Research Article

A Hybrid Lightning Search Algorithm-Simplex Method for Global Optimization

Yuting Lu,¹ Yongquan Zhou,^{2,3} and Xiuli Wu¹

¹School of Computer, Electronics and Information, Guangxi University, Nanning 530004, China

²College of Information Science and Engineering, Guangxi University for Nationalities, Nanning 530006, China

³Key Laboratory of Guangxi High Schools Complex System and Intelligent Computing, Nanning 530006, China

Correspondence should be addressed to Yongquan Zhou; yongquanzhou@126.com

Received 11 March 2017; Accepted 1 June 2017; Published 13 July 2017

Academic Editor: Pasquale Candito

Copyright © 2017 Yuting Lu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, a novel hybrid lightning search algorithm-simplex method (LSA-SM) is proposed to solve the shortcomings of lightning search algorithm (LSA) premature convergence and low computational accuracy and it is applied to function optimization and constrained engineering design optimization problems. The improvement adds two major optimization strategies. Simplex method (SM) iteratively optimizes the current worst step leaders to avoid the population searching at the edge, thus improving the convergence accuracy and rate of the algorithm. Elite opposition-based learning (EOBL) increases the diversity of population to avoid the algorithm falling into local optimum. LSA-SM is tested by 18 benchmark functions and five constrained engineering design problems. The results show that LSA-SM has higher computational accuracy, faster convergence rate, and stronger stability than other algorithms and can effectively solve the problem of constrained nonlinear optimization in reality.

1. Introduction

Optimization is concerned with finding the best solutions for a given problem. It is everywhere and important in many applications such as production line scheduling, power system optimization, 3D path planning, and traveling salesman problem (TSP). The goal of optimization is to minimize costs to maximize profitability and efficiency. In the broadest sense, techniques for dealing with different types of optimization problems are classified into exact and stochastic algorithms. When the problems are large and complex, especially if they are either NP-complete or NP-hard, the use of stochastic algorithms becomes mandatory. These stochastic algorithms do not guarantee an optimal solution, but they are able to find quasi-optimal solutions within a reasonable amount of time [1].

Metaheuristic optimization algorithms are stochastic algorithms that have become the most popular solution for solving optimization problems in the last few decades and they have the characteristics of simplicity, flexibility, strong robustness, and so on. Some of the most famous of

these algorithms are genetic algorithm (GA) [2], ant colony optimization (ACO) [3], particle swarm optimization (PSO) [4], harmony search (HS) [5], artificial bee colony (ABC) [6], cuckoo search (CS) [7], bat algorithm (BA) [8], bacterial foraging optimization (BFO) [9], black hole (BH) [10], and so forth. The GA simulates the Darwinian biological evolutionary principle. In this algorithm, individuals are expressed as genotypes. Each individual is selected, crossed, and mutated in the evolutionary process, in accordance with the “survival of the fittest” principle of evolution, and finally the optimal solution to the problem is gotten. The ACO algorithm is based on the phenomenon that ants communicate with each other during the foraging process by releasing pheromones to find the best path between the nest and the food source. The PSO algorithm is inspired by bird predation. This algorithm regards birds as no mass particles, and, in the search space, the change in particle position depends not only on their own experience, but also on the excellent individual of society. The HS algorithm simulates the process by which musicians adjust the different instrumental tones repeatedly to achieve the most beautiful and harmony. The ABC algorithm is

a bionic intelligent method for simulating bee populations to find excellent nectar. In short, the inspiration and wisdom from nature spawned the development of heuristic algorithms to solve complex and large-scale optimization problems. In addition, inspired by the natural phenomena of lightning, a new metaheuristic optimization algorithm called lightning search algorithm (LSA) [11] is proposed.

Lightning search algorithm was first proposed by Shareef et al. in 2015, which is inspired by the phenomenon of lightning discharge to the ground during a thunderstorm to solve constraint optimization problems. The proposed optimization algorithm utilizes the mechanism of step leader propagation and considers the involvement of fast particles known as projectiles in the formation of the binary tree structure of a step leader. It has been proved that LSA has high convergence rate and successfully solved the small-scale TSP problem. Although a long time did not pass since the introduction of LSA, it has been widely studied by scholars. In 2015, Shareef et al. used LSA to solve fuzzy logic PV inverter controller optimization problems [12]. In the same year, Ali et al. proposed a novel quantum-behaved lightning search algorithm (QLSA) to improve the fuzzy logic speed controller for an induction motor drive [13]. In 2016, Islam et al. proposed a variant of LSA to solve binary optimization problems called binary lightning search algorithm (BLSA) [14]. Sirjani and Shareef applied LSA for parameter extraction of solar cell models in different weather conditions [15]. Mutlag et al. applied LSA to design an optimal fuzzy logic controller for a PV inverter [16]. LSA was used to improve the artificial neural network (LSA-ANN) for the home energy management scheduling controller for the residential demand response strategy in 2016 [17]. LSA was also used to optimize the learning process of feedforward neural networks [18]. Because the LSA algorithm simulates the fast propagation characteristics of lightning, it has high convergence rate and strong robustness. However, there are some shortcomings in LSA itself, such as having insufficient stability, being easy to fall into local optimum, and having low convergence accuracy. Therefore, the goal of this paper is to improve the LSA and expand its application.

In this study, a novel optimization method called hybrid lightning search algorithm-simplex method (LSA-SM) has been applied to function optimization and constrained engineering design optimization. Simplex method (SM) has the characteristics of fast search speed, small computation amount, and strong local optimization ability. By introducing the reflection, expansion, and compression operations of the simplex method in the LSA algorithm to improve the step leaders in the worst position, the accuracy of the algorithm is improved. LSA-SM adds two strategies, SM and elite opposition-based learning (EOBL). SM iteratively optimizes the worst step leaders, making the population faster and closer to the global optimal solution. EOBL increases the diversity of population and expands the search space to avoid the algorithm falling into local optimum. The performance of proposed LSA-SM is tested by the well-known 18 benchmark functions and five constrained engineering design problems. The test results show that LSA-SM has faster convergence rate, higher optimization accuracy, and strong stability. The remainder of the paper is organized as follows.

FIGURE 1: Lightning discharge and stepped leader propagation.

Section 2 briefly introduces the original lightning search algorithm. Section 3 presents the novel hybrid lightning search algorithm-simplex method. Simulation experiment setup, results, and analysis are provided in Section 4. The proposed LSA-SM is applied to solve five constrained engineering design optimization problems in Section 5. Finally, Section 6 gives conclusions and future studies.

2. Lightning Search Algorithm (LSA)

Lightning search algorithm (LSA) is based on the natural phenomenon of lightning, and it is inspired by the probabilistic nature and sinuous characteristics of lightning discharges during a thunderstorm (Figure 1). This optimization algorithm is generalized from the mechanism of step leader propagation using the theory of fast particles known as projectiles. The projectile represents the initial population size, similar to the term “particle” used in the PSO. In the LSA, the solution refers to the tip of the current step leader’s energy [11, 12].

2.1. Projectile Model. LSA consists of three types of projectiles: transition, space, and lead projectiles. The transition projectiles create the first step leader population for solutions, the space projectiles engage in exploration and attempt to become the leader, and the lead projectiles attempt to find and exploit the optimal solution [15].

2.1.1. Transition Projectile. An early stage of formation of a stepped lead, the transition projectile $P^T = [p_1^T, p_2^T, \dots, p_N^T]$ is ejected from the thunder cell in a random direction. Therefore, it can be modeled as a random number drawn from the standard uniform probability distribution as follows:

$$f(x^T) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq x^T \leq b \\ 0 & \text{elsewhere,} \end{cases} \quad (1)$$

where x^T is a random number that may provide a solution and a and b are the lower and upper bounds, respectively, of the solution space. For a population of N stepped leaders

SL = $[sl_1, sl_2, \dots, sl_N]$, N random projectiles need to meet the solution dimension.

2.1.2. Space Projectile. The position of the space projectile $P^S = [p_1^S, p_2^S, \dots, p_N^S]$ at step + 1 can be modeled as a random number generated from the exponential distribution with shaping parameter μ as follows:

$$f(x^S) = \begin{cases} \frac{1}{\mu} e^{-x^S/\mu} & \text{for } x^S > 0 \\ 0 & \text{for } x^S \leq 0. \end{cases} \quad (2)$$

Thus, the position and direction of p_i^S at step + 1 can be written as follows:

$$p_{i_new}^S = p_i^S \pm \exp \text{rand}(\mu_i), \quad (3)$$

where $\exp \text{rand}$ is an exponential random number and μ_i is taken as the distance between the lead projectile p^L and the space projectile p_i^S under consideration. If $p_{i_new}^S$ provides a good solution at step + 1 and the projectile energy $E_{p_i}^S$ is greater than the step leader E_{sl_i} , then p_i^S is updated to $p_{i_new}^S$. Otherwise, they remain unchanged until the next step.

2.1.3. Lead Projectile. The lead projectile P^L moves closer to the ground, which can be modeled as a random number taken from the standard normal distribution as follows:

$$f(x^L) = \frac{1}{\sigma \sqrt{2\pi}} e^{-(x^L - \mu)^2 / 2\sigma^2}. \quad (4)$$

The randomly generated lead projectile can search in all directions from the current position defined by the shape parameter (μ_L). This projectile also has an exploitation ability defined by the scale parameter (σ_L). The scale parameter σ_L exponentially decreases as it progresses towards the Earth or as it finds the best solution. Thus, the position of p^L at step + 1 can be written as follows:

$$p_{new}^L = p^L + \text{normrand}(\mu_L, \sigma_L), \quad (5)$$

where normrand is a random number generated by the normal distribution function. Similarly, if p_{new}^L provides a good solution at step + 1 and the projectile energy $E_{p_i}^L$ is greater than the step leader E_{sl_i} , then p^L is updated to p_{new}^L . Otherwise, they remain unchanged until the next step.

2.2. Forking Procedure. Forking is an important property of a stepped leader, in which two simultaneous and symmetrical branches emerge. In the proposed algorithm, forking is realized in two ways. First, symmetrical channels are created because the nuclei collision of the projectile is realized by using the opposite number as follows:

$$\bar{p}_i = a + b - p_i, \quad (6)$$

where $\bar{p}_i$ and p_i are the opposite and original projectiles, respectively, in a one-dimensional system and a and b are the

boundary limits. In order to maintain the population size, the forking leader selects $\bar{p}_i$ or p_i with a better fitness value.

In the second type of forking, a channel is assumed to appear at a successful step leader tip because of the energy redistribution of the most unsuccessful leader after several propagation trials. The unsuccessful leader can be redistributed by defining the maximum allowable number of trials as channel time.

The flowchart of the LSA procedure is shown in Figure 2.

3. Hybrid Lightning Search Algorithm-Simplex Method (LSA-SM)

Standard LSA has a fast convergence rate, but there are still some shortcomings, such as premature convergence, easy fall into local optimum, poor solution accuracy, and low ability to solve multimodal optimization problems. In order to improve the search performance of LSA, a hybrid lightning search algorithm-simplex method (LSA-SM) is adopted. In view of the shortcomings of LSA, two optimization strategies, namely, simplex method (SM) and elite opposition-based learning (EOBL), are added to the standard LSA.

3.1. Simplex Method (SM). Simplex method has exceptional advantages in local search, often optimized to obtain higher precision. The method is proposed by Nelder and Mead, a simple and derivative-free line search method for finding a local minimum of a function. It is based on the idea of comparing the values of the objective function at the $N + 1$ vertices of a polytope (simplex) in N -dimensional space and moving the polyhedron towards the minimum point as the optimization progress [44, 45]. In this paper, we choose K step leaders with the worst objective function values and use SM to optimize their positions, as shown in Figure 3. This allows the algorithm to be closer to the optimal solution, improve its exploit ability, and speed it up to find the optimal solution. The procedures of the SM in this study are shown as follows.

Step 1. According to the objective function values of all searched step leaders, find the optimal point (with minimum function value) xg , the suboptimal point xb , and the K worst points; take one of them as xw .

Step 2. Calculate the center $\bar{x}$ of the optimal point xg and suboptimal point xb .

$$\bar{x} = \frac{xg + xb}{2}. \quad (7)$$

Step 3. Reflect xw through the center $\bar{x}$ to a new point xr . And calculate the objective function value at this point.

$$xr = \bar{x} + \alpha(\bar{x} - xw), \quad (8)$$

where $\alpha = 1$ is reflection coefficient.

FIGURE 2: Flowchart of the LSA.

Step 4. If $f(xr) < f(xg)$ then perform expansion to generate a new point xe ; otherwise go to Step 5. If $f(xe) < f(xg)$ then replace xw by xe ; otherwise replace xw by xr .

$$xe = \bar{x} + \gamma (xr - \bar{x}), \quad (9)$$

where $\gamma = 1.5$ is expansion coefficient.

Step 5. If $f(xr) > f(xw)$, reflection failed, and then perform contraction on point xw to produce a new point xc ; otherwise go to Step 6. If $f(xc) < f(xw)$ then replace xw by xc .

$$xc = \bar{x} + \beta (xw - \bar{x}), \quad (10)$$

where $\beta = 0.5$ is contraction coefficient.

FIGURE 3: Simplex method to search for different points.

Step 6. If $f(xg) < f(xr) < f(xw)$ then shrink xw to generate a new point xs . The shrinkage coefficient is the same as the contraction coefficient.

$$xs = \bar{x} + \beta(\bar{x} - xw). \quad (11)$$

The SM allows the current worst step leaders to find a better position at each iteration, which may be better than the optimal point. This avoids the population searching at the edge, guiding it towards the global optimum and the convergence accuracy and rate of the algorithm are improved.

3.2. Elite Opposition-Based Learning (EOBL). Although SM improves the convergence accuracy of the LSA, it is easy to fall into the local optimum, so the EOBL strategy is introduced to increase the diversity of population and expand the search space.

Opposition-based learning (OBL), formally introduced by Tizhoosh [46], is a new model of machine intelligence that takes into account current estimate and its opposite estimate at the same time to achieve a better solution. It has been proved that an opposite candidate solution has a higher chance to be closer to the global optimal solution than a random candidate solution [47]. Elite opposition-based learning (EOBL) [48] is based on the elite step leader using OBL principle to generate elite opposition-based population to participate in competitive evolution, so as to improve population diversity of LSA. The step leader with the best fitness value is defined as elite step leader $X_e = (x_{e,1}, x_{e,2}, \dots, x_{e,D})$; elite opposition-based solution of step leader $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$ can be defined as $X'_i = (x'_{i,1}, x'_{i,2}, \dots, x'_{i,D})$ using the following equation:

$$x'_{i,j} = k \cdot (Lb_j + Ub_j) - x_{e,j}, \quad (12)$$

$$i = 1, 2, \dots, n; \quad j = 1, 2, \dots, D,$$

where n is the population size, D is the dimension of X , $k \in U(0, 1)$, and (Lb_j, Ub_j) is the search bound. If the elite opposition-based step leader $x'_{i,j}$ exceeds the search boundary, the processing formula is used as follows:

$$x'_{i,j} = \text{rand}(Lb_j, Ub_j), \quad \text{if } x'_{i,j} < Lb_j \text{ or } x'_{i,j} > Ub_j. \quad (13)$$

According to the above principles, the EOBL procedures are as follows.

Step 1. Define an elite step leader X_e with the current best fitness value, and then use (12) to generate an elite opposition-based population with size n .

Step 2. If the elite opposition-based step leader exceeds the search boundary, then (13) is used.

Step 3. $2n$ step leaders are involved in evolutionary competition, and n step leaders with better fitness values are selected to enter the next generation.

Two strategies of SM and EOBL are added to the LSA to balance the exploitation and exploration of the algorithm; the flowchart of the LSA-SM is shown in Figure 4. From the above analysis, it can be concluded that the time complexity of LSA-SM is $O(\text{Max_iter} * (n * D + n * \text{Cof}))$ and the space complexity is $O(2n * D)$, where Max_iter is maximum number of iterations, n is the population size, D is the dimension of the problem, and Cof is the cost of the objective function.

4. Simulation Experiment and Result Analysis

To verify the performance of LSA-SM, the following will be a variety of numerical contrast experiments, using the benchmark functions to test LSA-SM and compare the results to some well-known algorithms. In this section, 18 benchmark functions are applied to test the LSA-SM, which are classic test functions used by many researchers [11, 28, 49–51]. These benchmark functions are listed in Tables 1–3 where Dim indicates dimension of the function, Range is the boundary of the function search space, and $f_{\min}$ is the global optimum. Generally speaking, the 18 benchmark functions used are minimization functions and can be divided into three groups: f_{01} to f_{05} are high-dimensional unimodal functions as group 1, f_{06} to f_{10} are high-dimensional multimodal functions as group 2, and f_{11} to f_{18} are fixed-dimension multimodal functions as group 3. Unimodal functions have only one optimal value, and they are suitable for testing the exploitation of algorithms. In contrast, multimodal functions have many local optima, and the number of local optima increases exponentially with dimension, which makes them suitable for exploration.

4.1. Experimental Setup. All numerical comparison experiments were run under MATLAB R2012a using an Intel(R) Xeon(R) CPU E5-1620 v3 @ 3.50 GHz processor and 8.00 GB memory.

4.2. Performance Comparison of Algorithms. In this section, seven algorithms are chosen to compare with LSA-SM, which are LSA [11], bat algorithm (BA) [8], flower pollination algorithm (FPA) [52], gravitational search algorithm (GSA) [51], grey wolf optimizer (GWO) [28], whale optimization algorithm (WOA) [53], and grasshopper optimization algorithm (GOA) [54]. The population size and maximum iteration of

TABLE 1: Unimodal benchmark functions.

Name	Benchmark function	Dim	Range	$f_{\min}$
Sphere	$f_{01}(x) = \sum_{i=1}^D x_i^2$	30	$[-100, 100]$	0
Schwefel 2.22	$f_{02}(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	30	$[-10, 10]$	0
Schwefel 1.2	$f_{03}(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$	30	$[-100, 100]$	0
Schwefel 2.21	$f_{04}(x) = \max_i \{ x_i , 1 \leq i \leq D\}$	30	$[-100, 100]$	0
Step	$f_{05}(x) = \sum_{i=1}^D ([x_i + 0.5])^2$	30	$[-100, 100]$	0

TABLE 2: Multimodal benchmark functions.

Name	Benchmark function	Dim	Range	$f_{\min}$
Schwefel	$f_{06}(x) = \sum_{i=1}^D -x_i \sin(\sqrt{ x_i })$	30	$[-500, 500]$	-12569.5
Rastrigin	$f_{07}(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]$	0
Ackley	$f_{08}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$	30	$[-32, 32]$	0
Griewank	$f_{09}(x) = \frac{1}{4000} \sum_{i=1}^D (x_i^2) - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]$	0
Penalized	$f_{10}(x) = \frac{\pi}{D} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_D - 1)^2 \right\}$ $+ \sum_{i=1}^D u(x_i, 10, 100, 4)$	30	$[-50, 50]$	0
	$y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$			

TABLE 3: Fixed-dimension multimodal benchmark functions.

Name	Benchmark function	Dim	Range	$f_{\min}$
Kowalik	$f_{11}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_i (b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	$[-5, 5]$	0.0003075
Branin	$f_{12}(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	2	$[-5, 10] \times [0, 15]$	0.398
Goldstein-Price	$f_{13}(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	$[-2, 2]$	3
Hartman 3	$f_{14}(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2\right)$	3	$[0, 1]$	-3.86
Hartman 6	$f_{15}(x) = -\sum_{i=1}^6 c_i \exp\left(-\sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2\right)$	6	$[0, 1]$	-3.32
Shekel 5	$f_{16}(x) = -\sum_{i=1}^5 \left[(x - a_i)(x - a_i)^T + c_i \right]^{-1}$	4	$[0, 10]$	-10.1532
Shekel 7	$f_{17}(x) = -\sum_{i=1}^7 \left[(x - a_i)(x - a_i)^T + c_i \right]^{-1}$	4	$[0, 10]$	-10.4028
Shekel 10	$f_{18}(x) = -\sum_{i=1}^{10} \left[(x - a_i)(x - a_i)^T + c_i \right]^{-1}$	4	$[0, 10]$	-10.5363

the eight algorithms are set to 50 and 1000, respectively. The number of the current worst step leaders of LSA-SM is $K = 10$. The control parameters associated with other algorithm are given below.

LSA parameter setting: channel time $T = 10$; forking rate $r = 0.01$. The source code of LSA is publicly available at <http://cn.mathworks.com/matlabcentral/fileexchange/54181-lightning-search-algorithm-lsa->.

BA parameter setting: loudness $A = 0.5$, pulse rate $r = 0.5$, frequency minimum $Q_{\min} = 0$, and frequency maximum $Q_{\max} = 2$. The source code of BA is publicly available at <http://cn.mathworks.com/matlabcentral/fileexchange/37582-bat-algorithm-demo->.

FPA parameter setting: switch probability $\rho = 0.8$. The source code of FPA is publicly available at <http://cn.mathworks.com/matlabcentral/fileexchange/45112-flower-pollination-algorithm->.

GSA parameter setting: norm = 2, $\alpha = 20$, $G_0 = 100$, and final_per = 2. The source code of GSA is publicly available at <http://cn.mathworks.com/matlabcentral/fileexchange/27756-gravitational-search-algorithm-gsa->.

GWO parameter setting: $a = 2 - l \times (2/\text{Max_iter})$ linearly decreases from 2 to 0. The source code of GWO is publicly available at <http://cn.mathworks.com/matlabcentral/fileexchange/44974-grey-wolf-optimizer-gwo->.

WOA parameter setting: $a = 2 - t \times (2/\text{Max_iter})$ linearly decreases from 2 to 0. The source code of WOA is publicly available at <http://cn.mathworks.com/matlabcentral/fileexchange/55667-the-whale-optimization-algorithm->.

GOA parameter setting: cMax = 1; cMin = 0.00004. The source code of GOA is publicly available at <http://cn.mathworks.com/matlabcentral/fileexchange/61421-grasshopper-optimisation-algorithm-goa->.

The results of the comparative experiments for the 20 independent runs of the algorithms are summarized in Tables 4–6, which are Best, Worst, Mean, and Std., four evaluation indices that represent the optimal fitness value, worst fitness value, mean fitness value, and standard deviation, respectively. For each benchmark function, the eight algorithms are sorted by standard deviation, and the standard deviation is as small as possible. The last column in Tables 4–6 is the sorting result of LSA-SM. The minimum, best median, and the minimum standard deviation values among the eight algorithms of each benchmark function are shown in bold.

In order to verify the validity of the above values, nonparametric Wilcoxon rank tests were performed on the results of LSA-SM and other comparison algorithms running 20 times for 18 benchmark functions. The test results are shown in Table 7, where the value of h being 1 indicates that the performance and comparative method are statistically different with 95% confidence and 0 implies there is no statistical difference. That is, when $p_value \geq 0.05$, then $h = 0$. As shown in Table 7, “ $h = 1$ ” accounts for the vast majority of the results, which implies that the proposed LSA-SM has different statistical significance from other algorithms.

According to the results in Table 4, LSA-SM can find the exact optimal value in unimodal benchmark functions, indicating that LSA-SM has strong robustness and high convergence accuracy. For functions f_{01} – f_{04} , the optimal

fitness value, mean fitness value, and standard deviation are better than other algorithms. For function f_{05} , the results of LSA-SM are the same as GSA, GWO, and WOA and better than LSA, BA, FPA, and GOA. The above analysis shows that LSA-SM has good stability and strong exploitation ability to solve unimodal benchmark functions.

In the results of multimodal functions in Tables 5 and 6, LSA-SM can find the global optimal solution or locate a near-global optimal value, indicating that LSA-SM has the ability to escape from poor local optimum. Table 5 is the experimental results of high-dimensional multimodal functions. For functions f_{07} – f_{10} , LSA-SM is ranked first among the eight algorithms, and its optimal fitness value, mean fitness value, and standard deviation have the highest accuracy. For function f_{07} , LSA-SM and WOA reach the best global minimum and the standard deviation is zero; GWO also obtains the best global minimum, but its result is not stable. For function f_{08} , LSA-SM and WOA get the same optimal fitness value, but the average and standard deviation of LSA-SM is better than WOA. For function f_{09} , LSA-SM, GWO, and WOA can find the exact optimal value, but LSA-SM has the strongest stability. For function f_{10} , the LSA-SM results are much better than other algorithms. For function f_{06} , although the standard deviation of LSA-SM is worse than that of FPA and GSA, the optimal fitness value and mean fitness value are better than the seven comparison algorithms. Therefore, LSA-SM has strong global exploration ability and high computational accuracy for solving high-dimensional multimodal functions.

Table 6 shows the experimental results for fixed-dimension multimodal functions (also called low-dimensional multimodal functions). It can be seen that LSA-SM and some other algorithms are able to search for the exact optimal value for functions f_{11} – f_{18} , indicating that LSA-SM has high convergence accuracy to solve the low-dimensional multimodal functions. For functions f_{11} , f_{13} , and f_{16} – f_{18} , LSA-SM has the least standard deviation, which indicates that LSA-SM is more stable than other algorithms. For function f_{12} , the four evaluation indices of LSA-SM are the same as LSA and GSA, and standard deviation of the three algorithms is zero. For function f_{14} , the standard variance of LSA-SM is only worse than GSA. For function f_{15} , LSA-SM is ranked third, and its mean fitness value and standard deviation are worse than those of GSA and FPA. In short, LSA-SM has certain advantages in optimizing the low-dimensional multimodal functions.

In summary, in the unimodal benchmark functions, LSA-SM can converge to the exact solution, which shows that LSA-SM has high exploitation ability. In the multimodal benchmark functions, LSA-SM can avoid local optimal solution and has high global exploration ability. Moreover, from the “Rank” column in Tables 4–6, it can be seen that LSA-SM has a strong stability.

For the three groups of benchmark functions, Figures 5–22 are the convergence curves drawn from the logarithm of mean minimum, which makes the performance of algorithms more convincing. And Figures 23–40 are the ANOVA test of the global minimum from 20 independent runs. As can be seen from Figures 5–22, the convergence accuracy of LSA-SM

TABLE 4: Results of unimodal benchmark functions.

Benchmark function	Result	Algorithms								Rank
		LSA-SM	LSA	BA	FPA	GSA	GWO	WOA	GOA	
f_{01} ($D = 30$)	Best	0	$2.19E - 36$	0.001533	289.7791	$1.53E - 17$	$4.18E - 73$	$9.3E - 189$	0.01471	1
	Worst	0	$9.81E - 22$	2523.855	694.2996	$2.99E - 17$	$5.47E - 70$	$3.4E - 167$	1.493134	
	Mean	0	$5.43E - 23$	590.6185	480.5182	$2.08E - 17$	$1.5E - 70$	$1.7E - 168$	0.357008	
	Std.	0	$2.19E - 22$	735.4083	123.384	$4.33E - 18$	$1.66E - 70$	0	0.385466	
f_{02} ($D = 30$)	Best	0	$2.8E - 09$	0.368079	15.0405	$1.49E - 08$	$2.63E - 42$	$1.2E - 117$	0.076814	1
	Worst	0	0.176046	2480.373	33.20639	$2.97E - 08$	$2.08E - 40$	$2.7E - 109$	3.855291	
	Mean	0	0.011954	149.0774	23.97925	$2.26E - 08$	$5.04E - 41$	$1.5E - 110$	1.493483	
	Std.	0	0.039601	550.5261	4.685298	$4.31E - 09$	$5.33E - 41$	$5.9E - 110$	1.157461	
f_{03} ($D = 30$)	Best	0	0.028542	1745.765	205.515	122.5354	$9.82E - 24$	125.8299	359.1287	1
	Worst	0	1.928264	28399.24	646.8589	414.9333	$1.78E - 19$	15797.51	2419.51	
	Mean	0	0.283358	7618.061	474.6606	224.0688	$1.38E - 20$	5423.81	989.0807	
	Std.	0	0.425606	6589.569	126.6975	90.38385	$4.02E - 20$	4804.579	532.2332	
f_{04} ($D = 30$)	Best	0	0.003176	20.29867	14.46131	$2.41E - 09$	$2.55E - 19$	$1.1E - 07$	2.665813	1
	Worst	0	0.55117	47.55456	21.29838	$4.45E - 09$	$4.43E - 16$	51.58371	13.08223	
	Mean	0	0.13218	28.95962	17.28069	$3.33E - 09$	$3.75E - 17$	10.91266	7.064873	
	Std.	0	0.139895	6.227619	2.001093	$5.59E - 10$	$9.79E - 17$	13.4977	2.515921	
f_{05} ($D = 30$)	Best	0	0	5344	388	0	0	0	8	1
	Worst	0	6	22107	807	0	0	0	44	
	Mean	0	2.4	12875.1	532.95	0	0	0	18.4	
	Std.	0	1.353358	4518.075	109.0174	0	0	0	9.34936	

TABLE 5: Results of multimodal benchmark functions.

Benchmark function	Result	Algorithms								Rank
		LSA-SM	LSA	BA	FPA	GSA	GWO	WOA	GOA	
f_{06} ($D = 30$)	Best	-9825.56	-9351.81	-8496	-7810.43	-4256.89	-7930.68	-9067.96	-8968.99	3
	Worst	-7337.76	-7040.88	-4751.55	-6948.99	-2499.65	-5228.28	-5969.66	-6138.18	
	Mean	-8288.62	-7972.7	-6597.79	-7451.38	-2994.2	-6319.63	-7273.86	-7230.55	
	Std.	579.2631	634.6361	978.568	225.2153	473.0903	734.5232	940.8279	705.6453	
f_{07} ($D = 30$)	Best	0	36.81344	41.99486	110.0765	9.949591	0	0	21.89829	1
	Worst	0	110.4402	133.5967	163.6662	25.86893	$5.68E - 14$	0	127.3554	
	Mean	0	67.75659	81.00623	132.9833	16.56607	$8.53E - 15$	0	66.23884	
	Std.	0	17.36836	34.96265	14.67229	3.782151	$2.08E - 14$	0	24.06253	
f_{08} ($D = 30$)	Best	8.88E - 16	1.340421	13.28169	4.245114	$2.72E - 09$	$7.99E - 15$	8.88E - 16	1.779031	1
	Worst	$4.44E - 15$	4.623521	16.48475	7.669989	$4.07E - 09$	$1.51E - 14$	$7.99E - 15$	4.92031	
	Mean	1.07E - 15	2.641427	14.91377	5.537861	$3.55E - 09$	$1.4E - 14$	$4.26E - 15$	2.870023	
	Std.	7.94E - 16	0.890342	0.897401	0.933813	$3.8E - 10$	$2.33E - 15$	$2.15E - 15$	0.679676	
f_{09} ($D = 30$)	Best	0	$1.11E - 16$	31.60788	3.744125	1.952645	0	0	0.19975	1
	Worst	0	0.041831	308.7602	8.961942	8.369517	0.020272	0.029831	0.535447	
	Mean	0	0.006156	97.01063	5.715779	4.129179	0.001419	0.001492	0.380767	
	Std.	0	0.010115	60.57474	1.153166	1.834566	0.004793	0.00667	0.110967	
f_{10} ($D = 30$)	Best	2.15E - 32	$7.83E - 24$	6.338175	6.64185	$7.74E - 20$	$1.26E - 06$	0.000202	0.837028	1
	Worst	$5.13E - 29$	2.597822	42.24743	17.12305	0.310964	0.079032	0.006291	8.283224	
	Mean	4.71E - 30	0.493764	22.3022	9.904098	0.051832	0.02644	0.001012	4.515233	
	Std.	1.4E - 29	0.721976	9.315506	2.499151	0.078873	0.017096	0.001486	1.725611	

TABLE 6: Results of fixed-dimension multimodal benchmark functions.

Benchmark function	Result	Algorithms								Rank
		LSA-SM	LSA	BA	FPA	GSA	GWO	WOA	GOA	
f_{11} ($D = 4$)	Best	0.000307	0.000307	0.000554	0.000309	0.000956	0.000307	0.000308	0.000658	1
	Worst	0.000307	0.001325	0.001415	0.000451	0.003509	0.020363	0.001292	0.020363	
	Mean	0.000307	0.000557	0.000937	0.000343	0.001889	0.002359	0.000523	0.003398	
	Std.	1.50E – 19	0.000432	0.000286	3.86E – 05	0.000556	0.006161	0.000244	0.00625	
f_{12} ($D = 2$)	Best	0.397887	0.397887	0.397887	0.397887	0.397887	0.397887	0.397887	0.397887	1
	Worst	0.397887	0.397887	0.397887	0.397888	0.397887	0.397889	0.397888	0.397887	
	Mean	0.397887	0.397887	0.397887	0.397887	0.397887	0.397888	0.397887	0.397887	
	Std.	0	0	1.67E – 08	1.3E – 07	0	4.48E – 07	7.32E – 08	1.57E – 12	
f_{13} ($D = 2$)	Best	3	3	3	3	3	3	3	3	1
	Worst	3	3	3.000012	3	3	84.00001	3.000006	3	
	Mean	3	3	3.000003	3	3	7.050002	3.000001	3	
	Std.	6.2E – 16	9E – 16	2.91E – 06	4.89E – 15	1.99E – 15	18.11215	1.88E – 06	3.57E – 13	
f_{14} ($D = 3$)	Best	–3.86278	–3.86278	–3.86278	–3.86278	–3.86278	–3.86278	–3.86278	–3.86278	2
	Worst	–3.86278	–3.86278	–3.86277	–3.86278	–3.86278	–3.8549	–3.86275	–3.86265	
	Mean	–3.86278	–3.86278	–3.86278	–3.86278	–3.86278	–3.86149	–3.86278	–3.86277	
	Std.	2.26E – 15	2.28E – 15	2.89E – 06	1.55E – 14	1.99E – 15	0.002689	7.28E – 06	2.97E – 05	
f_{15} ($D = 6$)	Best	–3.322	–3.322	–3.32196	–3.32162	–3.322	–3.32199	–3.32199	–3.322	3
	Worst	–3.2031	–3.2031	–3.20269	–3.31964	–3.322	–3.13764	–3.17039	–3.19678	
	Mean	–3.29822	–3.28038	–3.26233	–3.321	–3.322	–3.25826	–3.27509	–3.26187	
	Std.	0.048793	0.058182	0.060992	0.000524	4.56E – 16	0.066833	0.065901	0.061704	
f_{16} ($D = 4$)	Best	–10.1532	–10.1532	–10.1532	–10.1532	–10.1532	–10.1531	–10.1532	–10.1532	1
	Worst	–10.1532	–2.63047	–2.63046	–10.1532	–2.68286	–5.10019	–5.0552	–2.63047	
	Mean	–10.1532	–9.01692	–5.76334	–10.1532	–6.95803	–9.90023	–8.87863	–7.51533	
	Std.	3.43E – 15	2.382088	3.432055	1.01E – 06	3.451129	1.129812	2.264804	3.400665	
f_{17} ($D = 4$)	Best	–10.4029	–10.4029	–10.4029	–10.4029	–10.4029	–10.4029	–10.4029	–10.4029	1
	Worst	–10.4029	–3.7243	–1.83759	–10.4023	–10.4029	–10.4023	–5.08767	–1.83759	
	Mean	–10.4029	–8.87373	–5.70013	–10.4029	–10.4029	–10.4026	–9.86979	–7.58261	
	Std.	3.21E – 15	2.740302	3.599584	0.000134	3.58E – 15	0.000158	1.635458	3.628979	
f_{18} ($D = 4$)	Best	–10.5364	–10.5364	–10.5364	–10.5364	–10.5364	–10.5363	–10.5364	–10.5364	1
	Worst	–10.5364	–2.42173	–1.85948	–10.5323	–10.5364	–10.5356	–2.80662	–2.42734	
	Mean	–10.5364	–7.99514	–6.31646	–10.5359	–10.5364	–10.536	–7.64193	–8.0586	
	Std.	1.78E – 15	3.601864	3.969739	0.000913	1.95E – 15	0.000177	3.000784	3.527787	

is closer to the global optimal solution than LSA and other comparison algorithms in functions f_{01} – f_{11} . For functions f_{12} – f_{14} , the convergence curves of the eight algorithms finally reach the global minimum. For function f_{15} , FPA and GSA have better convergence values than LSA-SM, but their values are not much different. For functions f_{16} and f_{17} , LSA-SM and FPA eventually converge to global optimum. For function f_{18} , LSA-SM and GSA converge to the global optimal value. In addition, LSA-SM converges faster than other algorithms in all convergence graphs. As shown in Figures 23–40, it is easy to find that the standard deviation of LSA-SM is much smaller except function f_{06} , and the number of outliers is less than that of other algorithms. Moreover, the median of the LSA-SM (indicated by a red line) is closer to the best solution in f_{06} . This proves that LSA-SM is highly

stable. In conclusion, the proposed LSA-SM in this paper has the characteristics of fast convergence rate, high convergence accuracy, and strong stability.

In this section, in order to verify the optimization performance of the LSA-SM algorithm, 18 benchmark functions are used to simulate the experiment, and the results obtained by LSA-SM are compared with LSA, BA, FPA, GSA, GWO, WOA, and GOA. These benchmark functions include high-dimensional unimodal, high-dimensional multimodal, and fixed-dimension multimodal functions to verify the convergence, exploration, exploitation, and local optimal avoidance of the algorithms. The nonparametric Wilcoxon rank test is performed on the results of LSA-SM and other comparison algorithms to determine whether the proposed LSA-SM has different statistical significance from other algorithms. At the

TABLE 7: Statistical comparison between LSA-SM and the other seven algorithms.

F	LSA		BA		FPA		GSA		GWO		WOA		GOA	
	p value	h	p value	h	p value	h	p value	h	p value	h	p value	h	p value	h
f_{01}	$8.01E-09$	1	$8.01E-09$	1	$8.01E-09$	1	$8.01E-09$	1	$8.01E-09$	1	$8.01E-09$	1	$8.01E-09$	1
f_{02}	$8.01E-09$	1	$8.01E-09$	1	$8.01E-09$	1	$8.01E-09$	1	$8.01E-09$	1	$8.01E-09$	1	$8.01E-09$	1
f_{03}	$8.01E-09$	1	$8.01E-09$	1	$8.01E-09$	1	$8.01E-09$	1	$8.01E-09$	1	$8.01E-09$	1	$8.01E-09$	1
f_{04}	$8.01E-09$	1	$8.01E-09$	1	$8.01E-09$	1	$8.01E-09$	1	$8.01E-09$	1	$8.01E-09$	1	$8.01E-09$	1
f_{05}	$2.6E-08$	1	$8.01E-09$	1	$8.01E-09$	1	N/A	0	N/A	0	N/A	0	$7.77E-09$	1
f_{06}	0.067868	0	$1.58E-06$	1	$1.8E-06$	1	$6.8E-08$	1	$2.96E-07$	1	0.001014	1	$3.29E-05$	1
f_{07}	$8.01E-09$	1	$8.01E-09$	1	$8.01E-09$	1	$7.86E-09$	1	0.080359	0	N/A	0	$8.01E-09$	1
f_{08}	$1.13E-08$	1	$1.13E-08$	1	$1.13E-08$	1	$1.13E-08$	1	$3.09E-09$	1	$3.11E-06$	1	$1.13E-08$	1
f_{09}	$7.99E-09$	1	$8.01E-09$	1	$8.01E-09$	1	$8.01E-09$	1	0.162587	0	0.342112	0	$8.01E-09$	1
f_{10}	$6.8E-08$	1	$6.8E-08$	1	$6.8E-08$	1	$6.69E-08$	1	$6.8E-08$	1	$6.8E-08$	1	$6.8E-08$	1
f_{11}	$6.59E-08$	1	$6.59E-08$	1	$6.59E-08$	1	$6.59E-08$	1	$6.59E-08$	1	$6.59E-08$	1	$6.59E-08$	1
f_{12}	N/A	0	$8.01E-09$	1	$8.01E-09$	1	N/A	0	$8.01E-09$	1	$8.01E-09$	1	$7.96E-09$	1
f_{13}	0.893552	0	$2.94E-08$	1	$4.64E-08$	1	$1.3E-06$	1	$2.94E-08$	1	$2.94E-08$	1	$2.94E-08$	1
f_{14}	0.342112	0	$1.13E-08$	1	$6.84E-08$	1	$9.17E-05$	1	$1.13E-08$	1	$1.13E-08$	1	$1.13E-08$	1
f_{15}	0.22227	0	$9.22E-06$	1	0.000892	1	0.164334	0	$9.22E-06$	1	$4.11E-05$	1	$9.22E-06$	1
f_{16}	0.104553	0	$1.94E-08$	1	$1.94E-08$	1	$1.19E-07$	1	$1.94E-08$	1	$1.94E-08$	1	$1.94E-08$	1
f_{17}	0.663441	0	$3.3E-08$	1	$3.3E-08$	1	0.041961	1	$3.3E-08$	1	$3.3E-08$	1	$3.3E-08$	1
f_{18}	0.001374	1	$1.13E-08$	1	$1.13E-08$	1	$2.08E-08$	1	$1.13E-08$	1	$1.13E-08$	1	$1.13E-08$	1

FIGURE 4: Flowchart of the LSA-SM.

FIGURE 5: $D = 30$, convergence curves for f_{01} .

same time, the convergence curves and ANOVA test graphs of the algorithms are given.

From the results of the unimodal test functions, it is known that LSA-SM obtains the exact solution and the standard deviation is zero, which indicates that LSA-SM has high exploitation and strong stability. The LSA-SM achieves global minimum or near-global minimum in multimodal test functions, which proves that it has high exploration and avoids local optimum. From the convergence curves

FIGURE 6: $D = 30$, convergence curves for f_{02} . FIGURE 7: $D = 30$, convergence curves for f_{03} .

of test functions, it is concluded that LSA-SM has faster convergence rate than LSA and other algorithms. Moreover, in the standard deviation ranking, LSA-SM ranked first in the majority, indicating that the stability of LSA is very strong. This result is also clearly observed in the ANOVA test, where LSA-SM has a narrow interquartile range and the number of outliers is less than other algorithms. In addition, the nonparametric Wilcoxon rank test results show that LSA-SM has different statistical significance from other algorithms, which proves the effectiveness of LSA-SM results.

LSA-SM achieves better test results, mainly because SM and EOBL improve the exploration and exploitation of the

FIGURE 8: $D = 30$, convergence curves for f_{04} . FIGURE 9: $D = 30$, convergence curves for f_{05} .

hybrid algorithm. SM iteratively optimizes the step leaders in the worst position and promotes the local search of the algorithm, which makes the algorithm faster and closer to the global optimal solution. EOBL extends the search space to enhance the global search of the algorithm and avoid the algorithm falling into local optimization. In addition, according to the results in Tables 4–6, LSA, BA, FPA, and GOA are poor in high-dimensional unimodal and high-dimensional multimodal test functions, but the best value or approximate optimal value can be obtained in low-dimensional multimodal test functions, indicating that LSA, BA, FPA, and GOA have low exploitation and poor performance in optimizing high-dimensional functions. GSA,

FIGURE 10: $D = 30$, convergence curves for f_{06} . FIGURE 11: $D = 30$, convergence curves for f_{07} .

GWO, and WOA have achieved good results in most test functions, but their convergence accuracy needs to be further improved by strengthening local search. Moreover, from the convergence curve graphs, the early convergence rate of these seven algorithms is not very fast, mainly because the early search range of the algorithms is not large. In the ANOVA test graphs, the seven algorithms have a wider interquartile range and more outliers in some functions, indicating that the stability of these algorithms is not strong, mainly because they do not have a good balance between exploration and exploitation.

FIGURE 12: $D = 30$, convergence curves for f_{08} . FIGURE 13: $D = 30$, convergence curves for f_{09} .

In conclusion, the proposed LSA-SM in this paper has the characteristics of fast convergence rate, high convergence accuracy, and strong stability.

5. Constrained Engineering Design Optimization

Structural engineering optimization problems are complex; sometimes even the optimal solutions of interest do not exist [31]. These problems are usually nonlinear and have several equality and inequality constraints. In order to evaluate the performance of LSA-SM to solve practical application problems, we employed five classic constrained engineering design problems: tension/compression spring, cantilever beam, pressure vessel, welded beam, and three-bar truss.

FIGURE 14: $D = 30$, convergence curves for f_{10} . FIGURE 16: $D = 2$, convergence curves for f_{12} . FIGURE 15: $D = 4$, convergence curves for f_{11} . FIGURE 17: $D = 2$, convergence curves for f_{13} .

5.1. Tension/Compression Spring Design. This problem is from Arora and Belegundu, which aims to minimize the weight of a tension/compression spring, as shown in Figure 41. The minimum weight is subject to four constraints on minimum deflection (g_1), shear stress (g_2), surge frequency (g_3), and limits on outside diameter (g_4). There are three design variables: wire diameter $d(x_1)$, mean coil diameter $D(x_2)$, and the number of active coils $N(x_3)$ [23]. The mathematical formulas involved are as follows:

$$\text{Minimize: } f(X) = (x_3 + 2)x_2x_1^2,$$

$$\text{Subject to: } g_1(X) = 1 - \frac{x_2^3x_3}{71,785x_1^4} \leq 0,$$

$$g_2(X)$$

$$= \frac{4x_2^2 - x_1x_2}{12,566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1$$

$$\leq 0,$$

$$g_3(X) = 1 - \frac{140.45x}{x_2^2x_3} \leq 0,$$

$$g_4(X) = \frac{x_1 + x_2}{1.5} - 1 \leq 0,$$

(14)

where the variable regions are $0.05 \leq x_1 \leq 2.00$, $0.25 \leq x_2 \leq 1.30$, and $2.00 \leq x_3 \leq 15.0$. Table 8 compares the

FIGURE 18: $D = 3$, convergence curves for f_{14} . FIGURE 20: $D = 4$, convergence curves for f_{16} . FIGURE 19: $D = 6$, convergence curves for f_{15} . FIGURE 21: $D = 4$, convergence curves for f_{17} .

best optimization results obtained using LSA-SM with those reported in other literatures using different methods. We can see that the minimum weight obtained by LSA-SM to solve the tension/compression spring design problem is better than most of the methods except ABC, in which the minimum weight of LSA-SM is 0.01266524 and that of ABC is 0.012665.

5.2. Cantilever Beam Design. Cantilever beam consists of five square hollow blocks, as shown in Figure 42. The beam is rigidly supported at node 1, and there is a given vertical force acting at node 6. The problem is related to weight optimization, and the design variables are the height (or width) of the different beam elements, the number of which

is five, and the thickness is constant [31]. The detailed description of the cantilever beam problem is as follows:

$$\begin{aligned} \text{Minimize: } f(X) &= 0.0624(x_1 + x_2 + x_3 + x_4 + x_5), \end{aligned} \quad (15)$$

$$\begin{aligned} \text{Subject to: } g(X) &= \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \\ &\leq 0, \end{aligned} \quad (16)$$

where $0.01 \leq x_j \leq 100$, $j = 1, 2, \dots, 5$. Comparison results for cantilever beam design problem are listed in Table 9. Hence,

FIGURE 22: $D = 4$, convergence curves for f_{18} . FIGURE 23: $D = 30$, ANOVA test of global minimum for f_{01} .

it can be concluded that the best results of LSA-SM clearly outperform the other methods and it achieves the overall best design of the 1.339958. In addition, the optimization result of SOS is 1.33996, which is close to the result of LSA-SM.

5.3. Pressure Vessel Design. The pressure vessel design is a cylindrical vessel whose ends are capped by a hemispherical head as shown in Figure 43. The objective is to minimize the total cost. There are four design variables, T_s (thickness of the shell, x_1), T_h (thickness of the head, x_2), R (inner radius, x_3), and L (length of the cylindrical section of the vessel, x_4) [25], and four constraints. The thicknesses T_s and T_h of the variables are discrete values which are integer multiples

FIGURE 24: $D = 30$, ANOVA test of global minimum for f_{02} . FIGURE 25: $D = 30$, ANOVA test of global minimum for f_{03} .

of 0.0625 inches, and R and L are continuous values. This question can be described as follows:

Minimize: $f(X)$

$$= 0.06224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3, \quad (17)$$

$$\text{Subject to: } g_1(X) = -x_1 + 0.0193x_3 \leq 0, \quad (18)$$

$$g_2(X) = -x_2 + 0.00954x_3 \leq 0, \quad (19)$$

$$g_3(X) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1,296,000 \leq 0, \quad (20)$$

$$g_4(X) = x_4 - 240 \leq 0, \quad (21)$$

where $1 \times 0.0625 \leq x_1, x_2 \leq 99 \times 0.0625$ and $10 \leq x_3, x_4 \leq 200$. The comparison of results of different methods pressure

TABLE 8: Best results of the spring design example by different methods.

Reference	Method	Optimum variables				$f(X)$
		d	D	N		
Coello [19]	GA	0.051480	0.351661	11.632201		0.0127048
Coello and Montes [20]	GA	0.051989	0.363965	10.890522		0.0126810
Montes and Coello [21]	ES	0.051643	0.355360	11.397926		0.012698
Parsopoulos and Vrahatis [22]	UPSO	NA	NA	NA		0.01312
He and Wang [23]	CPSO	0.051728	0.357644	11.244543		0.0126747
Huang et al. [24]	CDE	0.051609	0.354714	11.410831		0.0126702
Mahdavi et al. [25]	IHS	0.05115438	0.34987116	12.0764321		0.0126706
Akay and Karaboga [26]	ABC	0.051749	0.358179	11.203763		0.012665
Mirjalili [27]	MFO	0.051994457	0.36410932	10.868421862		0.0126669
Mirjalili et al. [28]	GWO	0.05169	0.356737	11.28885		0.012666
Baykasoğlu and Ozsoydan [29]	AFA	0.0516674837	0.3561976945	11.3195613646		0.0126653049
Gandomi et al. [30]	BA	0.05169	0.35673	11.2885		0.01267
Present study	LSA-SM	0.05170453	0.3570899	11.26718		0.01266524

ES: evolution strategies; UPSO: unified particle swarm optimization; CPSO: coevolutionary particle swarm optimization; CDE: coevolutionary differential evolution; IHS: improved harmony search; MFO: moth-flame optimization algorithm; AFA: adaptive firefly algorithm; NA: there is no relevant data.

FIGURE 26: $D = 30$, ANOVA test of global minimum for f_{04} . FIGURE 27: $D = 30$, ANOVA test of global minimum for f_{05} .

vessel design is shown in Table 10. It can be seen that the minimum total cost of LSA-SM is 5942.6966, which is only worse than IHS in the listed methods. This result indicates that LSA-SM has successfully obtained the solution.

5.4. Welded Beam Design. Welded beam design is illustrated in Figure 44. This problem is designed for the minimum cost subject to constraints on shear stress (τ), bending stress in the beam (σ), buckling load on the bar (P_c), end deflection of the beam (δ), and side constraints. There are four design variables: $h(x_1)$, $l(x_2)$, $t(x_3)$, and $b(x_4)$ [24]. The problem can be formulated as follows:

$$\begin{aligned}
 &\text{Minimize: } f(X) \\
 &= 1.10471x_1^2x_2 \\
 &+ 0.04811x_3x_4(14.0 + x_2),
 \end{aligned} \tag{22}$$

FIGURE 28: $D = 30$, ANOVA test of global minimum for f_{06} .

TABLE 9: Best results of the cantilever beam design by different methods.

Method	Optimum variables					$f(X)$
	x_1	x_2	x_3	x_4	x_5	
CONLIN [31]	6.0100	5.3000	4.4900	3.4900	2.1500	NC
MMA [31]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
CS [31]	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999
GCA_I [32]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
GCA_II [32]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
SOS [33]	6.01878	5.30344	4.49587	3.49896	2.15564	1.33996
MFO [27]	5.98487177	5.316726924	4.49733258	3.51361646	2.16162029	1.33998808
LSA-SM	6.021636	5.310859	4.490882	3.497403	2.152906	1.339958

CONLIN: CONvex LINearization; MMA: method of moving asymptotes; CS: cuckoo search; GCA: generalized convex approximation; SOS: symbiotic organisms search; NC: not converging.

TABLE 10: Best results of the pressure vessel design by different methods.

Reference	Method	Optimum variables				$f(X)$
		T_s	T_h	R	L	
Coello [19]	GA	0.812500	0.437500	40.323900	200.000000	6288.7445
Coello and Montes [20]	GA	0.812500	0.437500	42.097398	176.654050	6059.9463
He and Wang [23]	CPSO	0.812500	0.437500	42.091266	176.746500	6061.0777
Huang et al. [24]	CDE	0.812500	0.437500	42.098411	176.746500	6059.7340
Mahdavi et al. [25]	IHS	0.75	0.375	38.86010	221.36553	5849.76169
Akay and Karaboga [26]	ABC	0.812500	0.437500	42.098446	176.636596	6059.714339
Gandomi et al. [30]	BA	0.8125	0.4375	42.0984456	176.6365958	6059.7143348
Gandomi et al. [31]	CS	0.8125	0.4375	42.0984456	176.6365958	6059.7143348
Baykasoglu and Ozsoydan [29]	AFA	0.8125	0.4375	42.09844611	176.6365894	6059.7142719
Kaveh and Talatahari [34]	ACO	0.8125	0.4375	42.103624	176.572656	6059.0888
Mezura-Montes and Coello [21]	ES	0.8125	0.4375	42.098087	176.640518	6059.7456
Mirjalili [27]	MFO	0.8125	0.4375	42.098445	176.636596	6059.7143
Rao et al. [35]	TLBO	NA	NA	NA	NA	6059.714335
Present study	LSA-SM	0.8103764	0.4005695	41.98842	178.0048	5942.6966

TLBO: teaching-learning-based optimization; NA: there is no relevant data.

$$\text{Subject to: } g_1(X) = \tau(X) - \tau_{\max} \leq 0, \quad (23)$$

$$g_2(X) = \sigma(X) - \sigma_{\max} \leq 0, \quad (24)$$

$$g_3(X) = x_1 - x_4 \leq 0, \quad (25)$$

$$\begin{aligned} g_4(X) &= 0.10471x_1^2 \\ &\quad + 0.04811x_3x_4(14.0 + x_2) - 5.0 \\ &\leq 0, \end{aligned} \quad (26)$$

$$g_5(X) = 0.125 - x_1 \leq 0, \quad (27)$$

$$g_6(X) = \delta(X) - \delta_{\max} \leq 0, \quad (28)$$

$$g_7(X) = P - P_c(X) \leq 0, \quad (29)$$

$$\begin{aligned} \text{where } \tau(X) &= \sqrt{(\tau')^2 + 2\tau'\tau''(x_2/2R) + (\tau'')^2}, \quad \tau' = \\ P/\sqrt{2}x_1x_2, \quad \tau'' &= MR/J, \quad M = P(L + x_2/2), \quad R = \end{aligned}$$

$$\begin{aligned} &\sqrt{x_2^2/4 + ((x_1 + x_3)/2)^2}, \quad J = 2\{\sqrt{2}x_1x_2[x_2^2/12 + ((x_1 + x_3)/ \\ &2)^2]\}, \quad \sigma(X) = 6PL/x_4x_3^2, \quad \delta(X) = 4PL^3/Ex_3^3x_4, \quad P_c(X) = \\ &(4.013E\sqrt{x_3^2x_4^6/36/L^2})(1 - (x_3/2L)\sqrt{E/4G}), \quad P = 6000 \text{ lb}, \quad L = \\ &14 \text{ in}, \quad E = 30 \times 10^6 \text{ psi}, \quad G = 12 \times 10^6 \text{ psi}, \quad \tau_{\max} = 13,600 \text{ psi}, \\ &\sigma_{\max} = 30,000 \text{ psi}, \quad \delta_{\max} = 0.25 \text{ in}, \quad \text{and } X = (x_1, x_2, x_3, x_4), \\ &(0.1, 0.1, 0.1, 0.1) \leq X \leq (2, 10, 10, 2). \end{aligned}$$

Table 11 compares the optimization results of LSA-SM and other optimization methods found in the literature, where LSA-SM achieves solutions that are superior to all other methods with minimum cost of 1.695247 for welded beam design problem.

5.5. Three-Bar Truss Design. Figure 45 shows a three-bar truss structure with the aim of achieving the minimum weight subjected to stress, deflection, and buckling constraints and evaluating the optimal cross sectional area (A_1, A_2) [27, 31]. This question has two variables and three constraints, involving the mathematical formulas as follows:

$$\text{Minimize: } f(X) = (2\sqrt{2}x_1 + x_2) * l, \quad (30)$$

TABLE II: Best results of the welded beam design by different methods.

Reference	Method	h	Optimum variables		b	$f(X)$
			l	t		
Deb [36]	GA	0.2489	6.1730	8.1789	0.2533	2.4328
Coello [19]	GA	0.208800	3.420500	8.997500	0.210000	1.748309
Coello and Montes [20]	GA	0.205986	3.471328	9.020224	0.206480	1.728226
Gandomi et al. [30]	BA	0.2015	3.562	9.0414	0.2057	1.7312
Akay and Karaboga [26]	ABC	0.205730	3.470489	9.036624	0.205730	1.724852
Parsopoulos and Vrahatis [22]	UPSO	0.2407	6.4851	8.2399	0.2497	2.4426
He and Wang [23]	CPSO	0.202369	3.544214	9.048210	0.205723	1.728024
Huang et al. [24]	CDE	0.203137	3.542998	9.033498	0.206179	1.733462
Mahdavi et al. [25]	IHS	0.20573	3.47049	9.03662	0.20573	1.7248
Mirjalili [27]	MFO	0.2057	3.4703	9.0364	0.2057	1.72452
Baykasoğlu and Ozsoydan [29]	AFA	0.205730	3.470489	9.036624	0.205730	1.724852
Kaveh and Talatahari [37]	CSS	0.2058	3.4681	9.0380	0.2057	1.7249
Present study	LSA-SM	0.2057296	3.253120	9.036624	0.2057296	1.695247

CSS: charged system search.

FIGURE 29: $D = 30$, ANOVA test of global minimum for f_{07} . FIGURE 30: $D = 30$, ANOVA test of global minimum for f_{08} .

$$\text{Subject to: } g_1(X) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0, \quad (31)$$

$$g_2(X) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0, \quad (32)$$

$$g_3(X) = \frac{1}{\sqrt{2}x_2 + x_1}P - \sigma \leq 0, \quad (33)$$

where the variable ranges are $0 \leq x_1, x_2 \leq 1$ and $l = 100$ cm, $P = 2$ KN/cm², and $\sigma = 2$ KN/cm². Table 12 lists the results of solving the three-bar truss design problem through the proposed LSA-SM and other eight different methods. The optimal target value of LSA-SM is 263.8958, which is close to the optimization results of Ray and Liew, DEDS, PSO-DE, and MBA (263.8958466, 263.8958434, 263.8958433, and 263.8958522, resp.), and LSA-SM is better than all the algorithms for solving the three-bar truss design problem.

FIGURE 31: $D = 30$, ANOVA test of global minimum for f_{09} .

TABLE 12: Best results of the three-bar truss design by different methods.

Reference	Method	Optimum variables		$f(X)$
		x_1	x_2	
Tsa [38]	a	0.788	0.408	263.68
Ray and Saini [39]	b	0.795	0.395	264.3
Ray and Liew [40]	SC	0.7886210370	0.4084013340	263.8958466
Gandomi et al. [31]	CS	0.78867	0.40902	263.9716
Zhang et al. [41]	DEDS	0.78867513	0.40824828	263.8958434
Liu et al. [42]	PSO-DE	0.7886751	0.4082482	263.8958433
Sadollah et al. [43]	MBA	0.7885650	0.4085597	263.8958522
Mirjalili [27]	MFO	0.788244770931922	0.409466905784741	263.895979682
Present study	LSA-SM	0.7886136	0.4084224	263.8958

a: a method to solve nonlinear fractional programming; b: a swarm strategy; SC: society and civilization; DEDS: differential evolution with dynamic stochastic selection; PSO-DE: hybridizing particle swarm optimization with differential evolution; MBA: mine blast algorithm.

FIGURE 32: $D = 30$, ANOVA test of global minimum for f_{10} .

In summary, LSA-SM has effectively solved five classical constrained engineering design problems, and its optimization results are superior to most of the methods in the literatures. These results show that LSA-SM has the outstanding ability to solve the problem of constrained nonlinear optimization in reality.

6. Conclusions

In the current study, a hybrid LSA-SM is proposed to solve the global optimization problem. The standard LSA algorithm is based on the natural phenomenon of lightning discharge to the ground, which has a fast convergence rate and simulates the process of lightning bifurcation to explore a better solution. In view of the poor convergence accuracy of the original LSA, a simplex method is added to improve its accuracy. Simplex method is a prominent method for local search. The accuracy of the LSA is improved by using the reflection, expansion, and compression operations of the SM to improve the step leaders with the worst fitness value. In addition, the elite opposition-based learning strategy is introduced to

FIGURE 33: $D = 4$, ANOVA test of global minimum for f_{11} .

expand the diversity of the population, to avoid LSA-SM falling into the local optimum. SM and EOBL are added to the LSA to balance the exploitation and exploration of the algorithm. The performance of LSA-SM is tested by 18 benchmark functions and compared with LSA, BA, FPA, GSA, GWO, WOA, and GOA. The experimental results show that LSA-SM can find the exact solution of most tested functions and has the characteristics of higher computational accuracy, faster convergence rate, and stronger stability. Finally, the hybrid LSA-SM algorithm is used to solve five engineering problems in reality. These problems are tension/compression spring, cantilever beam, pressure vessel, welded beam, and three-bar truss design problems. The results obtained by LSA-SM to solve these engineering design problems are much better than most of the algorithms in the literature, which proves that LSA-SM is suitable for solving constrained nonlinear optimization problems.

For future work, LSA-SM needs to further adjust the relevant parameters or increase the improvement strategy because LSA-SM does not find an accurate solution in some test functions. The proposed LSA-SM will be used to solve multiobjective optimization problems. In addition,

FIGURE 34: $D = 2$, ANOVA test of global minimum for f_{12} . FIGURE 35: $D = 2$, ANOVA test of global minimum for f_{13} . FIGURE 36: $D = 3$, ANOVA test of global minimum for f_{14} . FIGURE 37: $D = 6$, ANOVA test of global minimum for f_{15} . FIGURE 38: $D = 4$, ANOVA test of global minimum for f_{16} . FIGURE 39: $D = 4$, ANOVA test of global minimum for f_{17} .

FIGURE 40: $D = 4$, ANOVA test of global minimum for f_{18} .

FIGURE 41: The tension/compression spring design problem.

FIGURE 42: The cantilever beam design problem.

FIGURE 43: The pressure vessel design problem.

Shareef et al. have used LSA to solve the TSP problem with 20 cities [11]. In order to verify the performance of LSA-SM to solve the higher dimension problem, it will solve the large-scale TSP problem as the next study. Finally, LSA-SM will be applied to the optimal scheduling of tasks on cloud resources, and the results obtained by LSA-SM are compared

FIGURE 44: The welded beam design problem.

FIGURE 45: The three-bar truss design problem.

with the existing algorithms on the problem. The optimal scheduling of tasks in the cloud computing environment is a hot issue. In cloud computing, a number of tasks may need to be scheduled on different virtual machines in order to minimize makespan and increase system utilization. Task scheduling problem is NP-complete; hence finding an exact solution is intractable especially for large task sizes [55].

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by National Science Foundation of China under Grants nos. 61563008 and 61463007 and Project of Guangxi University for Nationalities Science Foundation under Grant no. 2016GXNSFAA380264.

References

- [1] T. Weise, M. Zapf, R. Chiong, and A. J. Nebro, "Why is optimization difficult," in *Nature-Inspired Algorithms for Optimisation, Studies in Computational Intelligence*, vol. 193, pp. 1-50, Springer, Berlin, Germany, 2009.

- [2] J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, USA, 1975.
- [3] M. Dorigo, *Optimization, Learning and Natural Algorithms [Ph.D. thesis]*, Politecnico di Milano, Milan, Italy, 1992.
- [4] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Piscataway, NJ, USA, 1995.
- [5] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.
- [6] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Applied Soft Computing Journal*, vol. 8, no. 1, pp. 687–697, 2008.
- [7] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*, pp. 210–214, IEEE Publications, 2009.
- [8] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, J. R. Gonzalez, D. A. Pelta, C. Cruz, G. Terrazas, and N. Krasnogor, Eds., vol. 284 of *Studies in Computational Intelligence*, pp. 65–74, Springer, Berlin, Germany, 2010.
- [9] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 52–67, 2002.
- [10] A. Hatamlou, "Black hole: a new heuristic optimization approach for data clustering," *Information Sciences. An International Journal*, vol. 222, pp. 175–184, 2013.
- [11] H. Shareef, A. A. Ibrahim, and A. H. Mutlag, "Lightning search algorithm," *Applied Soft Computing Journal*, vol. 36, pp. 315–333, 2015.
- [12] H. Shareef, A. H. Mutlag, and A. Mohamed, "A novel approach for fuzzy logic PV inverter controller optimization using lightning search algorithm," *Neurocomputing*, vol. 168, pp. 435–453, 2015.
- [13] J. A. Ali, M. A. Hannan, and A. Mohamed, "A novel quantum-behaved lightning search algorithm approach to improve the fuzzy logic speed controller for an induction motor drive," *Energies*, vol. 8, no. 11, pp. 13112–13136, 2015.
- [14] M. M. Islam, H. Shareef, A. Mohamed, and A. Wahyudie, "A binary variant of lightning search algorithm: BLSA," *Soft Computing*, pp. 1–20, 2016.
- [15] R. Sirjani and H. Shareef, "Parameter extraction of solar cell models using the lightning search algorithm in different weather conditions," *Journal of Solar Energy Engineering*, vol. 138, no. 4, 2016.
- [16] A. H. Mutlag, A. Mohamed, and H. Shareef, "A nature-inspired optimization-based optimum fuzzy logic photovoltaic inverter controller utilizing an eZdsp F28335 board," *Energies*, vol. 9, no. 3, article 120, 2016.
- [17] M. S. Ahmed, A. Mohamed, R. Z. Homod, and H. Shareef, "Hybrid LSA-ANN based home energy management scheduling controller for residential demand response strategy," *Energies*, vol. 9, no. 9, article 716, 2016.
- [18] H. Faris, I. Aljarah, N. Al-Madi, and S. Mirjalili, "Optimizing the learning process of feedforward neural networks using lightning search algorithm," *International Journal on Artificial Intelligence Tools*, vol. 25, no. 6, article 1650033, 2016.
- [19] C. A. C. Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Computers in Industry*, vol. 41, no. 2, pp. 113–127, 2000.
- [20] C. A. C. Coello and E. M. Montes, "Constraint-handling in genetic algorithms through the use of dominance-based tournament selection," *Advanced Engineering Informatics*, vol. 16, no. 3, pp. 193–203, 2002.
- [21] E. Mezura-Montes and C. A. Coello Coello, "An empirical study about the usefulness of evolution strategies to solve constrained optimization problems," *International Journal of General Systems*, vol. 37, no. 4, pp. 443–473, 2008.
- [22] K. E. Parsopoulos and M. N. Vrahatis, "Unified particle swarm optimization for solving constrained engineering optimization problems," in *Advances in Natural Computation*, vol. 3612 of *Lecture Notes in Computer Science*, pp. 582–591, Springer, Berlin, Germany, 2005.
- [23] Q. He and L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 1, pp. 89–99, 2007.
- [24] F. Z. Huang, L. Wang, and Q. He, "An effective co-evolutionary differential evolution for constrained optimization," *Applied Mathematics and Computation*, vol. 186, no. 1, pp. 340–356, 2007.
- [25] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1567–1579, 2007.
- [26] B. Akay and D. Karaboga, "Artificial bee colony algorithm for large-scale problems and engineering design optimization," *Journal of Intelligent Manufacturing*, vol. 23, no. 4, pp. 1001–1014, 2012.
- [27] S. Mirjalili, "Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.
- [28] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [29] A. Baykasoğlu and F. B. Ozsoydan, "Adaptive firefly algorithm with chaos for mechanical design optimization problems," *Applied Soft Computing*, vol. 36, pp. 152–164, 2015.
- [30] A. H. Gandomi, X. S. Yang, A. H. Alavi, and S. Talatahari, "Bat algorithm for constrained optimization tasks," *Neural Computing and Applications*, vol. 22, no. 6, pp. 1239–1255, 2013.
- [31] A. H. Gandomi, X. S. Yang, and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with Computers*, vol. 29, no. 1, pp. 17–35, 2013.
- [32] H. Chickermane and H. C. Gea, "Structural optimization using a new local approximation method," *International Journal for Numerical Methods in Engineering*, vol. 39, no. 5, pp. 829–846, 1996.
- [33] M.-Y. Cheng and D. Prayogo, "Symbiotic organisms search: a new metaheuristic optimization algorithm," *Computers & Structures*, vol. 139, pp. 98–112, 2014.
- [34] A. Kaveh and S. Talatahari, "An improved ant colony optimization for constrained engineering design problems," *Engineering Computations*, vol. 27, no. 1, pp. 155–182, 2010.
- [35] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.
- [36] K. Deb, "Optimal design of a welded beam via genetic algorithms," *AIAA Journal*, vol. 29, no. 11, pp. 2013–2015, 1991.
- [37] A. Kaveh and S. Talatahari, "A novel heuristic optimization method: charged system search," *Acta Mechanica*, vol. 213, no. 3–4, pp. 267–289, 2010.

- [38] J.-F. Tsai, "Global optimization of nonlinear fractional programming problems in engineering design," *Engineering Optimization*, vol. 37, no. 4, pp. 399–409, 2005.
- [39] T. Ray and P. Saini, "Engineering design optimization using a swarm with an intelligent information sharing among individuals," *Engineering Optimization*, vol. 33, no. 6, pp. 735–748, 2001.
- [40] T. Ray and K. M. Liew, "Society and civilization: an optimization algorithm based on the simulation of social behavior," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 4, pp. 386–396, 2003.
- [41] M. Zhang, W. Luo, and X. Wang, "Differential evolution with dynamic stochastic selection for constrained optimization," *Information Sciences*, vol. 178, no. 15, pp. 3043–3074, 2008.
- [42] H. Liu, Z. Cai, and Y. Wang, "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization," *Applied Soft Computing Journal*, vol. 10, no. 2, pp. 629–640, 2010.
- [43] A. Sadollah, A. Bahreininejad, H. Eskandar, and M. Hamdi, "Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems," *Applied Soft Computing*, vol. 13, no. 5, pp. 2592–2612, 2013.
- [44] E. Davoodi, M. T. Hagh, and S. G. Zadeh, "A hybrid improved quantum-behaved particle swarm optimization-simplex method (IQPSOS) to solve power system load flow problems," *Applied Soft Computing Journal*, vol. 21, pp. 171–179, 2014.
- [45] A. S. El-Wakeel, "Design optimization of PM couplings using hybrid particle swarm optimization-simplex method (PSO-SM) algorithm," *Electric Power Systems Research*, vol. 116, pp. 29–35, 2014.
- [46] H. R. Tizhoosh, "Opposition-based learning: a new scheme for machine intelligence," in *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation, International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA/IAWTIC '05)*, vol. 1, pp. 695–701, Vienna, Austria, 2005.
- [47] H. Wang, Z. Wu, and S. Rahnamayan, "Enhanced opposition-based differential evolution for solving high-dimensional continuous optimization problems," *Soft Computing*, vol. 15, no. 11, pp. 2127–2140, 2011.
- [48] Y. Zhou, R. Wang, and Q. Luo, "Elite opposition-based flower pollination algorithm," *Neurocomputing*, vol. 188, pp. 294–310, 2016.
- [49] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.
- [50] X.-S. Yang, "Test problems in optimization," in *Engineering Optimization: An Introduction with Metaheuristic Applications*, X.-S. Yang, Ed., John Wiley & Sons, 2010.
- [51] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Sciences*, vol. 213, pp. 267–289, 2010.
- [52] X. S. Yang, "Flower pollination algorithm for global optimization," in *Unconventional Computation and Natural Computation*, vol. 7445 of *Lecture Notes in Computer Science*, pp. 240–249, Springer, Berlin, Germany, 2012.
- [53] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
- [54] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: theory and application," *Advances in Engineering Software*, vol. 105, pp. 30–47, 2017.
- [55] M. Abdullahi, M. A. Ngadi, and S. M. Abdulhamid, "Symbiotic Organism Search optimization based task scheduling in cloud computing environment," *Future Generation Computer Systems*, vol. 56, pp. 640–650, 2016.