



# Hybrid Sine Cosine Algorithm with Integrated Roulette Wheel Selection and Opposition-Based Learning for Engineering Optimization Problems

Vu Hong Son Pham<sup>1</sup> · Nghiep Trinh Nguyen Dang<sup>1</sup> · Van Nam Nguyen<sup>1</sup>

Received: 10 July 2023 / Accepted: 17 October 2023  
© The Author(s) 2023

## Abstract

The sine cosine algorithm (SCA) is widely recognized for its efficacy in solving optimization problems, although it encounters challenges in striking a balance between exploration and exploitation. To improve these limitations, a novel model, termed the novel sine cosine algorithm (nSCA), is introduced. In this advanced model, the roulette wheel selection (RWS) mechanism and opposition-based learning (OBL) techniques are integrated to augment its global optimization capabilities. A meticulous evaluation of nSCA performance has been carried out in comparison with state-of-the-art optimization algorithms, including multi-verse optimizer (MVO), salp swarm algorithm (SSA), moth-flame optimization (MFO), grasshopper optimization algorithm (GOA), and whale optimization algorithm (WOA), in addition to the original SCA. This comparative analysis was conducted across a wide array of 23 classical test functions and 29 CEC2017 benchmark functions, thereby facilitating a comprehensive assessment. Further validation of nSCA utility has been achieved through its deployment in five distinct engineering optimization case studies. Its effectiveness and relevance in addressing real-world optimization issues have thus been emphasized. Across all conducted tests and practical applications, nSCA was found to outperform its competitors consistently, furnishing more effective solutions to both theoretical and applied optimization problems.

**Keywords** Evolutionary algorithm · Stochastic optimization · Sine cosine algorithm · Roulette wheel selection · Opposition-based learning

## 1 Introduction

### 1.1 Evolutionary Algorithm

In recent years, there has been a growing scholarly emphasis on the exploration of nature-inspired optimization algorithms, primarily due to their remarkable capabilities in addressing complex optimization challenges. Within this context, the work of Mirjalili, Mirjalili [1] introduced the

MVO algorithm, which draws inspiration from cosmological concepts. Their study not only demonstrates its competitive performance across benchmark assessments but also within real-world engineering scenarios, highlighting its potential to tackle complex challenges characterized by intricate search spaces. Similarly, Mirjalili [2] presented SCA, showcasing its effectiveness through rigorous benchmark testing and the optimization of an aircraft wing's cross-section. The study emphasizes its promise in resolving intricate real-world problems, particularly those constrained by both the complexity and the obscurity of their search domains.

It is noteworthy that the landscape of optimization algorithms includes the differential evolution (DE) method developed by Storn and Price [3], renowned for its simplicity and effectiveness in global optimization. Building on this trajectory, Mirjalili [4] introduced the MFO, inspired by the transverse orientation behavior of moths. The investigation demonstrates the algorithm's competitiveness through comprehensive benchmark tests and applications in real-world engineering domains. Notably, Mirjalili and Lewis [5]

✉ Nghiep Trinh Nguyen Dang  
ndntrinh.sdh20@hcmut.edu.vn

Vu Hong Son Pham  
pvhson@hcmut.edu.vn

Van Nam Nguyen  
nvnam.sdh20@hcmut.edu.vn

<sup>1</sup> Faculty of Civil Engineering, Ho Chi Minh City University of Technology (HCMUT), Vietnam National University (VNU-HCM), Ho Chi Minh City, Vietnam

pioneered the WOA, drawing inspiration from the intricate social behavior of humpback whales. Their work resonates with the algorithm's competitive prowess, illustrated through exhaustive assessments of mathematical optimization landscapes and the intricacies of structural design problems.

Adding further to this diverse spectrum, Saremi, Mirjalili [6] proposed GOA, deriving insights from the collective behavior of grasshopper swarms. Their study provides compelling evidence of its efficacy in solving optimization challenges, supported by rigorous benchmarking exercises and practical applications to intricate structural optimization scenarios. Finally, the work of Mirjalili, Gandomi [7] unveiled SSA, inspired by the cooperative swarming behavior of salps. Their comprehensive exploration demonstrates the algorithm's effectiveness in both single and multi-objective optimization landscapes, validated through mathematical function evaluations and real-world engineering design complexities.

In evolutionary algorithms reliant on population-based methods, the optimization process is commonly divided into two critical phases, irrespective of the algorithm's specific characteristics [8, 9]. The initial phase, often referred to as exploration, is designed to scan the search landscape and identify high-potential regions. In this phase, significant shifts in directions are made, potentially leading to notable results. The subsequent phase, known as exploitation, focuses on refining the existing choices based on the data that has been gathered during the exploration phase. These data are employed to facilitate the algorithm's convergence. Achieving a judicious balance between exploration and exploitation is considered essential for the effective accomplishment of comprehensive optimization by the algorithm.

The ongoing advancements in algorithmic design and optimization have captured significant scholarly attention [10]. This focus is substantiated by the commonly held belief that no single algorithm can universally address diverse optimization challenges. Consequently, a strong motivation has been observed among researchers to either augment existing methodologies or develop innovative algorithms capable of competing effectively with established solutions. In the specific area of multi-facility production scheduling, Pham, Trang [11] introduced an integration of the gray wolf optimizer (GWO) and the dragonfly algorithm (DA) to enhance optimization processes. In a similar vein, Son and Nguyen Dang [12] proposed an MVO model aimed at simultaneous time and cost optimization in small-scale scenarios. In the realm of environmental impact, Qiao, Lu [13] unveiled a hybrid algorithm that merges the lion swarm optimizer with a genetic algorithm (GA). The algorithm was found to improve both the stability and accuracy of carbon dioxide emissions forecasts, outperforming existing models. Regarding structural optimization, a study by Altay, Cetindemir [14] evaluated the SSA and introduced a modified version,

termed modified SSA (MSSA), for optimizing truss system structures. The study found that, unlike SSA, MSSA effectively addresses convergence issues and proves especially effective for discrete problems. In the domain of construction, Pham and Souliya [15] proposed a hybrid ant-lion optimizer (ALO) algorithm. This algorithm demonstrated improved capabilities for site layout planning by combining optimization techniques with heuristic methods. Meanwhile, Goksal, Karaoglan [16] introduced a heuristic solution for the vehicle routing problem, an NP-hard problem, by utilizing a PSO algorithm enhanced with variable neighborhood descent (VND) for local searches. Furthermore, Son, Duy [17] introduced a novel optimization algorithm that merges the DA and PSO to control construction material costs effectively.

## 1.2 Sine Cosine Algorithm

Since its inception in 2016, the SCA has garnered significant attention as a potential optimization technique. Its applications span diverse fields, addressing an array of complex issues. For example, in the realm of engineering, Shang, Zhou [18] unveiled a modified SCA to expedite convergence speed and promote population diversity. This modification involved redefining the position update formula and incorporating a Levy random walk mutation strategy for solving intricate engineering design problems. In the field of electrical networks, Raut and Mishra [19] introduced an SCA variant specifically tailored for the power distribution network reconfiguration (PDNR) problem. The algorithm aimed to minimize power loss as its sole objective. In a similar vein, Reddy, Panwar [20] presented a binary SCA aimed at optimizing the profit-based unit commitment (PBUC) problem in competitive electricity markets, demonstrating enhanced solution quality and convergence rates compared to existing methods. Within the sphere of bioinformatics and environmental science, Sahlol, Ewees [21] employed an SCA-optimized neural network model to enhance the prediction accuracy of oxidative stress biomarkers in fish liver tissue. Specifically, the model demonstrated improved performance when assessing the impact of varying selenium nanoparticle concentrations. For community detection and system modeling, Zhao, Zou [22] presented a discrete SCA tailored for community detection in complex networks. The algorithm showed superior effectiveness compared to existing methods like FM, BGLL, and GA on real-world network data. Aydin, Gozde [23] utilized both WOA and SCA for estimating critical parameters in photovoltaic (PV) cell models, targeting improved accuracy in system analysis and electrical generation efficiency.

Given the diverse nature of optimization problems, it is widely acknowledged that there is no universally applicable optimization algorithm competent in addressing diverse

optimization problems [10]. As a result, there have been numerous investigations aimed at improving the effectiveness of the SCA. For instance, Cheng and Duan [24] proposed a hybrid version that combines SCA and the cloud model to handle benchmark test functions with different dimensions. Bureerat and Pholdee [25] developed a hybrid model that combines SCA and DE for detecting structural damage. Turgut [26] proposed a model that integrates the SCA with the backtracking search algorithm to effectively address multi-objective problems in heat exchanger design. Bairathi and Gopalani [27] improved SCA by integrating the opposition-based mechanism to instruct multi-layer neural networks. Qu, Zeng [28] introduced an upgraded version of the SCA by incorporating a neighborhood search technique and a greedy Levy mutation. Son and Nguyen Dang [29] proposed a hybrid SCA model to optimize simultaneously time and cost in large-scale projects. Finally, Pham and Nguyen [30] proposed an integrated SCA version with tournament selection, OBL, and mutation and crossover methods to handle cement transport routing.

### 1.3 The Motivation of this Study

Since its introduction, the SCA has witnessed growing popularity across various scientific disciplines, a trend primarily attributed to its straightforward methodology. However, the algorithm has been criticized for its tendency toward premature convergence, a drawback often ascribed to an inadequately defined exploitation strategy within its search landscape [31]. As a result, academic interest has been piqued in the development of enhanced versions of the SCA framework, viewed as potential solutions for overcoming the intricate challenges frequently encountered in optimization tasks.

Numerous efforts have been undertaken to enhance the efficacy of the SCA, encompassing a range of strategies including its fusion with OBL [27], its integration with tournament selection [30], incorporation of the Levy flight approach [18, 28], and hybridizations with other algorithmic paradigms [25, 26, 28]. However, the integration of both the RWS and OBL methodologies to achieve a harmonious balance between the exploration and exploitation phases remains an underexplored area. This comprehensive integration aims to culminate in the pursuit of global optimization. Within the research landscape, this study endeavors to address this notable gap by embarking on a journey to unify the RWS and OBL techniques. This unification not only seeks to bridge an existing research void but also aims to present a streamlined and efficient tool for tackling optimization challenges, catering to a distinct requirement within the research panorama.

In the following section, the formulation of the nSCA is detailed. Section 3 is devoted to an exhaustive evaluation of the algorithm's convergence properties, including an

analysis of its performance metrics and behavioral patterns. Section 4 provides an empirical substantiation of the model's efficacy, achieved through its application in five real-world optimization case studies. Finally, the key findings of the research are summarized in Sect. 5, where potential avenues for future academic inquiry are also delineated.

## 2 Novel Version of Sine Cosine Algorithm

### 2.1 Roulette Wheel Selection (RWS)

The RWS mechanism is extensively employed across various optimization algorithms, including cuckoo search (CS), PSO, DE, GA, and ant colony optimization (ACO), marking its prominence as a commonly adopted technique in optimization disciplines. Pandey, Kulhari [32] introduced a roulette wheel-based cuckoo search clustering method for sentiment analysis. This method was found to outperform existing clustering methods like K-means and GWO in terms of mean accuracy, precision, and recall across nine sentimental datasets. Zhu, Yang [33] introduced a ranking weight-based RWS method to enhance the performance of comprehensive learning PSO. Experimental results indicate that this method surpasses other selection techniques in overall optimization efficiency. Yu, Fu [34] presented an improved RWS method designed for GA, targeting the traveling salesman problem. The method showed enhanced result precision and faster convergence rates. Ho-Huu, Nguyen-Thoi [35] introduced ReDE, a variant of the DE algorithm enhanced with RWS and elitist techniques. This variant was aimed at optimizing truss structures with frequency constraints, and numerical results suggest it outperforms several existing optimization methods. Lloyd and Amos [36] conducted the first comprehensive analysis of Independent Roulette (I-Roulette), an alternative to standard RWS in parallel ACO. The study revealed its capability for dynamic adaptation and faster convergence, especially when implemented on high-performance parallel architectures like GPUs.

### 2.2 Opposition-Based Learning (OBL)

The OBL technique has garnered significant attention for its wide-ranging applicability and effectiveness in various optimization applications. Originally introduced by Tizhoosh [37] in 2005, OBL serves as a novel framework for computational intelligence, creating complementary solutions to existing ones. Subsequent work has extended the utility of OBL in different computational algorithms, thereby yielding promising results in terms of faster convergence and improved performance. For example, Verma, Aggarwal [38] proposed a modified firefly algorithm that incorporates OBL. This innovation not only enhances initial candidate solutions but also employs

a dimension-based approach for updating the positions of individual fireflies. Experimental results confirmed faster convergence and superior performance in high-dimensional problems when compared to existing evolutionary algorithms. Similarly, Upadhyay, Kar [39] presented an opposition-based harmony search algorithm aimed at optimizing adaptive infinite impulse response system identification. They reported faster convergence rates and superior mean square error fitness values when compared to traditional optimization methods such as GA, PSO, and DE. In the realm of project management, Luong, Tran [40] introduced a novel algorithm termed opposition-based multiple objective differential evolution. This algorithm employs opposition numbers to address the time–cost-quality trade-off in construction projects, thereby improving both exploration and convergence rates. Wang, Wu [41] proposed an enhanced PSO algorithm named GOPSO, which incorporates generalized opposition-based learning along with Cauchy mutation. This approach was specifically designed to mitigate the problem of premature convergence in complex optimization scenarios. Ewees, Abd Elaziz [42] introduced OBLGOA, an enhanced GOA that incorporates OBL at two distinct stages. This implementation was shown to improve solution quality and reduce time complexity. The algorithm outperformed ten well-known optimization algorithms across twenty-three benchmark functions and four engineering problems. In summary, OBL has been effectively integrated into a variety of optimization algorithms, consistently offering advantages in terms of speed and performance.

### 2.3 Novel Version of SCA (nSCA)

In the nSCA algorithm, the location of each solution is specified by an array of variables. These arrays collectively constitute sets of solutions, which are systematically organized in a matrix format, as described in Eq. (1). Similarly, the sets of opposite solutions generated during the exploration stage are also presented in a matrix layout, as delineated in Eq. (2). These matrix-based representations facilitate the management and assessment of solutions within the algorithm, thereby enabling more effective exploration and optimization of the search landscape.

$$S = \begin{bmatrix} s_1^1 & s_1^2 & \dots & s_1^d \\ s_2^1 & s_2^2 & \dots & s_2^d \\ \dots & \dots & \dots & \dots \\ s_N^1 & s_N^2 & \dots & s_N^d \end{bmatrix}, \tag{1}$$

$$S^* = \begin{bmatrix} s_1^{1*} & s_1^{2*} & \dots & s_1^{d*} \\ s_2^{1*} & s_2^{2*} & \dots & s_2^{d*} \\ \dots & \dots & \dots & \dots \\ s_N^{1*} & s_N^{2*} & \dots & s_N^{d*} \end{bmatrix}. \tag{2}$$

In the initial population generation phase, the OBL method is utilized to create opposite solutions, as illustrated in Fig. 1. The specific process for incorporating OBL within nSCA is outlined in the accompanying pseudocode presented in Table 1. Subsequently, a fitness function evaluates both the randomly generated solutions and their oppositional counterparts. This evaluation identifies superior and inferior solutions. The algorithm retains the more performant solutions while discarding the less effective ones, thereby ensuring a consistent population size throughout the optimization process.

The opposite solution  $s^*$  of the solution  $s \in [b_l, b_u]$  can be identified as follow:

$$s^* = b_u + b_l - s, \tag{3}$$

where  $b_l$  and  $b_u$  denote the lower and upper boundary of alternative  $s$ , respectively.

Given a solution  $S$  characterized by  $d$  parameters, where each parameter constrained within  $[b_{l,j}, b_{u,j}]$ , an opposition solution  $S^* = (s_1^*, s_2^*, s_3^*, \dots, s_d^*)$  can be defined as follow:

$$s_j^* = b_{u,j} + b_{l,j} - s_j, \tag{4}$$

where  $b_{l,j}$  and  $b_{u,j}$  show the lower and upper limits of the  $j$ th dimension, respectively.

Upon refreshing the solution set during the initial population creation phase, the solutions undergo sorting to identify the current best-performing candidate. Subsequently, each solution's normalized fitness score is computed. This computation is integral to the functioning of the RWS mechanism, as depicted in Fig. 2. The formula for calculating the normalized fitness score is articulated in Eq. (5), while the mathematical representation of the RWS mechanism is provided in Eq. (6). These computational processes and mechanisms are pivotal in guiding the algorithm's solution selection and subsequent exploratory activities.

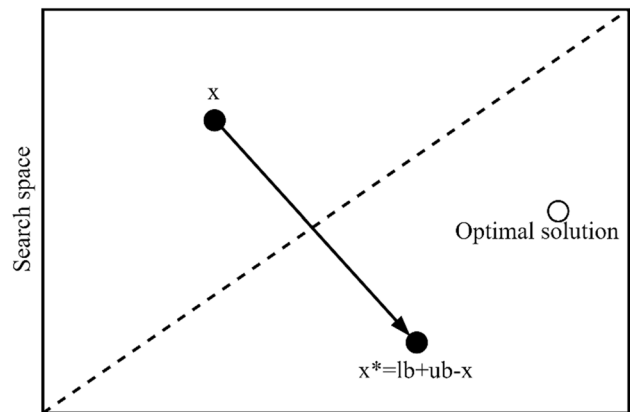


Fig. 1 The OBL concept

**Table 1** Pseudocode of the nSCA

---

**Input:** Population size ( $N$ ) and maximum iteration ( $I_{max}$ )

---

**Begin**  
 Generate randomized solutions;  
***OBL technique for the initial population creation stage***  
 [  
 Create opposite solutions using Eq. (4);  
 Determine the fitness value for each solution;  
 Update the new solution set (with  $N$  solutions);  
 ]  
**while** ( $I_{cur} < I_{max}$ ) **do**  
 Arrange the solutions in order from best to worst;  
 Calculate the normalized fitness score of each solution;  
 Update  $\sigma_1$  using Eq. (8);  
**for** (each solution  $i$ ) **do**  
**for** (each parameter  $j$ ) **do**  
***RWS technique***  
 [  
 Create a random  $\sigma_2$  value;  
 Update solution position by RWS using Eq. (5) and Eq. (6);  
 ]  
***Basic SCA***  
 [  
 Create a random  $\sigma_3$  value;  
 Update solution position using Eq. (7);  
 ]  
***OBL technique for exploitation stage***  
 [  
 Create a random  $\sigma_6$  value;  
 Determine jumping condition  $JC$  using Eq.(9)  
**if**  $\sigma_5 \leq JC$ ;  
 Create opposite solutions using Eq. (10)  
**end;**  
 Update fitness value of each solution;  
 Determine the superior solution using Eq. (11);  
 ]  
**end;**  
**end;**  
 Determine the best solution;  
**end**

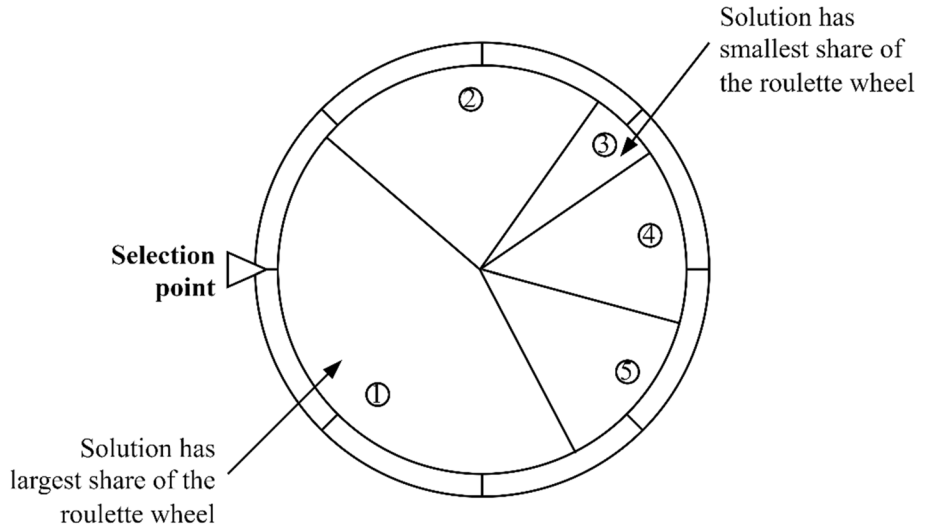
---

**Output:** Best solution and its fitness value.

---



Fig. 2 The RWS concept



$$NF(S_i) = \frac{F(S_i)}{\sqrt{\sum_1^N F(S_i)^2}}, \tag{5}$$

$$s_i^j = \begin{cases} s_1^j \sigma_2 < NF(S_i) \\ s_1^j \sigma_2 \geq NF(S_i) \end{cases}. \tag{6}$$

In Eqs. (5) and (6),  $NF(S_i)$  and  $F(S_i)$  denote the normalized fitness value and the fitness value of the  $i$ th solution,  $S_i$ , respectively. The notation  $s_i^j$  represents the  $j$ th parameter of the  $i$ th solution, while  $s_1^j$  refers to the  $j$ th parameter of the current best-performing solution. The variable  $\sigma_2$  is a random number that falls within the range of 0 to 1.

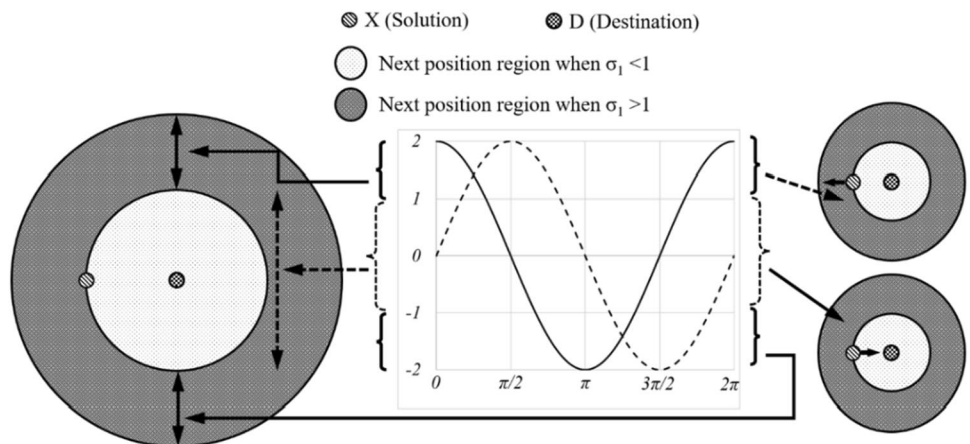
The partitioning of the optimization process into exploration and exploitation phases is a recurring theme in the existing literature, particularly in relation to population-based stochastic algorithms [8]. During the exploration phase, the optimization algorithm utilizes a higher degree of randomness to facilitate the combination of diverse solutions,

swiftly identifying promising areas within the search space. In contrast, the exploitation phase concentrates on the refinement of existing solutions through incremental adjustments, exhibiting significantly reduced levels of stochastic variability relative to the exploration stage. Within the SCA framework, specific mathematical expressions, represented by Eq. (7), govern the updating of agent positions in both exploration and exploitation stages. These equations are pivotal as they guide the search mechanism of the SCA, thereby enabling efficient exploration and targeted exploitation of the search landscape.

$$s_j^{t+1} = \begin{cases} s_j^t + \sigma_1 \times \sin(\sigma_4) \times \left| \sigma_5 P_j^t - s_j^t \right| \sigma_3 < 0.5 \\ s_j^t + \sigma_1 \times \cos(\sigma_4) \times \left| \sigma_5 P_j^t - s_j^t \right| \sigma_3 \geq 0.5 \end{cases}, \tag{7}$$

where  $s_j^t$  represents the position of the solution in the  $j$ th dimension at the  $t$ th iteration;  $\sigma_1$  defines the direction of movement;  $\sigma_3$  is a uniformly distributed random variable ranging between 0 and 1;  $\sigma_4$  serves as a stochastic variable

Fig. 3 The exploration and exploitation mechanisms of the SCA



that regulates the extent of movement toward or away from the target, while  $\sigma_5$  acts as a randomly determined weight for the destination; the position of the target solution in the  $j$ th dimension is denoted by  $D'_j$ , and the absolute value is symbolized by  $\|$ .

Figure 3 presents a detailed model to elucidate the efficacy of sine and cosine functions within the interval  $[-2, 2]$ . These trigonometric functions serve as versatile tools for navigational purposes, either by confining movement within the ranges defined by them or by facilitating extensions beyond these boundaries. Such flexibility is conducive to steering toward the desired objectives effectively. Importantly, the figure delineates the dynamic ranges of the sine and cosine functions, which play a crucial role in updating the positions of potential solutions. Furthermore, Eq. (7) introduces a stochastic variable, denoted as  $\sigma_4$ , with a range between 0 and  $2\pi$ . The inclusion of this stochastic element imbues the algorithm with a degree of randomness, thereby enhancing its exploratory capabilities. This feature allows for a more thorough evaluation of potential solutions within the given search landscape.

During each iteration cycle, the range of the sine and cosine functions, as outlined in Eq. (7), is adaptively modified to achieve a balanced trade-off between exploration and exploitation. This is further illustrated in Fig. 4. This dynamic adjustment is specifically engineered to effectively identify promising regions within the search space, thus facilitating more efficient discovery of the optimal solution. The guidelines for this modification process are set forth in Eq. (8), where the constant  $v$  is designated a value of 2. In this equation,  $I_{cur}$  symbolizes the current iteration count, and  $I_{max}$  represents the maximum number of iterations permitted.

$$\sigma_1 = v - I_{cur} \frac{v}{I_{max}}. \tag{8}$$

In the exploitation stage, as detailed in the pseudocode for nSCA presented in Table 1, solution updates are carried

out in accordance with Eq. (7). Following these updates, a jumping condition, denoted as  $JC$  in Eq. (9), is activated to dynamically generate an opposite solution in accordance with Eq. (10). It is noteworthy that this approach deviates from the methodology employed in the initial phase of population generation. Subsequent to the generation of opposite solutions, the objective function is applied to both the original solutions and the newly formed opposite solutions. The superior solution is retained, while the inferior one is eliminated. This process ensures that the population size remains constant, as mandated by Eq. (11).

$$JC = -\left(\frac{I_{cur}}{I_{max}}\right)^2 + 2\left(\frac{I_{cur}}{I_{max}}\right), \tag{9}$$

$$\text{Createoppositesolution } S_i^* \text{ of } S_i \text{ if } \sigma_6 < JC, \tag{10}$$

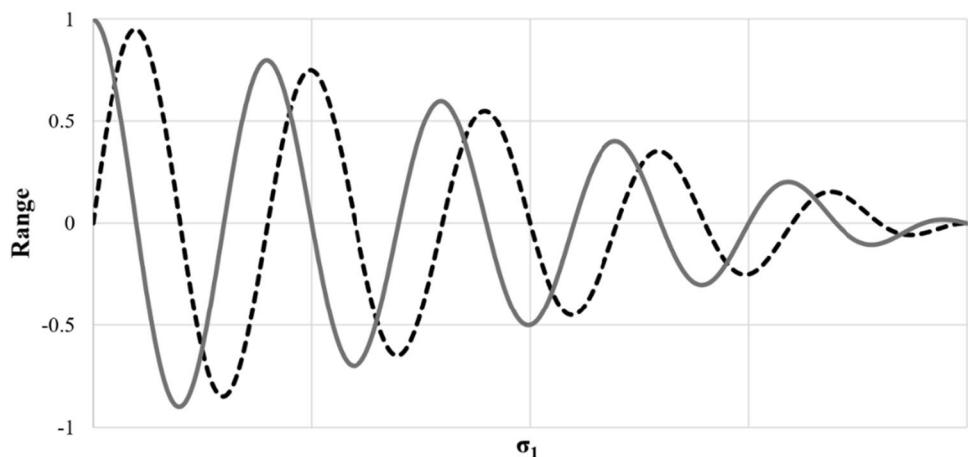
where  $S_i$  represents the  $i$ th solution while  $S_i^*$  represents the opposite solution of the  $i$ th solution created by OBL;  $\sigma_6$  is a uniformly distributed random variable between 0 and 1.

$$S_{new} = \begin{cases} S_i \text{ if } F(S_i) \text{ is superior solution} \\ S_i^* \text{ if } F(S_i^*) \text{ is superior solution} \end{cases}. \tag{11}$$

### 3 Convergence Analysis

In the field of optimization, encompassing the application of evolutionary algorithms and metaheuristics, the validation of algorithmic effectiveness is critically dependent on the use of specialized test cases. This is particularly important given the inherently stochastic nature of these methodologies, where achieving optimal results requires the careful selection of a diverse and appropriate set of test functions. The aim of this section is to evaluate the performance of the nSCA algorithm, as substantiated through its application

**Fig. 4** The range of sine and cosine exhibits a decreasing pattern



to 23 classical test functions, as well as the CEC2017 set. Each of these test functions has unique characteristics, designed to enable an in-depth assessment of the algorithm’s performance.

### 3.1 Convergence Analysis on Classical Benchmark Functions

The efficacy of the nSCA algorithm was rigorously assessed using an extensive set of 23 test functions [43–45]. These

**Table 2** 23 classical benchmark test functions

Type	Function	Dim	Range	fmin
Uni-modal	$f1(x) = \sum_{i=1}^n x_i^2$	10	[-100, 100]	0
Uni-modal	$f2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	10	[-10, 10]	0
Uni-modal	$f3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j\right)^2$	10	[-100, 100]	0
Uni-modal	$f4(x) = \max\{ x_i , 1 \leq i \leq n\}$	10	[-100, 100]	0
Uni-modal	$f5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	10	[-30, 30]	0
Uni-modal	$f6(x) = \sum_{i=1}^n ( x_i + 0.5 )^2$	10	[-100, 100]	0
Uni-modal	$f7(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1)$	10	[-1.28, 1.28]	0
Multi-modal	$f8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	10	[-500, 500]	0
Multi-modal	$f9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	10	[-5.12, 5.12]	0
Multi-modal	$f10(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	10	[-32, 32]	0
Multi-modal	$f11(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	10	[-600, 600]	0
Multi-modal	$f12(x) = \frac{\pi}{n} \{10 \sin^2(\pi y_1) + \sum_{i=1}^n (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 + \sum_{i=1}^n u(x_i, 10, 100, 4)\}$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m x_i > a \\ 0 - a < x_i < a \\ k(-x_i - a)^m x_i < -a \end{cases}$	10	[-50, 50]	0
Multi-modal	$f13(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \right\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	10	[-50, 50]	0
Fixed	$f14(x) = \left( \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$	2	[-65, 65]	1
Fixed	$f15(x) = \sum_{i=1}^{11} \left[ a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[-5, 5]	0.0003
Fixed	$f16(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[-5, 5]	-1.0316
Fixed	$f17(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	[-5, 5]	0.398
Fixed	$f18(x) = \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[-2, 2]	3
Fixed	$f19(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2\right)$	3	[0, 1]	-3.86
Fixed	$f20(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2\right)$	6	[0, 1]	-3.32
Fixed	$f21(x) = -\sum_{i=1}^5 \left[ (X - a_i)(X - a_i)^T + c_i \right]^{-1}$	4	[0, 10]	-10.1532
Fixed	$f22(x) = -\sum_{i=1}^7 \left[ (X - a_i)(X - a_i)^T + c_i \right]^{-1}$	4	[0, 10]	-10.4028
Fixed	$f23(x) = -\sum_{i=1}^{10} \left[ (X - a_i)(X - a_i)^T + c_i \right]^{-1}$	4	[0, 10]	-10.5363



functions were grouped into three distinct categories, as outlined in Table 2: unimodal, multimodal, and fixed functions. The unimodal category consists of functions with a single global optimum and no local optimum, serving as a basis to evaluate the algorithm's capacity for rapid convergence and focused exploitation. In contrast, multimodal functions feature multiple local optima in addition to a global optimum, enabling a thorough assessment of the algorithm's capability to navigate around local optima for effective exploration of the search space. Finally, the fixed category includes modified versions of both unimodal and multimodal functions, which are altered through operations such as rotation, shifting, and bias. These composite functions are designed to

evaluate the algorithm's adaptability and performance in complex optimization landscapes.

To rigorously evaluate the performance capabilities of the nSCA algorithm in optimization tasks, an ensemble of 25 search agents was employed to locate the global optimum within a suite of 23 test functions. This experiment was conducted over a span of 300 iterations. The performance of nSCA was subsequently benchmarked against a selection of leading metaheuristic algorithms, including SSA, MVO, MFO, WOA, GOA, and the original SCA. Due to the stochastic components intrinsic to these algorithms, each was executed 30 times to ensure result reliability. Key statistical metrics, including average values (*avg*) and

**Table 3** Findings of unimodal test functions

Algorithm/ function		<i>f1</i>	<i>f2</i>	<i>f3</i>	<i>f4</i>	<i>f5</i>	<i>f6</i>	<i>f7</i>
nSCA	<i>avg</i>	0.000E+00	0.000E+00	7.397E-20	4.104E-14	6.490E+00	8.300E-02	4.987E-04
	<i>std</i>	0.000E+00	0.000E+00	1.553E-19	3.122E-14	3.414E-01	1.173E-01	1.783E-04
SCA	<i>avg</i>	9.622E-06	2.223E-05	1.442E+00	9.930E-02	1.075E+01	5.542E-01	3.874E-03
	<i>std</i>	1.861E-05	6.164E-05	4.677E+00	1.527E-01	1.338E+01	1.592E-01	3.327E-03
MFO	<i>avg</i>	1.938E-06	6.667E-01	9.506E+02	5.972E+00	6.306E+03	3.639E-06	1.805E-02
	<i>std</i>	3.866E-06	2.494E+00	2.136E+03	7.560E+00	2.238E+04	5.468E-06	1.100E-02
MVO	<i>avg</i>	5.341E-02	6.829E-02	4.659E-01	1.733E-01	1.821E+02	4.938E-02	5.073E-03
	<i>std</i>	2.785E-02	2.342E-02	2.501E-01	6.474E-02	4.293E+02	2.584E-02	3.297E-03
WOA	<i>avg</i>	5.886E-39	1.841E-28	6.534E+04	4.670E+01	2.846E+01	1.124E+00	8.049E-03
	<i>std</i>	2.303E-38	4.220E-28	1.714E+04	2.969E+01	3.106E-01	4.292E-01	9.725E-03
SSA	<i>avg</i>	1.643E-09	1.146E-01	6.111E-01	1.387E-02	1.649E+02	1.662E-09	3.515E-02
	<i>std</i>	7.488E-10	3.335E-01	1.310E+00	3.211E-02	2.165E+02	7.119E-10	2.286E-02
GOA	<i>avg</i>	1.951E-07	2.091E+00	5.783E-03	1.148E-03	2.641E+03	1.081E-07	7.726E-02
	<i>std</i>	3.497E-07	2.780E+00	2.836E-02	3.726E-03	1.110E+04	1.123E-07	1.209E-01

**Table 4** Findings of multi-modal test functions

Algo- rithm/ function		<i>f8</i>	<i>f9</i>	<i>f10</i>	<i>f11</i>	<i>f12</i>	<i>f13</i>
nSCA	<i>avg</i>	-3.038E+03	0.000E+00	8.882E-16	0.000E+00	2.502E-02	1.663E-01
	<i>std</i>	2.271E+02	0.000E+00	9.861E-32	0.000E+00	8.652E-03	5.582E-02
SCA	<i>avg</i>	-2.070E+03	2.422E+00	8.854E-02	1.191E-01	1.537E-01	3.442E-01
	<i>std</i>	1.371E+02	4.851E+00	4.696E-01	1.271E-01	6.262E-02	1.092E-01
MFO	<i>avg</i>	-3.194E+03	2.482E+01	1.220E+00	1.437E-01	1.465E-01	4.769E-03
	<i>std</i>	4.049E+02	1.122E+01	3.689E+00	8.912E-02	4.153E-01	5.449E-03
MVO	<i>avg</i>	-2.874E+03	1.979E+01	3.408E-01	4.563E-01	1.537E-01	1.697E-02
	<i>std</i>	3.771E+02	8.600E+00	5.965E-01	1.600E-01	3.025E-01	9.071E-03
WOA	<i>avg</i>	-9.375E+03	3.780E+00	7.046E-15	8.115E-03	5.758E-02	1.016E+00
	<i>std</i>	1.908E+03	2.035E+01	3.991E-15	4.370E-02	2.799E-02	4.184E-01
SSA	<i>avg</i>	-2.689E+03	1.691E+01	1.023E+00	1.773E-01	1.742E+00	3.580E-03
	<i>std</i>	4.119E+02	7.542E+00	9.615E-01	1.109E-01	2.077E+00	5.472E-03
GOA	<i>avg</i>	-1.553E+03	1.238E+01	9.102E-01	1.951E-01	9.537E-02	3.980E-03
	<i>std</i>	2.531E+02	5.821E+00	1.124E+00	1.158E-01	2.005E-01	5.381E-03

**Table 5** Findings of composite test functions

Algorithm/ function		<i>f14</i>	<i>f15</i>	<i>f16</i>	<i>f17</i>	<i>f18</i>
nSCA	<i>avg</i>	9.980E-01	4.877E-04	-1.032E+00	3.979E-01	3.000E+00
	<i>std</i>	2.701E-10	1.153E-04	1.296E-08	3.869E-07	8.893E-07
SCA	<i>avg</i>	2.130E+00	9.769E-04	-1.032E+00	4.033E-01	3.000E+00
	<i>std</i>	1.863E+00	3.634E-04	1.215E-04	8.638E-03	6.540E-04
MFO	<i>avg</i>	2.544E+00	2.052E-03	-1.032E+00	3.979E-01	3.000E+00
	<i>std</i>	2.189E+00	3.748E-03	0.000E+00	1.110E-16	3.347E-15
MVO	<i>avg</i>	1.031E+00	5.467E-03	-1.032E+00	3.979E-01	8.400E+00
	<i>std</i>	1.784E-01	8.224E-03	1.438E-06	2.104E-06	2.020E+01
WOA	<i>avg</i>	3.752E+00	6.887E-04	-1.032E+00	3.979E-01	3.001E+00
	<i>std</i>	3.082E+00	3.874E-04	6.999E-08	6.024E-05	1.510E-03
SSA	<i>avg</i>	1.659E+00	4.414E-03	-1.032E+00	4.026E-01	3.000E+00
	<i>std</i>	9.677E-01	7.240E-03	8.968E-14	4.462E-03	4.017E-13
GOA	<i>avg</i>	3.057E+00	1.032E-02	-1.032E+00	3.979E-01	5.700E+00
	<i>std</i>	3.399E+00	2.029E-02	2.306E-12	4.712E-12	1.454E+01

**Table 6** Findings of composite test functions (continued)

Algorithm/ function		<i>f19</i>	<i>f20</i>	<i>f21</i>	<i>f22</i>	<i>f23</i>
nSCA	<i>avg</i>	-3.862E+00	-3.199E+00	-8.800E+00	-8.989E+00	-8.825E+00
	<i>std</i>	1.513E-03	7.798E-02	2.495E+00	2.586E+00	2.887E+00
SCA	<i>avg</i>	-3.853E+00	-2.942E+00	-1.805E+00	-2.136E+00	-3.131E+00
	<i>std</i>	3.664E-03	2.013E-01	1.696E+00	1.431E+00	1.469E+00
MFO	<i>avg</i>	-3.863E+00	-3.217E+00	-6.308E+00	-6.529E+00	-7.338E+00
	<i>std</i>	2.665E-15	8.801E-02	3.465E+00	3.267E+00	3.698E+00
MVO	<i>avg</i>	-3.863E+00	-3.263E+00	-6.116E+00	-8.140E+00	-7.891E+00
	<i>std</i>	7.066E-06	6.360E-02	2.796E+00	3.280E+00	3.332E+00
WOA	<i>avg</i>	-3.848E+00	-3.208E+00	-7.436E+00	-7.472E+00	-7.485E+00
	<i>std</i>	3.527E-02	9.835E-02	2.601E+00	3.042E+00	3.003E+00
SSA	<i>avg</i>	-3.863E+00	-3.223E+00	-7.570E+00	-6.917E+00	-8.287E+00
	<i>std</i>	8.404E-06	7.535E-02	3.441E+00	3.765E+00	3.471E+00
GOA	<i>avg</i>	-3.538E+00	-3.265E+00	-4.813E+00	-6.006E+00	-5.225E+00
	<i>std</i>	3.546E-01	6.617E-02	3.075E+00	3.436E+00	3.364E+00

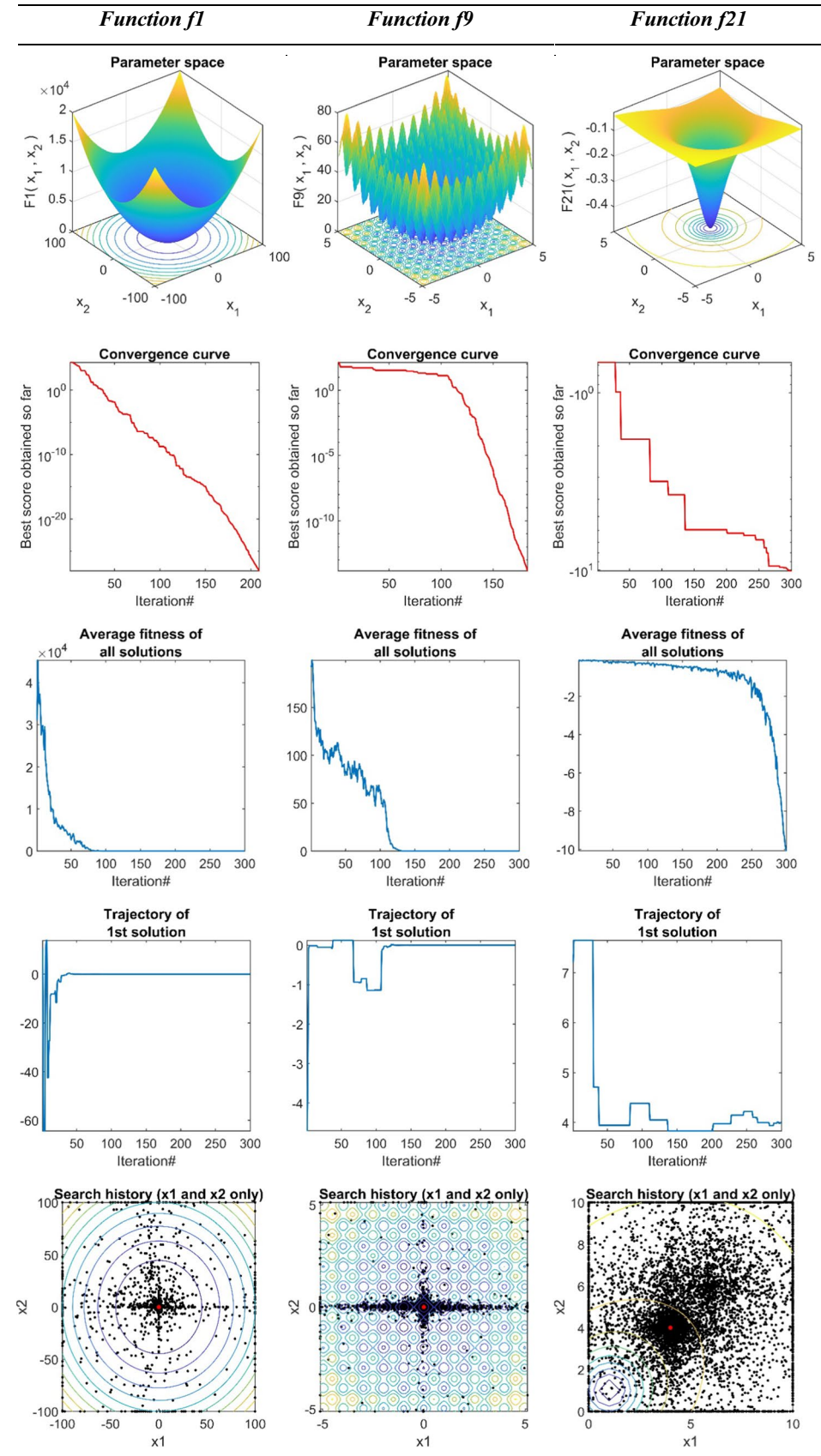
standard deviations (*std*), were calculated, and are presented in Tables 3, 4, 5 and 6. This comprehensive approach provides valuable insights into the comparative effectiveness of nSCA and other algorithms in optimization contexts.

In the realm of unimodal test functions, as evidenced by the results in Table 3, nSCA holds a marked advantage over its competitors. Specifically, within the scope of unimodal optimization, nSCA's exploitation capabilities surpass those of SCA, MFO, MVO, WOA, SSA, and GOA in most test functions. This data effectively emphasizes nSCA's proficiency in handling unimodal optimization challenges. Regarding multimodal optimization, Table 4 provides data that confirm nSCA's superior performance over SCA, MFO, MVO, WOA, SSA, and GOA in most test cases. This impressive showing reinforces nSCA's capabilities in

effectively navigating complex search spaces and avoiding local optima. Lastly, when examined in the context of fixed test functions, nSCA shows performance metrics that are on par with those of SCA, WOA, MVO, SSA, GOA, and MFO, as illustrated in Tables 5 and 6. These results lend further support to nSCA's considerable versatility and competitive edge when compared to other state-of-the-art optimization algorithms.

Additional performance metrics such as the convergence curve, average solution fitness, trajectory of the first solution, and search history were scrutinized to provide a more nuanced assessment of nSCA's effectiveness. The study employed a configuration of 300 iterations and 25 search agents to examine three representative test functions (*f1*, *f9*, and *f21*). Each of these functions represents a different

**Fig. 5** Convergence curve, average fitness of all solutions, trajectory of the first solution, and search histories of functions f1, f9 and f21



category: unimodal, multimodal, and composite, as depicted in Fig. 5. Analysis of the convergence curve and average fitness reveals a consistent improvement in the quality of the search agents over successive iterations. This observation underscores nSCA's capability to enhance the quality of initially randomized solutions in specific optimization tasks.

Analysis of the trajectory of the first solution underscores nSCA's abilities in both convergence and local search optimization. This is supported by the notable fluctuations in average fitness levels during the exploration phase and the relatively stable metrics seen in the exploitation stage, as cited in reference [46]. Further, the search histories associated with functions  $f1$ ,  $f9$ , and  $f21$  substantiate nSCA's aptitude for identifying and concentrating on high-potential regions within the search space. The incorporation of RWS and OBL mechanisms proves to be beneficial, facilitating initial exploration and contributing to the ultimate convergence of optimal solutions initially identified during the exploration phase.

Figures 6, 7 and 8 display the convergence patterns for the 23 test functions, obtained over 150 iterations employing 25 search agents. The findings suggest that more efficient convergence for the majority of the test functions analyzed is achieved by the nSCA in comparison to other algorithms such as the original SCA, MVO, MFO, SSA, GOA, and WOA.

### 3.2 CEC2017 Benchmark Test Functions

The CEC2017 test functions constitute a specialized set of benchmarks, introduced at the 2017 IEEE Congress on Evolutionary Computation (CEC), focusing on the optimization of real parameters. Building upon the groundwork established by previous benchmark suites, the CEC2017 collection is designed to present a diverse range of challenges to optimization algorithms. These functions are generally considered to provide more realistic problem scenarios in comparison to the traditional set of 23 benchmark functions.

Spanning both unimodal and multimodal optimization landscapes, the CEC2017 suite also encompasses separable and non-separable problem domains. Moreover, it incorporates shifted and rotated variations, thereby offering a comprehensive environment for the testing of optimization algorithms. This extensive array of test scenarios enables researchers to conduct in-depth evaluations, thereby discerning the merits and limitations of various optimization methods under different conditions.

The efficacy of nSCA is evaluated using the IEEE CEC2017 benchmark suites [47]. These test functions are categorized into four distinct groups: unimodal, multimodal, hybrid, and composition. Table 7 offers a comprehensive breakdown of the definitions associated with the CEC2017 benchmark challenges. To increase the level of

complexity and rigorously assess the capabilities of the proposed method in handling complex optimization problems, all functions within the CEC2017 suite are configured as 30-dimensional problems.

Tables 8 and 9 provide an in-depth statistical comparison between nSCA and other swarm-based optimization algorithms such as SSA, MVO, MFO, WOA, GOA, and the original SCA. To ensure a rigorous and unbiased evaluation, each algorithm was executed 30 times on a variety of benchmark functions. Statistical metrics like mean values (*avg*) and standard deviations (*std*) were subsequently calculated from these multiple runs. For the purposes of this study, a cohort of 50 search agents was deployed, each limited to a maximum of 300 iterations. A careful analysis of the data presented in Tables 8 and 9 clearly shows that nSCA consistently outperforms its counterparts, specifically SSA, MVO, MFO, WOA, GOA, and the original SCA, in various benchmark categories including unimodal, multimodal, hybrid, and composition functions.

## 4 Engineering Optimization Challenges

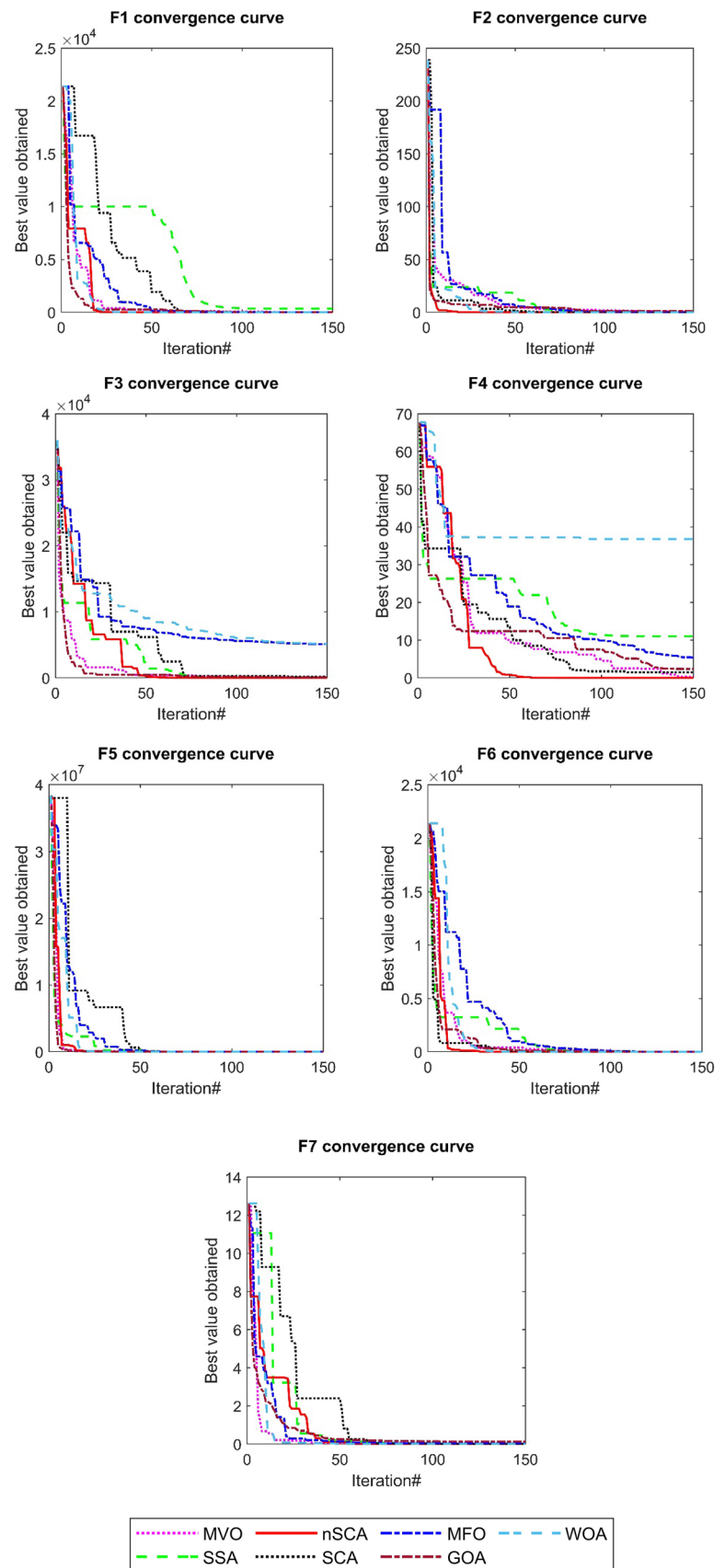
The purpose of this section is to assess the performance of nSCA as evidenced through its deployment in five real-world technical optimization problems, each characterized by varying inequality constraints. The primary focus lies in evaluating the capability of the algorithm to manage these constraints effectively throughout the optimization process.

### 4.1 Cantilever Beam Design Challenge

The objective of this optimization task is to achieve minimization of the weight of a cantilever beam, which is constructed from hollow square blocks. The structure consists of five such blocks, with the first block being fixed in position and the fifth subjected to a vertical load. A visual representation of the five parameters that determine the cross-sectional geometry of the blocks is provided in Fig. 9. Detailed formulations for addressing this problem can be found in Appendix 1.

The findings from an exhaustive analysis of this task are summarized in Table 10, which provides a comprehensive breakdown of key performance indicators. The data convincingly demonstrate that the nSCA algorithm consistently yields results that are either commensurate with or superior to those of leading optimization algorithms such as COA [52], RFO [51], GOA [6], MVO [1], ALO [50], CS [48] and SOS [49]. These findings strongly substantiate the algorithm's capability to address and optimize complex, constraint-bound problems effectively. Additionally, the results underscore the algorithm's aptitude for real-world

**Fig. 6** Convergence behavior of nSCA, SCA, SSA, MVO, MFO, WOA, and GOA for unimodal test functions





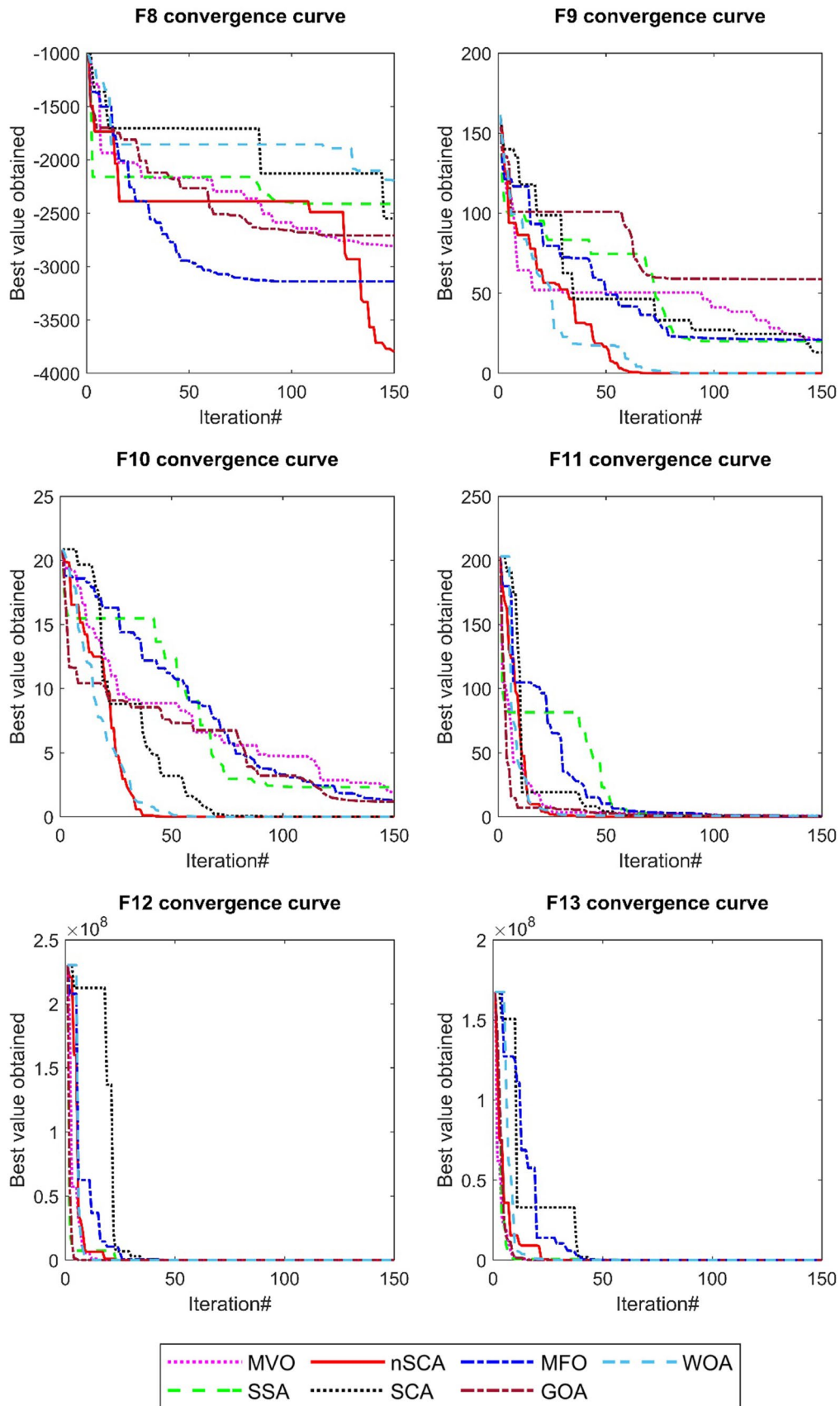


Fig. 7 Convergence behavior of nSCA, SCA, SSA, MVO, MFO, WOA, and GOA for multimodal test functions



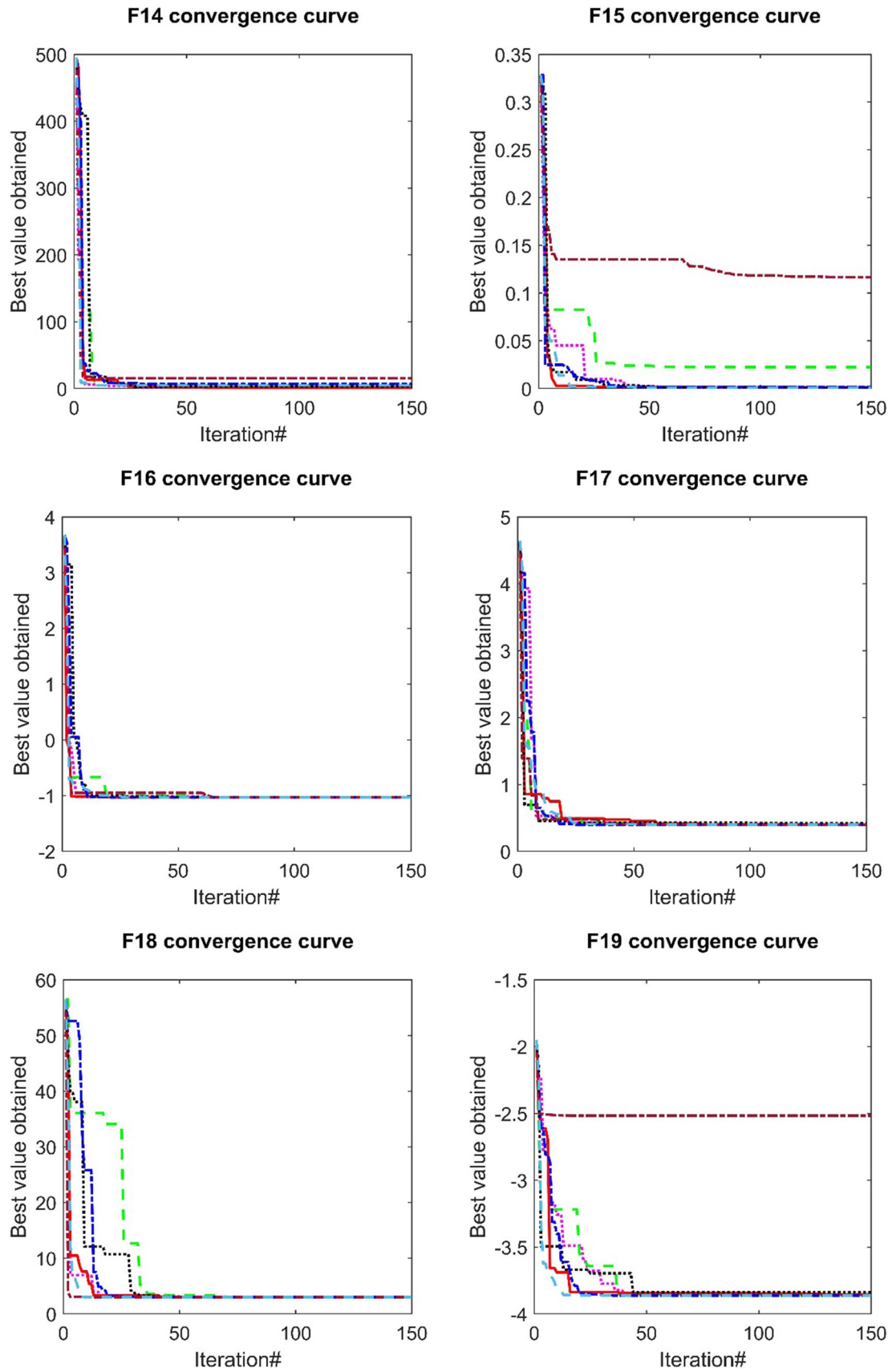


Fig. 8 Convergence behavior of nSCA, SCA, SSA, MVO, MFO, WOA, and GOA for composite test functions

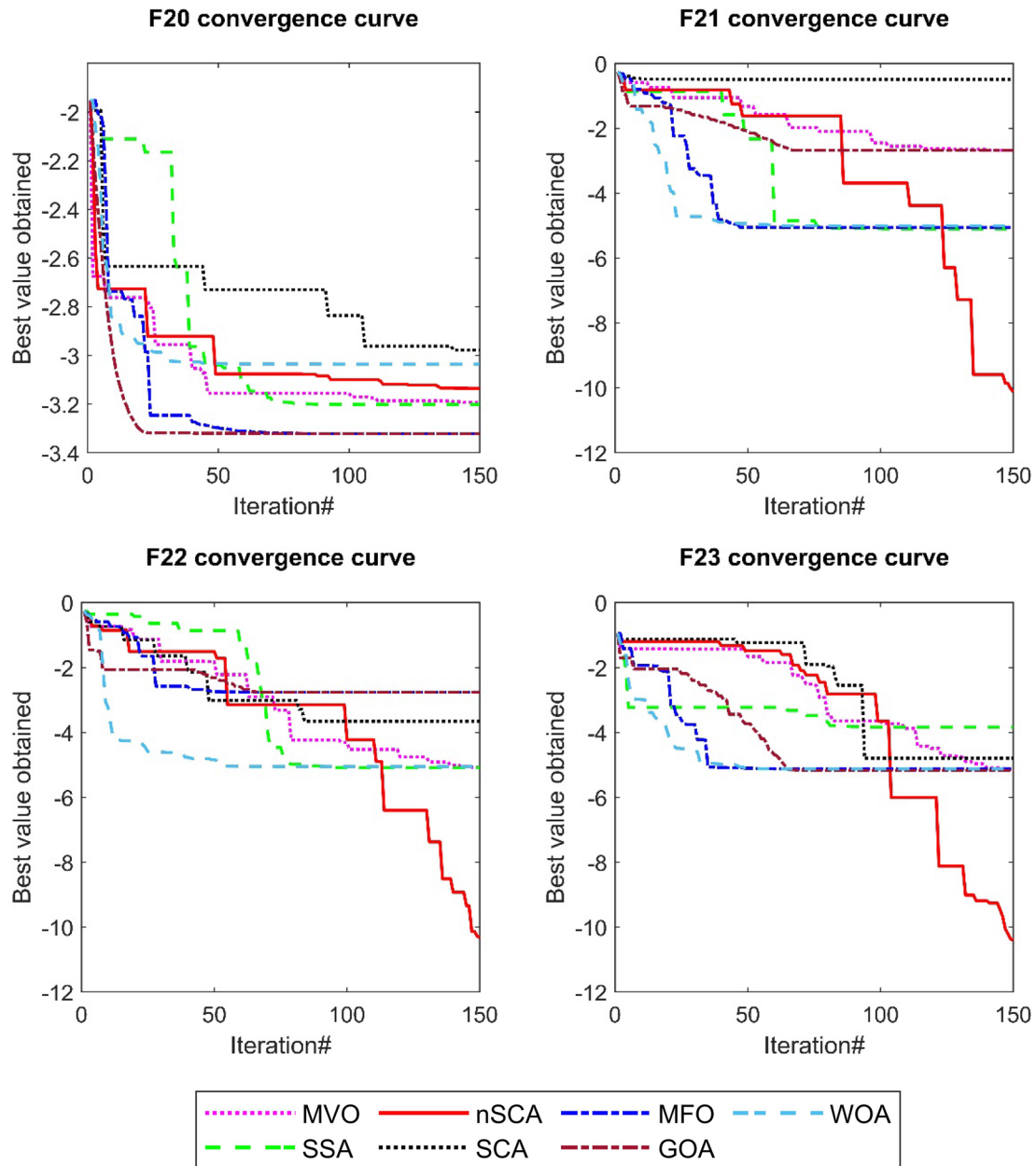


Fig. 8 (continued)

engineering applications, highlighting its proficiency in navigating intricate problem landscapes.

### 4.2 Pressure Vessel Design Challenge

The primary objective of this optimization task is the reduction of manufacturing costs associated with the fabrication of a pressure vessel. A representation of the vessel’s unique design, featuring one flat and one hemispherical end, is illustrated in Fig. 10. The variables subject to optimization encompass the inner radius ( $R$ ), shell thickness ( $T_s$ ), length of the cylindrical section exclusive of the head ( $L$ ),

and the head's thickness ( $T_h$ ). These variables are pivotal in establishing the optimal design of the vessel. Specific mathematical equations and constraints have been formulated to encapsulate the dual aim of cost minimization and design requirement adherence. Comprehensive formulations for this task can be found in Appendix 1.

The outcomes of a comprehensive evaluation of this problem are summarized in Table 11, which offers a detailed analysis of various performance metrics. The data presented in this table confirm the reliable effectiveness of the nSCA algorithm, often matching or even surpassing other well-established optimization methods such as SCSO [58], RFO

**Table 7** CEC2017 benchmark functions

Type	Function	Name	n	Range	fmin
Unimodal	<i>F1</i>	Shifted and Rotated Bent Cigar Function	30	[− 100, 100]	100
Unimodal	<i>F2</i>	Shifted and Rotated Zakharov Function	30	[− 100, 100]	200
Multimodal	<i>F3</i>	Shifted and Rotated Rosenbrock's Function	30	[− 100, 100]	300
Multimodal	<i>F4</i>	Shifted and Rotated Rastrigin's Function	30	[− 100, 100]	400
Multimodal	<i>F5</i>	Shifted and Rotated Expanded Scaffer's <i>F7</i> Function	30	[− 100, 100]	500
Multimodal	<i>F6</i>	Shifted and Rotated Lunacek Bi_Rastrigin Function	30	[− 100, 100]	600
Multimodal	<i>F7</i>	Shifted and Rotated Non-Continuous Rastrigin's Function	30	[− 100, 100]	700
Multimodal	<i>F8</i>	Shifted and Rotated Levy Function	30	[− 100, 100]	800
Multimodal	<i>F9</i>	Shifted and Rotated Schwefel's Function	30	[− 100, 100]	900
Hybrid	<i>F10</i>	Hybrid Function 1 ( $N=3$ )	30	[− 100, 100]	1000
Hybrid	<i>F11</i>	Hybrid Function 2 ( $N=3$ )	30	[− 100, 100]	1100
Hybrid	<i>F12</i>	Hybrid Function 3 ( $N=3$ )	30	[− 100, 100]	1200
Hybrid	<i>F13</i>	Hybrid Function 4 ( $N=4$ )	30	[− 100, 100]	1300
Hybrid	<i>F14</i>	Hybrid Function 5 ( $N=4$ )	30	[− 100, 100]	1400
Hybrid	<i>F15</i>	Hybrid Function 6 ( $N=4$ )	30	[− 100, 100]	1500
Hybrid	<i>F16</i>	Hybrid Function 7 ( $N=5$ )	30	[− 100, 100]	1600
Hybrid	<i>F17</i>	Hybrid Function 8 ( $N=5$ )	30	[− 100, 100]	1700
Hybrid	<i>F18</i>	Hybrid Function 9 ( $N=5$ )	30	[− 100, 100]	1800
Hybrid	<i>F19</i>	Hybrid Function 10 ( $N=6$ )	30	[− 100, 100]	1900
Composition	<i>F20</i>	Composition Function 1 ( $N=3$ )	30	[− 100, 100]	2000
Composition	<i>F21</i>	Composition Function 2 ( $N=3$ )	30	[− 100, 100]	2100
Composition	<i>F22</i>	Composition Function 3 ( $N=4$ )	30	[− 100, 100]	2200
Composition	<i>F23</i>	Composition Function 4 ( $N=4$ )	30	[− 100, 100]	2300
Composition	<i>F24</i>	Composition Function 5 ( $N=5$ )	30	[− 100, 100]	2400
Composition	<i>F25</i>	Composition Function 6 ( $N=5$ )	30	[− 100, 100]	2500
Composition	<i>F26</i>	Composition Function 7 ( $N=6$ )	30	[− 100, 100]	2600
Composition	<i>F27</i>	Composition Function 8 ( $N=6$ )	30	[− 100, 100]	2700
Composition	<i>F28</i>	Composition Function 9 ( $N=3$ )	30	[− 100, 100]	2800
Composition	<i>F29</i>	Composition Function 10 ( $N=3$ )	30	[− 100, 100]	2900

[51], AOA [57], GSA [1], MVO [1], ACO [56], ES [55], DE [54], and PSO [53]. These results robustly endorse the capabilities of nSCA in proficiently navigating the search space, an ability further augmented by the integration of roulette wheel selection (RWS) and opposition-based learning (OBL). Additionally, the findings underscore the algorithm's versatility, demonstrating its suitability for application in engineering contexts, particularly in instances where the attributes of the search domain are either ambiguous or poorly defined.

### 4.3 Three-Bar Truss Design Challenge

The primary objective of this challenge is the weight reduction of the truss structure, to be achieved within the boundaries of various constraints. Successful truss design necessitates the consideration of essential limitations, including those related to stress, deflection, and buckling factors. The engineering characteristics pertinent to this issue are

illustrated in Fig. 11. Although the objective function may appear straightforward, it is governed by multiple intricate constraints, rendering the achievement of an optimal solution notably challenging. Detailed formulations relevant to this problem are provided in Appendix 1.

Table 12 provides an exhaustive comparison between the nSCA and various state-of-the-art optimization methods, including GOA [6], MVO [1], ALO [50], MBA [63], CS [48], PSO-DE [62], DEDS [61], as well as models put forth by Ray and Saini [61] and Tsai [62]. The data strongly suggest that nSCA consistently performs at a level comparable to the best algorithms in the field, thereby establishing itself as a formidable competitor in achieving optimal outcomes.

### 4.4 Gear Train Design Challenge

The objective of this technical task, illustrated in Fig. 12, is the minimization of the gear ratio through the optimization of four discrete variables: the tooth counts on gears  $n_A$ ,

**Table 8** Results of different algorithms on CEC 2017 test functions

Algorithm/ function	nSCA		SCA		MFO		MVO	
	avg	std	avg	std	avg	std	avg	std
$f_1$	1.673E+06	5.238E+05	2.476E+09	5.951E+08	8.998E+10	1.364E+10	7.955E+09	7.955E+09
$f_2$	2.033E+02	1.264E+00	2.223E+04	3.760E+03	1.818E+05	2.033E+02	2.024E+04	2.024E+04
$f_3$	3.410E+02	1.981E+01	6.730E+02	5.558E+01	2.631E+04	3.410E+02	1.230E+03	1.230E+03
$f_4$	6.406E+02	4.522E+01	3.529E+03	5.403E+02	9.054E+04	6.406E+02	1.122E+04	1.122E+04
$f_5$	5.000E+02	4.236E-04	5.000E+02	1.991E-03	5.000E+02	5.000E+02	5.000E+02	5.000E+02
$f_6$	7.056E+03	4.454E+03	2.505E+04	7.862E+03	7.053E+04	7.056E+03	2.392E+04	2.392E+04
$f_7$	7.001E+02	1.709E-01	7.008E+02	2.173E-01	7.053E+02	7.001E+02	7.002E+02	7.002E+02
$f_8$	8.096E+02	4.336E+00	8.036E+02	9.177E-01	8.778E+02	8.096E+02	8.138E+02	8.138E+02
$f_9$	3.903E+03	7.371E+02	7.253E+03	3.487E+02	8.068E+03	3.903E+03	5.246E+03	5.246E+03
$f_{10}$	6.458E+04	3.440E+04	1.017E+05	1.532E+04	3.610E+07	6.458E+04	1.124E+06	1.124E+06
$f_{11}$	1.321E+07	1.133E+07	1.063E+08	3.438E+07	9.059E+09	1.321E+07	5.156E+08	5.156E+08
$f_{12}$	7.148E+05	3.690E+05	3.980E+07	1.212E+07	9.857E+09	7.148E+05	4.184E+08	4.184E+08
$f_{13}$	2.372E+05	1.577E+05	4.551E+05	2.070E+05	9.227E+06	2.372E+05	1.663E+06	1.663E+06
$f_{14}$	3.103E+05	1.236E+05	6.645E+06	3.256E+06	5.068E+09	3.103E+05	2.501E+08	2.501E+08
$f_{15}$	2.120E+04	1.393E+04	3.634E+05	4.336E+05	1.822E+09	2.120E+04	1.110E+07	1.110E+07
$f_{16}$	1.898E+04	1.139E+04	5.566E+04	1.777E+04	2.214E+12	1.898E+04	6.087E+04	6.087E+04
$f_{17}$	1.077E+05	2.545E+04	1.534E+05	9.205E+04	5.207E+06	1.077E+05	1.040E+06	1.040E+06
$f_{18}$	1.231E+05	5.499E+04	4.547E+08	4.226E+08	5.496E+13	1.231E+05	1.007E+10	1.007E+10
$f_{19}$	2.521E+03	3.037E+02	3.348E+03	3.244E+02	1.986E+04	2.521E+03	3.242E+03	3.242E+03
$f_{20}$	2.370E+03	1.358E+02	4.111E+03	5.749E+02	7.309E+04	2.370E+03	6.899E+03	6.899E+03
$f_{21}$	2.276E+03	4.110E+00	2.371E+03	1.468E+01	3.603E+03	2.276E+03	2.410E+03	2.410E+03
$f_{22}$	2.580E+03	1.023E+02	9.680E+03	9.584E+02	4.918E+04	2.580E+03	1.512E+04	1.512E+04
$f_{23}$	2.637E+03	8.176E+01	6.396E+03	6.000E+02	2.938E+04	2.637E+03	1.025E+04	1.025E+04
$f_{24}$	2.825E+03	1.424E+01	3.068E+03	3.961E+01	1.338E+04	2.825E+03	3.174E+03	3.174E+03
$f_{25}$	3.336E+03	1.447E+01	3.712E+03	7.921E+01	3.558E+03	3.336E+03	3.672E+03	3.672E+03
$f_{26}$	3.129E+03	2.173E+01	3.345E+03	3.687E+01	3.339E+03	3.129E+03	3.258E+03	3.258E+03
$f_{27}$	2.884E+03	1.855E+02	3.254E+03	3.602E+01	4.566E+03	2.884E+03	3.418E+03	3.418E+03
$f_{28}$	6.838E+04	2.680E+04	2.513E+08	2.280E+08	1.260E+12	6.838E+04	2.028E+08	2.028E+08
$f_{29}$	4.549E+06	1.069E+07	1.692E+08	1.331E+08	1.876E+11	4.549E+06	7.344E+08	7.344E+08

$n_B$ ,  $n_C$  and  $n_D$ . The gear ratio is utilized as a measure of the relationship between the angular speeds of the output and input shafts. Incrementation by units of one characterizes these discrete variables. Emphasis in the problem formulation is placed on establishing constraints for the permissible range of these variables. Detailed specifications related to this challenge are delineated in Appendix 1.

Table 13 presents an in-depth comparison between the nSCA and a range of well-known optimization techniques. The data in this table highlight a remarkable similarity in the performance of nSCA to that of leading optimization methods, including MVO [1], ISA [66], CS [48], MBA [63], ABC [63], as well as models developed by Deb and Goyal [65] and Kannan and Kramer [64]. These results strongly affirm the effectiveness of the proposed nSCA algorithm, demonstrating its capabilities even when faced with challenges involving discrete variables. The proficiency of nSCA in managing discrete variables expands its range of

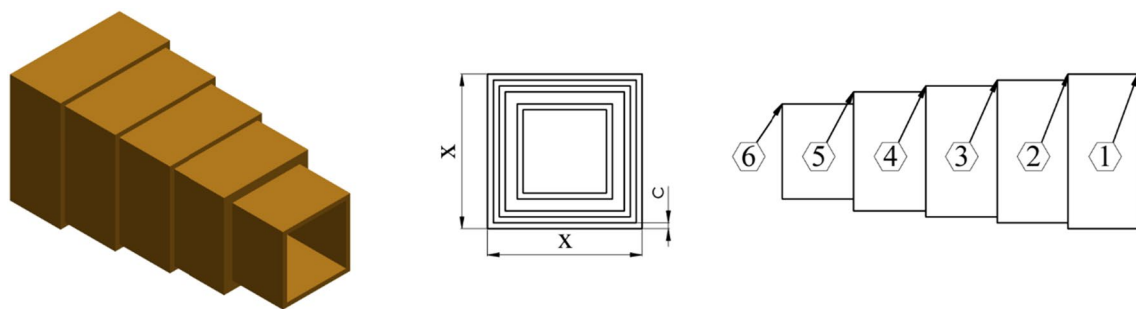
applicability and emphasizes its suitability for addressing a diverse array of optimization problems across various disciplines.

#### 4.5 Welded Beam Design Challenge

The overarching aim of this engineering task is the minimization of manufacturing costs associated with a welded beam. An overview of the system and structural parameters relevant to this challenge is provided in Fig. 13, emphasizing four principal design variables: the length of the attached bar ( $l$ ), weld thickness ( $h$ ), the thickness of the bar ( $b$ ), and the height of the bar ( $t$ ). For the design to be considered feasible, the beam must satisfy seven specific constraints when subjected to a top-applied load. These constraints encompass various factors, such as side constraints, end deflection of the beam ( $\delta$ ), shear stress ( $\tau$ ), bending stress in the beam ( $\theta$ ), and the buckling load

**Table 9** Results of different algorithms on CEC 2017 test functions (continued)

Algorithm/ function	WOA		SSA		GOA	
	<i>avg</i>	<i>std</i>	<i>avg</i>	<i>std</i>	<i>avg</i>	<i>std</i>
$f_1$	1.173E+10	4.452E+09	3.071E+08	2.929E+08	2.956E+10	9.237E+09
$f_2$	3.673E+04	8.689E+03	1.784E+04	1.657E+04	5.578E+04	1.473E+04
$f_3$	1.715E+03	6.542E+02	1.908E+03	2.571E+03	4.670E+03	2.149E+03
$f_4$	2.681E+04	7.889E+03	1.316E+04	4.742E+03	4.233E+04	1.131E+04
$f_5$	5.000E+02	5.886E-03	5.000E+02	6.230E-03	5.000E+02	5.350E-03
$f_6$	8.931E+03	5.754E+03	5.636E+03	5.043E+03	2.185E+04	1.046E+04
$f_7$	7.007E+02	4.485E-01	7.003E+02	3.207E-01	7.012E+02	7.512E-01
$f_8$	8.223E+02	8.944E+00	8.154E+02	5.883E+00	8.267E+02	9.067E+00
$f_9$	6.227E+03	1.046E+03	6.795E+03	1.656E+03	6.475E+03	7.326E+02
$f_{10}$	2.917E+05	7.841E+05	3.683E+06	1.335E+07	1.764E+05	5.627E+04
$f_{11}$	5.249E+07	6.431E+07	1.578E+09	2.915E+09	1.088E+09	1.019E+09
$f_{12}$	1.407E+08	1.761E+08	6.667E+08	2.042E+09	1.329E+09	1.832E+09
$f_{13}$	1.601E+06	1.632E+06	6.281E+06	8.579E+06	3.452E+05	4.985E+05
$f_{14}$	2.814E+07	1.275E+08	1.133E+08	3.035E+08	1.720E+08	2.098E+08
$f_{15}$	2.640E+06	6.244E+06	4.074E+05	7.356E+05	7.325E+05	1.125E+06
$f_{16}$	8.209E+04	4.352E+04	1.957E+13	7.809E+13	8.121E+04	4.987E+04
$f_{17}$	8.981E+04	3.157E+04	1.227E+06	3.242E+06	7.440E+04	2.249E+04
$f_{18}$	1.514E+09	2.366E+09	2.108E+08	1.027E+09	6.913E+09	7.999E+09
$f_{19}$	9.784E+03	2.739E+03	1.288E+04	4.114E+03	7.149E+03	2.670E+03
$f_{20}$	1.634E+04	7.044E+03	1.209E+04	6.494E+03	2.574E+04	1.137E+04
$f_{21}$	3.564E+03	6.592E+02	3.699E+03	6.781E+02	3.834E+03	1.109E+03
$f_{22}$	2.580E+04	1.206E+04	2.173E+04	1.507E+04	3.682E+04	1.030E+04
$f_{23}$	1.565E+04	6.379E+03	1.137E+04	8.621E+03	2.315E+04	6.395E+03
$f_{24}$	3.434E+03	2.334E+02	3.956E+03	9.404E+02	4.441E+03	8.919E+02
$f_{25}$	4.227E+03	6.850E+02	8.381E+03	2.883E+03	4.084E+03	1.198E+03
$f_{26}$	3.778E+03	2.764E+02	4.033E+03	3.799E+02	3.462E+03	2.093E+02
$f_{27}$	3.482E+03	2.881E+02	4.784E+03	1.483E+03	4.138E+03	5.127E+02
$f_{28}$	2.356E+08	4.259E+08	8.230E+11	3.834E+12	4.714E+08	4.628E+08
$f_{29}$	6.502E+08	8.684E+08	3.880E+09	7.263E+09	2.369E+09	2.520E+09



**Fig. 9** Cantilever beam design challenge

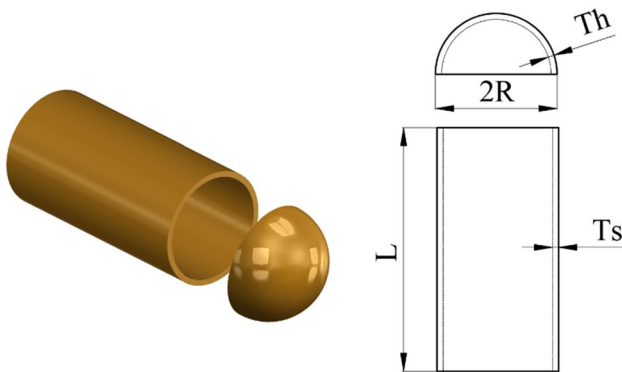
on the bar ( $P_c$ ). Comprehensive formulations pertinent to this task are outlined in Appendix 1.

Table 14 presents a comprehensive comparison between the nSCA and various other cutting-edge optimization techniques. The findings presented in the table offer compelling

evidence that the nSCA consistently achieves superior outcomes when juxtaposed with established algorithms, including SSA [68], RFO [51], MVO [1], GSA [1], CPSO [1], HS [53], and GA [67]. The outcomes elucidated in Table 14 distinctly illustrate that the nSCA proficiently identifies optimal

**Table 10** Comparison findings of cantilever beam design challenge

Optimization technique	Optimal parameters					Optimal weight
	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	
CS [48]	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999
SOS [49]	6.01878	5.30344	4.49587	3.49896	2.15564	1.33996
ALO [50]	6.01812	5.31142	4.4883	3.49751	3.49751	3.49751
MVO [1]	6.023940221548	5.30601123355	4.4950113234	3.4960223242	2.15272617	1.3399595
GOA [6]	6.011674	5.31297	4.48307	3.50279	2.16333	1.33996
RFO [51]	6.00845	5.30485	4.49215	3.4984	2.14463	1.334954
COA [52]	6.017257314	5.307150983	4.491255551	3.508156789	2.149913022	1.33996
nSCA (This study)	5.944606	4.865280	4.503500	3.492579	2.134620	1.303342



**Fig. 10** Pressure vessel design challenge

solutions even within the confines of complex constrained challenges.

The remarkable performance of the nSCA in effectively navigating intricate problem spaces serves to underscore its potential in addressing practical engineering applications marked by multifaceted and intricate constraints. This further underscores the significant role

that the nSCA plays as a valuable instrument within the domain of engineering optimization. Its capabilities offer promising avenues for the enhancement of problem-solving strategies and the facilitation of effective decision-making processes.

### 5 Conclusion

This study introduces an innovative approach that synergistically merges the roulette wheel selection (RWS) mechanism with opposition-based learning (OBL) to enhance the efficacy of the sine cosine algorithm (SCA) in navigating intricate search spaces. This integration gives rise to a novel iteration of the SCA, referred to as nSCA. The comprehensive assessment of nSCA performance is meticulously conducted through comparative experiments involving a range of state-of-the-art algorithms, including MVO, MFO, SSA, WOA, GOA, and the original SCA. To rigorously gauge its capabilities, 23 benchmark test functions are employed, offering a thorough benchmarking of nSCA performance. Additionally, the practical effectiveness of nSCA is demonstrated by successfully addressing five distinct engineering optimization problems.

**Table 11** Comparison findings of pressure vessel design challenge

Optimization technique	Optimal parameters				Optimal cost
	$T_s$	$T_h$	$R$	$L$	
PSO [53]	0.8125	0.4375	42.09127	176.7465	6061.078
DE [54]	0.8125	0.4375	42.09841	176.6377	6059.734
ES [55]	0.8125	0.4375	42.09809	176.6405	6059.746
ACO [56]	0.8125	0.4375	42.10362	176.5727	6059.089
MVO [1]	0.8125	0.4375	42.09074	176.7387	6060.807
GSA [1]	1.125	0.625	55.98866	84.4542	8538.836
AOA [57]	0.830374	0.416206	42.751270	169.345400	6048.784400
RFO [51]	0.81425	0.44521	42.20231	176.62145	6113.3195
SCSO [58]	0.7798	0.9390	40.3864	199.2918	5917.46
nSCA (this study)	0.779260	0.386883	40.358599	199.486844	5895.086



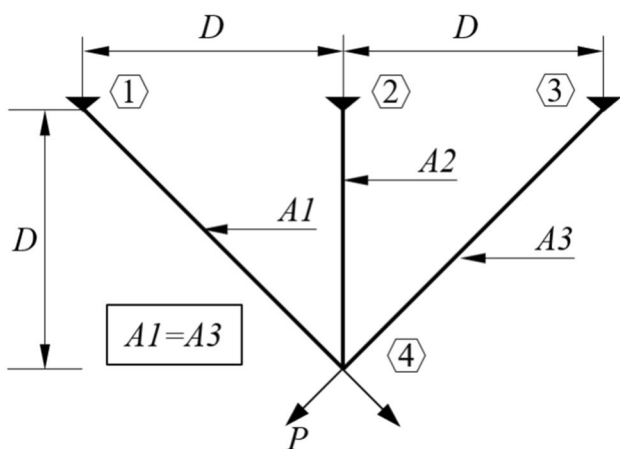


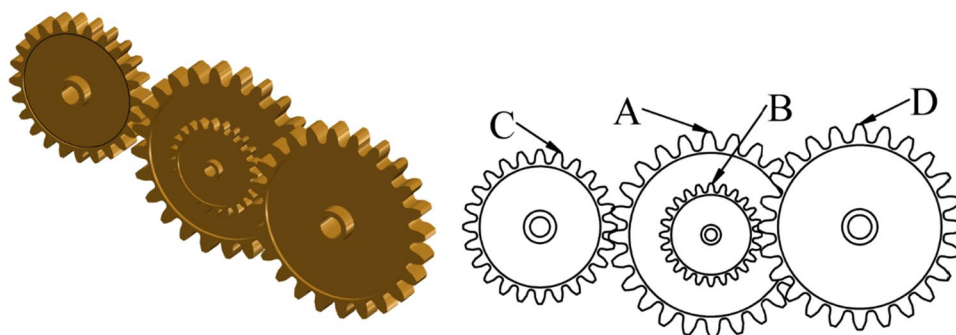
Fig. 11 Three-bar truss design challenge

Table 12 Comparison findings of three-bar truss design challenge

Optimization technique	Optimal parameters		Optimal weight
	$x_1$	$x_2$	
Ray and Saini [59]	0.795	0.395	264.3
Tsai [60]	0.788	0.408	263.68
DEDS [61]	0.788675	0.408248	263.8958
PSO-DE [62]	0.788675	0.408248	263.8958
CS [48]	0.78867	0.40902	263.9716
MBA [63]	0.788565	0.40856	263.8959
ALO [50]	0.788663	0.408283	263.8958
MVO [1]	0.78860276	0.40845307	263.8958499
GOA [6]	0.788898	0.40762	263.8959
nSCA (this study)	0.788675135	0.408248290	263.8958434

The outcomes underscore the superiority of nSCA when compared to alternative evolutionary computation approaches, highlighting its ability to generate exceptionally competitive solutions across both benchmark test functions and real-world engineering optimization challenges. These compelling findings emphasize the value of nSCA as an indispensable tool in the domain of engineering optimization, promising significant contributions to

Fig. 12 Gear train design challenge



problem-solving strategies and decision-making processes. Given these substantial insights, it is evident that nSCA presents an impactful and robust approach well-equipped to address intricate optimization challenges encountered in real-world scenarios.

### Appendix 1

#### Cantilever Beam Design

Consider:

$$\vec{x} = [x_1 x_2 x_3 x_4 x_5]$$

Minimize:

$$f(\vec{x}) = 0.6224(x_1 + x_2 + x_3 + x_4 + x_5)$$

Subject to:

$$g(\vec{x}) = \frac{61}{x_1^3} + \frac{27}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0$$

Variable range:

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

#### Pressure Vessel Design Problem

Consider:

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

Minimize:

$$f(\vec{x}) = 0.6224x_1 x_3 x_4 + 1.7781x_2 x_3^2 + 3.1661x_1^2 x_4 + 19.84x_1^2 x_3$$

Subject to:

**Table 13** Comparison findings of gear train design challenge

Optimization technique	Optimal parameters				Optimal gear ratio
	$n_A$	$n_B$	$n_C$	$n_D$	
Kannan and Kramer [64]	33	15	13	41	2.1469E-08
Deb and Goyal [65]	49	16	19	43	2.7019E-12
ABC [63]	49	16	19	43	2.7009E-12
MBA [63]	43	16	19	49	2.7009E-12
CS [48]	43	16	19	49	2.7009E-12
ISA [66]	N/A	N/A	N/A	N/A	2.7009E-12
MVO [1]	43	16	19	49	2.7009E-12
nSCA (this study)	43	16	19	49	2.7009E-12

$$g_1(\vec{x}) = -x_1 + 0.0193x_3 \leq 0,$$

$$g_2(\vec{x}) = -x_2 + 0.00954x_3 \leq 0,$$

$$g_3(\vec{x}) = -\pi x_3^2 x_4 - \frac{4}{3} \pi x_3^3 + 1296000 \leq 0,$$

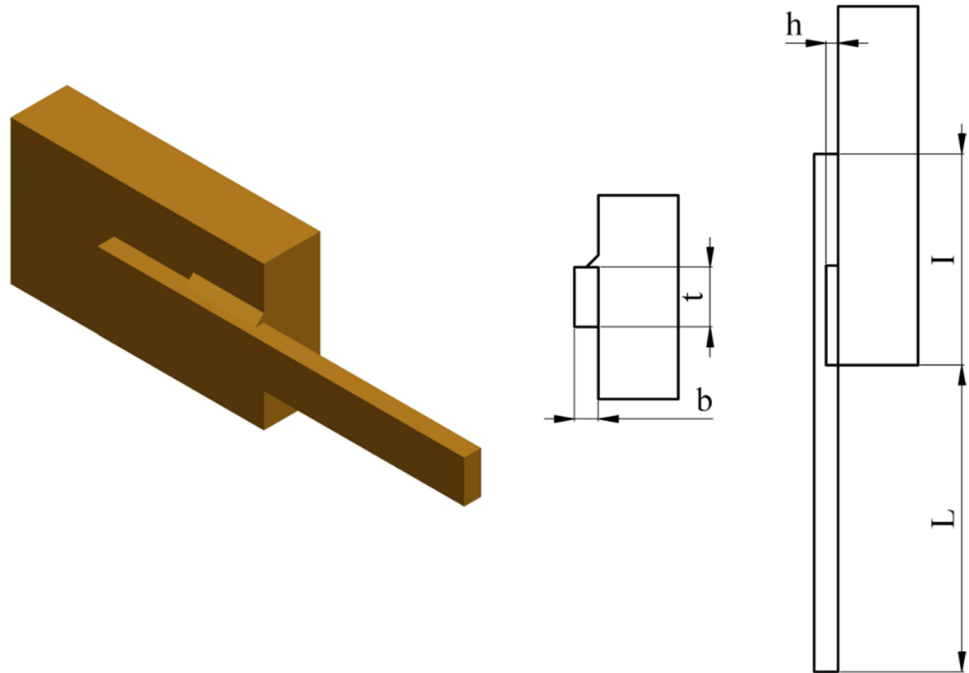
$$g_4(\vec{x}) = x_4 - 240 \leq 0.$$

Variable range:

$$0 \leq x_1, x_2 \leq 99,$$

$$10 \leq x_3, x_4 \leq 200.$$

**Fig. 13** Welded beam design challenge



**Table 14** Comparison findings of welded beam design challenge

Optimization technique	Optimal parameters				Optimal cost
	$h$	$l$	$t$	$b$	
GA [67]	0.1828	4.0483	9.3666	0.2059	1.82455147
HS [53]	0.2442	6.2231	8.2915	0.2443	2.3807
CPSO [1]	0.202369	3.544214	9.04821	0.205723	1.72802
GSA [1]	0.182129	3.856979	10	0.202376	1.87995
MVO [1]	0.205463	3.473193	9.044502	0.205695	1.72645
RFO [51]	0.21846	3.51024	8.87254	0.22491	1.86612
SSA [68]	0.2057	3.4714	9.0366	0.2057	1.72491
nSCA (this study)	0.204248	3.282510	9.035460	0.205798	1.697358

### Three-Bar Truss Design Problem

Consider:

$$\vec{x} = [x_1 x_2] = [A_1 A_2].$$

Minimize:

$$f(\vec{x}) = (2\sqrt{2}x_1 + x_2) \times l.$$

Subject to:

$$g_1(\vec{x}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2x_1^2 + 2x_1x_2}} P - \sigma \leq 0,$$

$$g_2(\vec{x}) = \frac{x_2}{\sqrt{2x_1^2 + 2x_1x_2}} P - \sigma \leq 0,$$

$$g_3(\vec{x}) = \frac{1}{\sqrt{2}x_2 + x_1} P - \sigma \leq 0.$$

Variable range:

$$0 \leq x_1, x_2 \leq 1,$$

where  $l = 100$  cm,  $P = 2$  KN/cm<sup>2</sup>,  $\sigma = 2$  KN/cm<sup>2</sup>.

### Gear Train Design Problem

Consider:

$$\vec{x} = [x_1 x_2 x_3 x_4] = [n_A n_B n_C n_D].$$

Minimize:

$$f(\vec{x}) = \left( \frac{1}{6.931} - \frac{x_2 x_3}{x_1 x_4} \right)^2.$$

Variable range:

$$12 \leq x_1, x_2, x_3, x_4 \leq 60.$$

### Welded Beam Design Problem

Consider:

$$\vec{x} = [x_1 x_2 x_3 x_4] = [hltb].$$

Minimize:

$$f(\vec{x}) = 1.10471x_1^2 x_2 + 0.04811x_3 x_4 (14.0 + x_2).$$

Subject to:

$$g_1(\vec{x}) = \tau(\vec{x}) - \tau_{\max} \leq 0,$$

$$g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{\max} \leq 0,$$

$$g_3(\vec{x}) = \delta(\vec{x}) - \delta_{\max} \leq 0,$$

$$g_4(\vec{x}) = x_1 - x_4 \leq 0,$$

$$g_5(\vec{x}) = P - P_c(\vec{x}) \leq 0,$$

$$g_6(\vec{x}) = 0.125 - x_1 \leq 0,$$

$$g_7(\vec{x}) = 1.10471x_1^2 + 0.04811x_3 x_4 (14 + x_2) - 5.0 \leq 0.$$

Variable range:

$$0.1 \leq x_1, x_4 \leq 2,$$

$$0.1 \leq x_2, x_3 \leq 10,$$

where

$$\tau(\vec{x}) = \sqrt{(\tau')^2 + \frac{2\tau'\tau''x_2}{2R} + (\tau'')^2}, \tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J},$$

$$M = P\left(L + \frac{x_2}{2}\right), R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2},$$

$$J = 2 \left\{ \sqrt{2}x_1x_2 \left[ \frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2 \right] \right\},$$

$$\sigma(\vec{x}) = \frac{6PL}{x_4x_3^2}, \delta(\vec{x}) = \frac{6PL^3}{Ex_3^2x_4},$$

$$P_c(\vec{x}) = \frac{4.013E \sqrt{\frac{x_3^2 x_4^6}{36}}}{L^2} \left( 1 - \frac{x_3}{2L} \sqrt{\frac{E}{4G}} \right),$$

$$P = 6000\text{lb}, L = 14\text{in.}, \delta_{\max} = 0.25\text{in.},$$

$$E = 30 \times 10^6\text{psi}, G = 12 \times 10^6\text{psi},$$

$$\tau_{\max} = 13600\text{psi}, \sigma_{\max} = 30000\text{psi}.$$

**Acknowledgements** We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

**Author Contributions** All authors, including VHSP, NTND, and VNN jointly contributed to the writing of the main manuscript, preparation of all figures and tables, and reviewed and approved the final version prior to submission.

**Funding** This research did not receive dedicated funding from public, commercial, or non-profit grant agencies.

**Data Availability** The corresponding author is available to provide the data, model, or code underlying the findings of this study upon request, in accordance with reasonable conditions.

## Declarations

**Conflict of Interest** There is no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Mirjalili, S., Mirjalili, S.M., Hatamlou, A.: Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **27**(2), 495–513 (2016)
- Mirjalili, S.: SCA: a sine cosine algorithm for solving optimization problems. *Knowl.-Based Syst.* **96**, 120–133 (2016)
- Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* **11**(4), 341 (1997)
- Mirjalili, S.: Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. *Knowl.-Based Syst. Based Syst.* **89**, 228–249 (2015)
- Mirjalili, S., Lewis, A.: The whale optimization algorithm. *Adv. Eng. Softw.* **95**, 51–67 (2016)
- Saremi, S., Mirjalili, S., Lewis, A.: Grasshopper optimisation algorithm: theory and application. *Adv. Eng. Softw.* **105**, 30–47 (2017)
- Mirjalili, S., et al.: Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **114**, 163–191 (2017)
- Črepinšek, M., Liu, S.-H., Mernik, M.: Exploration and exploitation in evolutionary algorithms: a survey. *ACM Comput. Surv. (CSUR)* **45**(3), 1–33 (2013)
- Lin, L., Gen, M.: Auto-tuning strategy for evolutionary algorithms: balancing between exploration and exploitation. *Soft. Comput.* **13**(2), 157–168 (2009)
- Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1**(1), 67–82 (1997)
- Pham, V.H.S., Trang, N.T.N., Dat, C.Q.: Optimization of production schedules of multi-plants for dispatching ready-mix concrete trucks by integrating grey wolf optimizer and dragonfly algorithm. *Eng. Construct. Architect. Manage.* (2023). <https://doi.org/10.1108/ECAM-12-2022-1176>
- Son, P.V.H., Nguyen Dang, N.T.: Optimizing time and cost simultaneously in projects with multi-verse optimizer. *Asian J. Civ. Eng.* (2023). <https://doi.org/10.1007/s42107-023-00652-y>
- Qiao, W., et al.: A hybrid algorithm for carbon dioxide emissions forecasting based on improved lion swarm optimizer. *J. Clean. Prod.* **244**, 118612 (2020)
- Altay, O., Cetindemir, O., Aydogdu, I.: Size optimization of planar truss systems using the modified salp swarm algorithm. *Eng. Optimiz.* (2023). <https://doi.org/10.1080/0305215X.2022.2160449>
- Pham, V.H.S., Souliha, F.V.: A hybrid ant lion optimizer (ALO) algorithm for construction site layout optimization. *J. Soft Comput. Civ. Eng.* **7**(4), 50–71 (2023)
- Goksal, F.P., Karaoglan, I., Altiparmak, F.: A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery. *Comput. Ind. Eng.* **65**(1), 39–53 (2013)
- Son, P.V.H., Duy, N.H.C., Dat, P.T.: Optimization of construction material cost through logistics planning model of dragonfly algorithm—particle swarm optimization. *KSCE J. Civ. Eng.* **25**(7), 2350–2359 (2021)
- Shang, C., Zhou, T.-T., Liu, S.: Optimization of complex engineering problems using modified sine cosine algorithm. *Sci. Rep.* **12**(1), 20528 (2022)
- Raut, U. and Mishra, S.: Power distribution network reconfiguration using an improved sine–cosine algorithm-based meta-heuristic search. In: *Soft Computing for Problem Solving: SocProS 2017*, Volume 1. 2019. Springer
- Reddy, K.S., et al.: A new binary variant of sine–cosine algorithm: development and application to solve profit-based unit commitment problem. *Arab. J. Sci. Eng.* **43**, 4041–4056 (2018)
- Sahlol, A.T., et al.: Training feedforward neural networks using Sine-Cosine algorithm to improve the prediction of liver enzymes on fish farmed on nano-selenite. In: *2016 12th International Computer Engineering Conference (ICENCO)*. 2016. IEEE
- Zhao, Y., Zou, F. and Chen, D.: A discrete sine cosine algorithm for community detection. In: *Intelligent Computing Theories and Application: 15th International Conference, ICIC 2019, Nanchang, China, August 3–6, 2019, Proceedings, Part I* 15. 2019. Springer
- Aydin, O., et al.: Comparative parameter estimation of single diode PV-cell model by using sine-cosine algorithm and whale optimization algorithm. In: *2019 6th International Conference on Electrical and Electronics Engineering (ICEEE)*. 2019. IEEE
- Cheng, J., Duan, Z.: Cloud model based sine cosine algorithm for solving optimization problems. *Evol. Intel.* **12**, 503–514 (2019)
- Bureerat, S. and Pholdee, N.: Adaptive sine cosine algorithm integrated with differential evolution for structural damage detection. In: *International Conference on Computational Science and Its Applications*. 2017. Springer
- Turgut, O.E.: Thermal and economical optimization of a shell and tube evaporator using hybrid backtracking search—sine–cosine algorithm. *Arab. J. Sci. Eng.* **42**(5), 2105–2123 (2017)
- Bairathi, D. and Gopalani, D.: Opposition-based sine cosine algorithm (OSCA) for training feed-forward neural networks. In: *2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. 2017. IEEE
- Qu, C., et al.: A modified sine-cosine algorithm based on neighborhood search and greedy levy mutation. *Comput. Intell. Neurosci.* *Intell. Neurosci.* (2018). <https://doi.org/10.1155/2018/4231647>
- Son, P.V.H., Nguyen Dang, N.T.: Solving large-scale discrete time–cost trade-off problem using hybrid multi-verse optimizer model. *Sci. Reports* **13**(1), 1987 (2023)

30. Pham, V.H.S., Nguyen, V.N.: Cement transport vehicle routing with a hybrid sine cosine optimization algorithm. *Adv. Civ. Eng.* **2023**, 2728039 (2023)
31. Abualigah, L., Diabat, A.: Advances in sine cosine algorithm: a comprehensive survey. *Artif. Intell. Rev.* **54**(4), 2567–2608 (2021)
32. Pandey, A.C., Kulhari, A., Shukla, D.S.: Enhancing sentiment analysis using roulette wheel selection based cuckoo search clustering method. *J. Ambient. Intell. Humaniz. Comput.* **13**(1), 1–29 (2022)
33. Zhu, Y.-P., et al.: A ranking weight based roulette wheel selection method for comprehensive learning particle swarm optimization. In: 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC). 2022. IEEE
34. Yu, F., et al.: Improved roulette wheel selection-based genetic algorithm for TSP. In: 2016 International Conference on Network and Information Systems for Computers (ICNISC). 2016. IEEE
35. Ho-Huu, V., et al.: An improved differential evolution based on roulette wheel selection for shape and size optimization of truss structures with frequency constraints. *Neural Comput. Appl.* **29**, 167–185 (2018)
36. Lloyd, H. and Amos, M.: Analysis of independent roulette selection in parallel ant colony optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference. 2017
37. Tizhoosh, H.R.: Opposition-based learning: a new scheme for machine intelligence. In: International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06). 2005. IEEE
38. Verma, O.P., Aggarwal, D., Patodi, T.: Opposition and dimensional based modified firefly algorithm. *Expert Syst. Appl.* **44**, 168–176 (2016)
39. Upadhyay, P., et al.: A novel design method for optimal IIR system identification using opposition based harmony search algorithm. *J. Franklin Inst.* **351**(5), 2454–2488 (2014)
40. Luong, D.-L., Tran, D.-H., Nguyen, P.T.: Optimizing multi-mode time-cost-quality trade-off of construction project using opposition multiple objective difference evolution. *Int. J. Constr. Manag.* **21**(3), 271–283 (2021)
41. Wang, H., et al.: Enhancing particle swarm optimization using generalized opposition-based learning. *Inf. Sci.* **181**(20), 4699–4714 (2011)
42. Ewees, A.A., Abd-Elaziz, M., Houssein, E.H.: Improved grasshopper optimization algorithm using opposition-based learning. *Expert Syst. Appl.* **112**, 156–172 (2018)
43. Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. *IEEE Trans. Evol. Comput.* **3**(2), 82–102 (1999)
44. Yang, X.-S.: Test problems in optimization. arXiv preprint [arXiv:1008.0549](https://arxiv.org/abs/1008.0549), 2010.
45. Digalakis, J.G., Margaritis, K.G.: On benchmarking functions for genetic algorithms. *Int. J. Comput. Math.* **77**(4), 481–506 (2001)
46. Van den Bergh, F., Engelbrecht, A.P.: A study of particle swarm optimization particle trajectories. *Inf. Sci.* **176**(8), 937–971 (2006)
47. Wu, G., Mallipeddi, R. and Suganthan, P.N.: Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization. National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report, 2017.
48. Gandomi, A.H., Yang, X.-S., Alavi, A.H.: Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Eng. Comput. Comput.* **29**, 17–35 (2013)
49. Cheng, M.-Y., Prayogo, D.: Symbiotic organisms search: a new metaheuristic optimization algorithm. *Comput. Struct.* **139**, 98–112 (2014)
50. Mirjalili, S.: The ant lion optimizer. *Adv. Eng. Softw.* **83**, 80–98 (2015)
51. Polap, D., Woźniak, M.: Red fox optimization algorithm. *Expert Syst. Appl.* **166**, 114107 (2021)
52. Jia, H., et al.: Crayfish optimization algorithm. *Artif. Intell. Rev. Intell. Rev.* (2023). <https://doi.org/10.1007/s10462-023-10567-4>
53. Lee, K.S., Geem, Z.W.: A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. *Comput. Methods Appl. Mech. Eng.* **194**(36–38), 3902–3933 (2005)
54. Li, L.-J., et al.: A heuristic particle swarm optimizer for optimization of pin connected structures. *Comput. Struct.* **85**(7–8), 340–349 (2007)
55. Mezura-Montes, E., Coello, C.A.C.: An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. *Int. J. Gen. Syst.* **37**(4), 443–473 (2008)
56. Kaveh, A., Talatahari, S.: An improved ant colony optimization for constrained engineering design problems. *Eng. Comput. Comput.* (2010). <https://doi.org/10.1108/02644401011008577>
57. Abualigah, L., et al.: The arithmetic optimization algorithm. *Comput. Methods Appl. Mech. Eng.* **376**, 113609 (2021)
58. Seyyedabbasi, A., Kiani, F.: Sand Cat swarm optimization: a nature-inspired algorithm to solve global optimization problems. *Eng. Comput. Comput.* **39**(4), 2627–2651 (2023)
59. Ray, T., Saini, P.: Engineering design optimization using a swarm with an intelligent information sharing among individuals. *Eng. Optim.* **33**(6), 735–748 (2001)
60. Tsai, J.-F.: Global optimization of nonlinear fractional programming problems in engineering design. *Eng. Optim.* **37**(4), 399–409 (2005)
61. Zhang, M., Luo, W., Wang, X.: Differential evolution with dynamic stochastic selection for constrained optimization. *Inf. Sci.* **178**(15), 3043–3074 (2008)
62. Liu, H., Cai, Z., Wang, Y.: Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Appl. Soft Comput.* **10**(2), 629–640 (2010)
63. Sadollah, A., et al.: Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. *Appl. Soft Comput.* **13**(5), 2592–2612 (2013)
64. Kannan, B. and Kramer, S.N.: An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. 1994.
65. Deb, K., Goyal, M.: A combined genetic adaptive search (GeneAS) for engineering design. *Comput. Sci. Inform.* **26**, 30–45 (1996)
66. Gandomi, A.H.: Interior search algorithm (ISA): a novel approach for global optimization. *ISA Trans.* **53**(4), 1168–1183 (2014)
67. Coello Coello, C.A.: Constraint-handling using an evolutionary multiobjective optimization technique. *Civ. Eng. Syst.* **17**(4), 319–346 (2000)
68. Hashim, F.A., Hussien, A.G.: Snake Optimizer: A novel metaheuristic optimization algorithm. *Knowl.-Based Syst.-Based Syst.* **242**, 108320 (2022)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.