34] J.-S. Chou and D.-N. Truong, "A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean," *Appl. Math. Comput.*, vol. 389, Jan. 2021, Art. no. 125535.

[35] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[36] O. K. Erol and I. Eksin, "A new optimization method: Big bang-big crunch," *Adv. Eng. Softw.*, vol. 37, no. 2, pp. 106–111, 2006.

[37] R. A. Formato, "Central force optimization: A new nature inspired computational framework for multidimensional search and optimization," in *Nature Inspired Cooperative Strategies for Optimization* (Studies in Computational Intelligence), vol. 129, N. Krasnogor, G. Nicosia, M. Pavone, and D. Pelta, Eds. Berlin, Germany: Springer, 2018, pp. 221–238.

[38] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *J. Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, 2009.

[39] J. J. Flores, R. Lápez, and J. Barrera, "Gravitational interactions optimization," in *Learning and Intelligent Optimization* (Lecture Notes in Computer Science), vol. 6683, C. A. Coello, Ed. Berlin, Germany: Springer, 2011, pp. 226–237.

[40] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, "Water cycle algorithm—A novel Metaheuristic optimization method for solving constrained engineering optimization problems," *Comput. Struct.*, vols. 110–111, pp. 151–166, Nov. 2012.

[41] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," *Inf. Sci.*, vol. 222, pp. 175–184, Feb. 2012.

[42] Q. Qisong, X. Gening, and F. Xiaoning, "A new specular reflection optimization algorithm," *Adv. Mech. Eng.*, vol. 7, no. 10, pp. 1–10, 2015.

[43] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016.

[44] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 495–513, 2016.

[45] A. Kaveh and A. Dadras, "A novel meta-heuristic optimization algorithm: Thermal exchange optimization," *Adv. Eng. Softw.*, vol. 110, pp. 69–84, Aug. 2017.

[46] A. Sajwan and A. Yadav, "AEFA: Artificial electric field algorithm for global optimization," *Swarm Evol. Comput.*, vol. 48, pp. 93–108, Aug. 2019.

[47] M. Dehghani, Z. Montazeri, G. Dhiman, O. P. Malik, R. Morales-Menendez, R. A. Ramirez-Mendoza, A. Dehghani, J. M. Guerrero, and L. Parra-Arroyo, "A spring search algorithm applied to engineering optimization problems," *Appl. Sci.*, vol. 10, no. 18, p. 6173, Sep. 2020.

[48] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems," *Comput.-Aided Des.*, vol. 43, no. 3, pp. 303–315, Mar. 2011.

[49] J. Xue, Y. Wu, Y. Shi, and S. Cheng, "Brain storm optimization algorithm for multi-objective optimization problems," in *Advances in Swarm Intelligence* (Lecture Notes in Computer Science), vol. 7331, Y. Tan, Y. Shi, and Z. Ji, Eds. Berlin, Germany: Springer, 2012, pp. 513–519.

[50] F. Ramezani and S. Lotfi, "Social-based algorithm (SBA)," *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2837–2856, May 2013.

[51] A. H. Kashan, "League championship algorithm (LCA): An algorithm for global optimization inspired by sport championships," *Appl. Soft Comput.*, vol. 16, pp. 171–200, Mar. 2014.

[52] S. Satapathy and A. Naik, "Social group optimization (SGO): A new population evolutionary optimization technique," *Complex Intel. Syst.*, vol. 2, no. 3, pp. 173–203, 2016.

[53] S. Balochian and H. Baloochian, "Social mimic optimization algorithm and engineering applications," *Expert Syst. Appl.*, vol. 134, pp. 178–191, Nov. 2019.

[54] S. H. Samareh Moosavi and V. K. Bardsiri, "Poor and rich optimization algorithm: A new human-based and multi populations algorithm," *Eng. Appl. Artif. Intell.*, vol. 86, pp. 165–181, Nov. 2019.

[55] Q. Askari, I. Younas, and M. Saeed, "Political optimizer: A novel socio-inspired meta-heuristic for global optimization," *Knowl.-Based Syst.*, vol. 195, May 2020, Art. no. 105709.

[56] M. Dehghani, Z. Montazeri, H. Givi, J. Guerrero, G. Dhiman, and G. Bikram, "Darts game optimizer: A new optimization technique based on darts game," *Int. J. Intell. Eng. Syst.*, vol. 13, no. 5, pp. 286–294, Oct. 2020.

[57] Q. Askari, M. Saeed, and I. Younas, "Heap-based optimizer inspired by corporate rank hierarchy for global optimization," *Expert Syst. Appl.*, vol. 161, Dec. 2020, Art. no. 113702.

[58] M. A. Lones, "Mitigating metaphors: A comprehensible guide to recent nature-inspired algorithms," *Social Netw. Comput. Sci.*, vol. 1, no. 1, Jan. 2020, Art. no. 49.

[59] O. Olorunda and A. P. Engelbrecht, "Measuring exploration/exploitation in particle swarms using swarm diversity," in *Proc. IEEE Congr. Evol. Comput.*, Hong Kong, 2008, pp. 1128–1134.

[60] K. Sörensen, "Metaheuristics—The metaphor exposed," *Int. Trans. Oper. Res.*, vol. 22, no. 1, pp. 3–18, Jan. 2015.

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

[62] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli, "Hybrid metaheuristics in combinatorial optimization: A survey," *Appl. Soft Comput.*, vol. 11, no. 6, pp. 4135–4151, 2011.

[63] M. Corazza, G. Fasano, and R. Gusso, "Particle swarm optimization with non-smooth penalty reformulation, for a complex portfolio selection problem," *Appl. Math. Comput.*, vol. 224, pp. 611–624, Mar. 2013.

[64] F. D'Andreagiovanni, J. Krolikowski, and J. Pulaj, "A fast hybrid primal heuristic for multiband robust capacitated network design with multiple time periods," *Appl. Soft Comput.*, vol. 26, pp. 497–507, Dec. 2015.

[65] F. D'Andreagiovanni, F. Mett, A. Nardin, and J. Pulaj, "Integrating LP-guided variable fixing with MIP heuristics in the robust design of hybrid wired-wireless FTTx access networks," *Appl. Soft Comput.*, vol. 61, pp. 1074–1087, Mar. 2017.

[66] M. H. Mozaffari, H. Abdy, and S. H. Zahiri, "IPO: An inclined planes system optimization algorithm," *Comput. Inf.*, vol. 35, no. 1, pp. 222–240, May 2016.

[67] A. Mohammadi, "Compare the performance of heuristic algorithms GA, IPO and PSO for optimal design of the LEVEL SHIFTER circuit," *J. Soft Comput. Inf. Technol.*, vol. 5, no. 2, pp. 40–50, 2018.

[68] M. H. Mozaffari, H. Abdy, and S. H. Zahiri, "Application of inclined planes system optimization on data clustering," in *Proc. 1st Iranian Conf. Pattern Recognit. Image Anal. (PRIA)*, Mar. 2013, pp. 1–3.

[69] M. H. Mozaffari and S. H. Zahiri, "Unsupervised data and histogram clustering using inclined planes system optimization algorithm," *Image Anal. Stereol.*, vol. 33, no. 1, pp. 65–74, 2014.

[70] A. Soltany Mahboob and S. H. Zahiri, "Application of IPO: A heuristic neuro-fuzzy classifier," *Evol. Intell.*, vol. 12, no. 2, pp. 165–177, Jun. 2019.

[71] N. S. Shahraki and S.-H. Zahiri, "Inclined planes optimization algorithm in optimal architecture of MLP neural networks," in *Proc. 3rd Int. Conf. Pattern Recognit. Image Anal. (IPRIA)*, Apr. 2017, pp. 189–194.

[72] A. Mohammadi and S. H. Zahiri, "Analysis of swarm intelligence and evolutionary computation techniques in IIR digital filters design," in *Proc. 1st Conf. Swarm Intell. Evol. Comput. (CSIEC)*, Mar. 2016, pp. 64–69.

[73] A. Mohammadi and S. H. Zahiri, "Inclined planes system optimization algorithm for IIR system identification," *Int. J. Mach. Learn. Cybern.*, vol. 9, no. 3, pp. 541–558, Mar. 2018.

[74] A. Mohammadi, S. H. Zahiri, and S. M. Razavi, "Infinite impulse response systems modeling by artificial intelligent optimization methods," *Evolving Syst.*, vol. 10, no. 2, pp. 221–237, Jun. 2019.

[75] M. R. Esmaeili and S. H. Zahiri, "Epileptic seizure detection using Inclined Planes system Optimization algorithm (IPO)," *Signal Data Process.*, vol. 13, no. 4, pp. 29–42, 2017.

[76] R. B. Darband, A. Soltany Mahboob, S. Shahbandegan, and S. M. Razavi, "Multimodal optimization using inclined planes system optimization algorithm," in *Proc. 6th Iranian Conf. Signal Process. Intell. Syst. (ICSPIS)*, Dec. 2020, pp. 1–5.

[77] N. S. Shahraki and S. H. Zahiri, "Multi-objective optimization algorithms in analog active filter design," in *Proc. 8th Iranian Joint Congr. Fuzzy Intell. Syst. (CFIS)*, 2020, pp. 105–109.

[78] A. Mohammadi and S. H. Zahiri, "IIR model identification using a modified inclined planes system optimization algorithm," *Artif. Intell. Rev.*, vol. 48, no. 2, pp. 237–259, Aug. 2017.

[79] A. Mohammadi, M. Mohammadi, and S. H. Zahiri, "Design of optimal CMOS ring oscillator using an intelligent optimization tool," *Soft Comput.*, vol. 22, no. 24, pp. 8151–8166, Dec. 2018.

[80] A. Mohammadi, M. Mohammadi, and S. H. Zahiri, "A novel solution based on multi-objective ai techniques for optimization of CMOS LC_VCOs," *J. Telecommun., Electron. Comput. Eng.*, vol. 7, no. 2, pp. 137–144, 2015.

[81] S. Mohammadi-Esfahrood, A. Mohammadi, and S. H. Zahiri, "A simplified and efficient version of inclined planes system optimization algorithm," in *Proc. 5th Conf. Knowl. Based Eng. Innov. (KBEI)*, Tehran, Iran, Feb. 2019, pp. 504–509.

[82] A. Soltany Mahboob and S. H. Zahiri, "Variable length IPO and its application in concurrent design and train of ANFIS systems," *Int. J. Speech Technol.*, vol. 49, no. 6, pp. 2233–2255, Jun. 2019.

[83] L. Xin, "Chaos sparrow search optimization algorithm," *J. Beijing Univ. Aeronaut. Astronaut.*, vol. 47, no. 8, pp. 1712–1720, 2021.

[84] M. Qinghua and Z. Qiang, "Improved sparrow algorithm combining Cauchy mutation and Opposition-based learning," *J. Frontiers Comput. Sci.*, vol. 4, p. 15, Dec. 2020. [Online]. Available: http://kns.cnki.net/kcms/detail/11.5602.tp.20201203.1601.006.html

[85] Y. Zhu and N. Yousefi, "Optimal parameter identification of PEMFC stacks using adaptive sparrow search algorithm," *Int. J. Hydrogen Energy*, vol. 46, no. 14, pp. 9541–9552, Feb. 2021.

[86] J. Yuan, Z. Zhao, Y. Liu, B. He, L. Wang, B. Xie, and Y. Gao, "DMPPT control of photovoltaic microgrid based on improved sparrow search algorithm," *IEEE Access*, vol. 9, pp. 16623–16629, 2021.

[87] G. Liu, C. Shu, Z. Liang, B. Peng, and L. Cheng, "A modified sparrow search algorithm with application in 3D route planning for UAV," *Sensors*, vol. 21, no. 4, p. 1224, Feb. 2021.
[88] T. Liu, Z. Yuan, L. Wu, and B. Badami, "An optimal brain tumor detection by convolutional neural network and enhanced sparrow search algorithm," *Proc. Inst. Mech. Eng., H, J. Eng. Med.*, vol. 235, no. 4, pp. 459–469, Apr. 2021.
[89] J. Zhang, K. Xia, Z. He, Z. Yin, and S. Wang, "Semi-supervised ensemble classifier with improved sparrow search algorithm and its application in pulmonary nodule detection," *Math. Problems Eng.*, vol. 2021, pp. 1–18, Feb. 2021.
[90] C. Zhang and S. Ding, "A stochastic configuration network based on chaotic sparrow search algorithm," *Knowl.-Based Syst.*, vol. 220, Mar. 2021, Art. no. 106924.
[91] C. Ouyang, D. Zhu, and Y. Qiu, "Lens learning sparrow search algorithm," *Math. Problems Eng.*, vol. 2021, pp. 1–17, May 2021.
[92] J. Zhou and D. Chen, "Carbon price forecasting based on improved CEEMDAN and extreme learning machine optimized by sparrow search algorithm," *Sustainability*, vol. 13, no. 9, p. 4896, Apr. 2021.
[93] L. Xu, D. Cai, W. Shen, and H. Su, "Denoising method for fiber optic gyro measurement signal of face slab deflection of concrete face rockfill dam based on sparrow search algorithm and variational modal decomposition," *Sens. Actuators A, Phys.*, vol. 331, Nov. 2021, Art. no. 112913.
[94] W. Tuerxun, X. Chang, G. Hongyu, J. Zhijie, and Z. Huajian, "Fault diagnosis of wind turbines based on a support vector machine optimized by the sparrow search algorithm," *IEEE Access*, vol. 9, pp. 69307–69315, 2021.
[95] C. Ouyang, Y. Qiu, and D. Zhu, "A multi-strategy improved sparrow search algorithm," *J. Phys., Conf. Ser.*, vol. 1848, no. 1, Apr. 2021, Art. no. 012042.
[96] L. Yang, Z. Li, D. Wang, H. Miao, and Z. Wang, "Software defects prediction based on hybrid particle swarm optimization and sparrow search algorithm," *IEEE Access*, vol. 9, pp. 60865–60879, 2021.
[97] Y. Lei, G. De, and L. Fei, "Improved sparrow search algorithm based DV-Hop localization in WSN," in *Proc. Chin. Autom. Congr. (CAC)*, 2020, pp. 2240–2244.
[98] O. Chengtian, L. Yujia, and Z. Donglin, "An adaptive chaotic sparrow search optimization algorithm," in *Proc. IEEE 2nd Int. Conf. Big Data, Artif. Intell. Internet Things Eng. (ICBAIE)*, Mar. 2021, pp. 76–82.
[99] P. Wang, Y. Zhang, and H. Yang, "Research on economic optimization of microgrid cluster based on chaos sparrow search algorithm," *Comput. Intell. Neurosci.*, vol. 2021, Mar. 2021, Art. no. 5556780.
[100] F. J. Kuang, W. H. Xu, and Z. Jin, "Artificial bee colony algorithm based on self-adaptive Tent chaos search," *Control Theory Appl.*, vol. 31, no. 11, pp. 1502–1509, 2014.
[101] L. Shan, "Chaos optimization algorithm based on Tent map," *Control Decis.*, vol. 2, pp. 179–182, Dec. 2005.
[102] N. Zhang, "The gravitational search algorithm based on improved Tent chaos," *Control Decis.*, vol. 35, no. 4, pp. 893–900, 2020.
[103] X. S. Yang, "Firefly algorithm, Lévy flights and global optimization," in *Research and Development in Intelligent Systems*, M. Bramer, R. Ellis, and M. Petridis, Eds. London, U.K.: Springer, 2010, pp. 209–218.
[104] H. R. Tizhoosh, "Opposition-based learning: A new scheme for machine intelligence," in *Proc. Int. Conf. Comput. Intell. Modeling, Control Autom.*, 2005, pp. 695–701.
[105] S. Mahdavi, S. Rahnamayan, and K. Deb, "Opposition based learning: A literature review," *Swarm Evol. Comput.*, vol. 39, pp. 1–23, Apr. 2018.
[106] P. Wang, W. Gao, X. Qian, H. Gou, and S. Wang, "Hybrid fireworks explosion optimization algorithm using elite opposition-based learning," *J. Comput. Appl.*, vol. 34, no. 10, pp. 2886–2890, 2014.
[107] J. G. Digalakis and K. G. Margaritis, "On benchmarking functions for genetic algorithms," *Int. J. Comput. Math.*, vol. 77, no. 4, pp. 481–506, Sep. 2001.
[108] X. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *Int. J. Bio-Inspired Comput.*, vol. 2, no. 2, pp. 78–84, 2010.
[109] A. Kumar, R. K. Misra, and D. Singh, "Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retreat phase," *Proc. IEEE Congr. Evol. Comput. (CEC)*, Donostia, Spain, Dec. 2017, pp. 1835–1842.
[110] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, Mar. 2011.
[111] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*. London, U.K.: Chapman & Hall, 2007.
[112] T. Ziyu and Z. Dingxue, "A modified particle swarm optimization with an adaptive acceleration coefficients," in *Proc. Asia–Pacific Conf. Inf. Process.*, Jul. 2009, pp. 330–332.
[113] G. Q. Bao and K. F. Mao, "Particle swarm optimization algorithm with asymmetric time varying acceleration coefficients," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2009, pp. 2134–2139.
[114] Z. Cui, J. Zeng, and Y. Yin, "An improved PSO with time-varying accelerator coefficients," in *Proc. 8th Int. Conf. Intell. Syst. Design Appl.*, Kaohsuing, Taiwan, Nov. 2008, pp. 638–643.
[115] H. Xu, X. Liu, and J. Su, "An improved grey wolf optimizer algorithm integrated with cuckoo search," in *Proc. 9th IEEE Int. Conf. Intell. Data Acquisition Adv. Comput. Systems: Technol. Appl. (IDAACS)*, Bucharest, Romania, Sep. 2017, pp. 490–493.
[116] E. Sandgren, "Nonlinear integer and discrete programming in mechanical design optimization," *J. Mech. Des.*, vol. 112, no. 2, pp. 223–229, 1990.
[117] J.-F. Fu, R. G. Fenton, and W. L. Cleghorn, "A mixed integer-discrete-continuous programming method and its application to engineering design optimization," *Eng. Optim.*, vol. 17, no. 4, pp. 263–280, Jun. 1991.
[118] C. Zhang and H.-P. Wang, "Mixed-discrete nonlinear optimization with simulated annealing," *Eng. Optim.*, vol. 21, no. 4, pp. 277–291, Sep. 1993.
[119] C. Yj and W. Qh, "Mechanical design optimization by mixed-variable evolutionary programming," in *Proc. IEEE Int. Conf. Evol. Comput. (ICEC)*, Indianapolis, IN, USA, Dec. 1997, pp. 443–446.
[120] B. K. Kannan and S. N. Kramer, "An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design," *J. Mech. Des.*, vol. 116, no. 2, pp. 405–411, 2008.
[121] S.-J. Wu and P.-T. Chow, "Genetic algorithms for nonlinear mixed discrete integer optimization problems via meta-genetic parameter optimization," *Eng. Optim.*, vol. 24, no. 2, pp. 137–159, May 1995.
[122] K. Deb and M. Goyal, "A combined genetic adaptive search (Gene AS) for engineering design," *J. Comput. Sci. Inform.*, vol. 26, pp. 30–45, Dec. 1996.
[123] C.-X. Guo, J.-S. Hu, B. Ye, and Y.-J. Cao, "Swarm intelligence for mixed-variable design optimization," *J. Zhejiang Univ.*, vol. 5, no. 7, pp. 851–860, Jul. 2004.
[124] K. E. Parsopoulos and M. N. Vrahatis, "Unified particle swarm optimization for solving constrained engineering optimization problems," in *Proc. Int. Conf. Natural Comput., Adv. Natural Comput.* (Lecture Notes in Computer Science), vol. 3612, L. Wang, K. Chen, and Y. S. Ong, Eds. Berlin, Germany: Springer, 2005, pp. 582–591.
[125] A. H. Gandomi, G. J. Yun, X.-S. Yang, and S. Talatahari, "Chaos-enhanced accelerated particle swarm optimization," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 18, no. 2, pp. 327–340, Feb. 2013.
[126] M. Zhang, W. Luo, and X. Wang, "Differential evolution with dynamic stochastic selection for constrained optimization," *Inf. Sci.*, vol. 178, no. 15, pp. 3043–3074, Aug. 2008.
[127] A. Sadollah, A. Bahreininejad, H. Eskandar, and M. Hamdi, "Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems," *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2592–2612, May 2013.
[128] H. Liu, Z. Cai, and Y. Wang, "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization," *Appl. Soft Comput.*, vol. 10, no. 2, pp. 629–640, Mar. 2010.
[129] H. Chickermane and H. C. Gea, "Structural optimization using a new local approximation method," *Int. J. Numer. Methods Eng.*, vol. 39, no. 5, pp. 829–846, 1996.
[130] Q. He and L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems," *Eng. Appl. Artif. Intell.*, vol. 20, pp. 89–99, Feb. 2007.
[131] E. Mezura-Montes and C. A. C. Coello, "An empirical study about the usefulness of evolution strategies to solve constrained optimization problems," *Int. J. Gen. Syst.*, vol. 37, no. 4, pp. 443–473, 2008.
[132] C. A. C. Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Comput. Ind.*, vol. 41, no. 2, pp. 113–127, Mar. 2000.
[133] A. Kaveh and M. Khayatazad, "A new meta-heuristic method: Ray optimization," *Comput. Struct.*, vols. 112–113, pp. 283–294, Dec. 2012.
[134] H. Kahramanli, "A modified cuckoo optimization algorithm for engineering optimization," *Int. J. Future Comput. Commun.*, vol. 1, no. 2, pp. 199–201, 2012.

[135] C. A. C. Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art," *Comput. Methods Appl. Mech. Eng.*, vol. 191, pp. 1245–1287, Jan. 2002.

[136] K. J. Ku, S. S. Rao, and L. Chen, "Taguchi-aided search method for design optimization of engineering systems," *Eng. Optim.*, vol. 30, no. 1, pp. 1–23, Jan. 1998.

[137] S. Akhtar, K. Tai, and T. Ray, "A socio-behavioural simulation model for engineering design optimization," *Eng. Optim.*, vol. 34, no. 4, pp. 341–354, Jan. 2002.

[138] T. Ray and P. Saini, "Engineering design optimization using a swarm with an intelligent information sharing among individuals," *Eng. Optim.*, vol. 33, no. 6, pp. 735–748, Aug. 2001.

[139] H. Guo, X. Che, and W. Xiao, "Chaos-genetic optimal algorithm and application in mechanical optimal design," *J. Mech. Design*, vol. 20, no. 10, pp. 23–25, 2003.

[140] Z. Zhengjia, H. Hongzhong, and C. Xin, "A genetic-neural network algorithm in optimum design," *J. Southwest Jiaotong Univ.*, vol. 35, no. 1, pp. 67–70, 2000.

[141] S. Guozheng, *Optimization Design and Application*. Beijing, China: People's Transportation Press, 1992.

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

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

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

YONGGANG LIU received the B.S. degree from the College of Mechanical and Electrical Engineering, Taiyuan University of Science and Technology, Taiyuan, Shanxi, China. He is currently a Senior Engineer at Henan Weihua Heavy Machinery Company Ltd., Changyuan, Henan, China. His research interest includes the green design of crane.

QIANG ZHOU received the M.S. degree from the College of Mechanical and Electrical Engineering, Taiyuan University of Science and Technology, Taiyuan, Shanxi, China. He is currently a Senior Engineer at Henan Weihua Heavy Machinery Company Ltd., Changyuan, Henan, China. His research interests include the electrical design and control of lifting machinery.

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

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

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