Enhancement of Bernstain-Search Differential Evolution Algorithm to Solve Constrained Engineering Problems

Hoda Zamani[1,2,3,*], Mohammad H. Nadimi-Shahraki[1,2,3], Shokooh Taghian[1,2], Mahdis Banaie-Dezfouli[1,2]

[1]Faculty of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran

[2]Big Data Research Center, Najafabad Branch, Islamic Azad University, Najafabad, Iran

[3]Intelligent Big Data Analysis in Medicine (IBDAM), Universal Scientific Education and Research Network (USERN), Najafabad, Iran

[*]Corresponding author: zamanie_hoda@ymail.com

Abstract— There are many constraints in solving the engineering optimization problems, through which finding a feasible solution is a challenging task. Although researchers have proposed some effective algorithms to cope with this challenge, most of their proposed algorithms suffer from the low diversity of the population and trapped by the local optima. In this paper, an enhancement of the bernstain-search differential evolution algorithm named EBSD is developed to solve the constrained engineering problems. In the EBSD, the trial pattern vector is improved, and the Chebyshev chaotic map is used to increase the diversity of the population. The EBSD algorithm is evaluated by four constrained engineering design problems, including pressure vessel, welded beam, tension/compression spring, and three-bar truss. In all experiments, EBSD is compared by the state-of-the-art swarm intelligence algorithms: CLPSO, DE/BBO, EEGWO, WDE, ChOA, and BSD. The experimental results show that the EBSD algorithm is very competitive compared to the state-of-the-art algorithms to solve these constrained engineering problems.

Keywords - Optimization; Metaheuristic Algorithms; Swarm intelligence algorithms; Bernstain-search differential evolution algorithm; Engineering design optimization problems.

I. INTRODUCTION

Metaheuristic algorithms belong to a family of approximate algorithms that have been developed for a wide range of applications such as medical [1-3], engineering [4-7], and the tourism industry [8]. These algorithms provide a near-optimal solution in a reasonable time without having to adapt to each problem deeply. Metaheuristic algorithms are inspired by mimicking biological or physical phenomena to solve the optimization problem. The metaheuristic algorithms are mostly classified into single-solution-based and population-based search algorithms [9, 10]. The single-solution-based search algorithms are mostly exploitation oriented and manipulate a single solution during the search process [9, 10]. The most popular single-solution based search algorithms are simulated annealing (SA) [11], tabu search (TS) [12], iterated local search (ILS) [13] and guided local search (GLS) [14]. These algorithms suffer from shortcomings, such as slow convergence and local optima trapping [9, 15]. On the other hand, the population-based search algorithms are mostly exploration oriented to evolve the population of solutions using meaningful search strategies. These algorithms have been extended and used more than single-solution-based algorithms. Then, the population-based search algorithms can be broadly classified into two categories: swarm intelligence (SI) and evolutionary algorithms (EA) [10, 16].

Swarm intelligence (SI) algorithms originated from the social behavior of species such as birds [17, 18], aquatic animals [19], terrestrial animals [20-22], and insects [23-26] in nature. Particle swarm optimization (PSO) is a pioneer contribution, originally developed by Kennedy and Eberhart [27], as a solution methodology to continuous nonlinear problems. It evolves particles (search agents) using the local and global search strategies to find the best promising area. Another popular SI algorithm is ant colony optimization (ACO) [23] that is proposed for solving discrete computational problems and is widely used in different real-world applications [19, 25, 28, 29]. The SI algorithms are more prevalent in engineering applications due to do not require gradient information, and also they proposed effective operators for bypassing the local optima [9, 16]. In the last years, many real-world problems have been solved using SI algorithms such as early diagnosis of coronary artery disease [30], breast cancer diagnosis [31], and automatic classification of brain strokes in CT images [32]. The SI algorithms were adapted by utilizing transfer functions [33] to solve various discrete optimization problems [31, 34, 35]. Meanwhile, evolutionary algorithms (EA) were inspired by Darwin's evolutionary theory of biological evolution. In EA algorithms, the population is randomly initialized using a uniform distribution in the search space, and then the trial vectors are generated by the recombination and mutation operators. Genetic algorithm (GA) [36], evolution strategy (ES) [37], and differential evolution (DE) [38] are the well-known EA algorithms. The recent and effective EA algorithms are bernstain-search differential evolution (BSD) [39], weighted differential evolution algorithm (WDE) [40], and effective multi-trial vector-based differential evolution (MTDE) [41]. Stagnation and diversity are two issues that significantly affect the performance of metaheuristic algorithms. In stagnation, the search strategies cannot generate a better solution forever, and the population cannot converge to the point [42]. Effective operators and meaningful search strategis enhance the population diversity and accelerate the convergence rate over the populations to a promising area. Over the years, considerable approaches have been developed to overcome the problem of stagnation and increasing diversity by enhancing search strategies, population structures, and communication topology. Some of these developments are successful to increase diversity, such as the grey artificial bee colony algorithm (GABC) [43], conscious neighborhood-based crow search algorithm (CCSA) [16], and diversity enhanced PSO with neighborhood search (DNSPSO) [44].

One of the essential applications of swarm intelligence and evolutionary algorithms is to solve engineering design optimization (EDO) problems. The EDO problems have different natures of objective functions, decision variables, non-linearity, and non-convexity of the constraint functions. The objective function is the essential component of an optimization problem that satisfies all problem constraints by choosing optimal decision variables. Each EDO problem has two types of restrictions, directly and indirectly, related to decision variables [15]. The direct constraints restrict the possible value of the decision variables in the specific range, but the indirect constraint the restrictions are defined in terms of formulas [15]. Thus, the constrained EDO problem is defined as an optimization process to find an optimal feasible solution that satisfies all the constraints. These problems aim to maximize or minimize the corresponding objective function. Assume S is the D-dimensional search space limited between lower (l) and upper (u) boundaries with q objective function. The parameter Ω denoted the feasible region in this search space such that $\Omega \subseteq S$. A solution with n decision variables in the feasible region Ω indicated as $\vec{x} = x_1, x_2, \dots, x_n$. Generally, the constrained EDO problem is formulated by Eq. (9). In this equation, the restrictions $g_j(\vec{X})$ and $h_j(\vec{X})$ are the value of j-th equality and inequality constraint, respectively. The parameters q and m are the number of equality and inequality constraints. With increasing complexity, these problems' search space is spotted with fragmented feasible regions [15]; hence, effective methods need to avoid infeasible decision variables and locally optimal possible areas [15, 45]. Several methods are proposed to prevent infeasible decision variables, such as removing, refinement, and penalty functions. The removal method eliminates each possible solution that does not satisfy the constraints. The refinement method refines infeasible solutions to render them feasible solutions to reach the globally optimal feasible region. The penalty functions add a penalty function to the objective function [15].

$$\begin{aligned} &optimize \quad \mathrm{F}(\vec{X}),\ \vec{X} = (x_1, x_2, \dots, x_n) \in R^n \qquad l_i \le x_i \le u_i \\ &Subject\ to \begin{cases} g_j(\vec{X}) \le 0 & j = 1, 2, \dots, q \\ h_j(\vec{X}) = 0 & j = q+1, \dots, m \end{cases} \end{aligned} \tag{9}$$

Although researchers have proposed effective algorithms to cope with this challenge, most of their proposed algorithms are trapped by the local optima. In this paper, an enhancement of the bernstain-search differential evolution algorithm named EBSD is developed to solve the constrained engineering problems. The proposed EBSD is evaluated using four engineering design problems for solving EDO problems. Furthermore, the obtained results were compared with the state-of-the-art SI and EA variants to prove the superiority of the EBSD algorithm over the contender algorithms in solving EDO problems. The experimental results show that the EBSD algorithm can achieve an optimum solution, thus improving solutions to reach the globally optimal feasible region and accelerate the convergence rate. The rest of the paper is organized as follows. Section II reviews the related algorithms for solving EDO problems. Section III describes the standard bernstain-search differential evolution algorithm (BSD). Section IV proposes the enhancement of bernstain-search differential evolution (EBSD) algorithm for solving EDO problems. The experimental evaluations are described in Section V. The conclusion is given in Section VI.

II. Related Works

Engineering Optimization promotes optimization techniques to achieve design or other goals in engineering [46-51]. Engineering design optimization (EDO) deals with different types of constraints that modify the shape of the search space. During the last decades, a wide variety of algorithms has been proposed based on gradient and non-gradient information to find reasonable solutions for solving EDO problems. The gradient-based information algorithms cannot differentiate between the local and global optima; hence they easily trap into local optima. The non-gradient based information algorithms (or metaheuristic) approximate the feasible solutions accurately, and their useful search operators are less likely to be trapped into the local minima. Metaheuristic algorithms proposed for solving EDO problems, based on inspiration, are mainly investigated in two different groups swarm intelligence (SI) and evolutionary (EA) algorithms. In the following, the state-of-the-art algorithms for solving EDO problems are reviewed.

A. Swarm intelligence algorithms to solve EDO problems

Recently, numerous SI algorithms have been proposed and improved in solving engineering problems. In this regard, some of the most proposed algorithms are whale optimization algorithm (WOA) [19], ant lion optimizer (ALO) [24], grey wolf optimizer (GWO) [22], crow search algorithm (CSA) [18], queuing search (QS) [52], and mine blast algorithm (MBA) [53]. Recently some improvement algorithms are proposed to increase the efficiency of canonical algorithms, such as improved grey wolf optimizer (IGWO) [6], Sine cosine grey wolf optimizer (SC-GWO) [54], cuckoo search and differential evolution (CSDE) [4], improved fruit fly optimization algorithm (FOA) based on the linear diminishing step and logistic chaos mapping (DSLC-FOA) [55], improved accelerated particle swarm optimization algorithm (IAPSO) [5], exploration-enhanced GWO (EEGWO) [21], improved vector PSO (IVPSO) [56]. Sayarshad used the bees algorithm (BA) for solving the material handling equipment (MHE) problem [57]. Many search strategies are developed to increase the efficiency of the canonical algorithms for solving constrained optimization problems. He and et al. proposed a co-evolutionary particle swarm optimization approach (CPSO) [45] for constrained optimization problems. In the CPSO, two kinds of swarms interactively evolved the population of solutions. Furthermore, multiple swarms are used to search for a promising area, and a single swarm is performed for evolving penalty factors during the optimization process. Ray and Liew [58] presented a swarm optimization algorithm with a multilevel information sharing methods to deal with constraints. Cagnina et al. [59] introduced a simple constraint particle swarm optimization (SiC-PSO) algorithm to solve the constrained EDO problems.

B. Evolutionary algorithms to solve EDO problems

Evolutionary algorithms are an efficient scheme to solve EDO problems. The performance of evolutionary algorithms using different constraint handling methods is investigated by Runarsson and Yao [60]. They proposed a stochastic ranking (SR) in the evolution strategy (ES) to balance the objective and penalty functions. Hamida and et al. [61] proposed the adaptive segregational constraint handling evolutionary algorithm (ASCHEA) for constrained optimization problems. A population-level adaptive penalty function is presented in the ASCHEA algorithm to address the constraints [61]. The simple multi-membered evolution strategy (SMES) [7] used a diversity mechanism based on infeasible solutions instead of a penalty function. The GA algorithm is a powerful optimization technique, and many variants of GA have been applied to a wide range of engineering optimization problems [62-64]. The PSO-GA is a new hybrid algorithm of SI and EA families for solving constrained optimization problems. In the PSO-GA [65] algorithm, the population of particles operates in the direction of improving the vectors using a modification of genetic operators. The multi-trial vector-based differential evolution (MTDE) [41] was proposed by Nadimi et al. for solving global optimization problems, and the applicability of MTDE is evaluated for solving the EDO problems.

III. Bernstain-Search Differential Evolution (BSD) Algorithm

This section's conceptual description of the bernstain-search differential evolution (BSD) [39] algorithm is provided in detail. The BSD algorithm belongs to the family of a universal differential evolution algorithm (uDE) with such properties as easily controllable, simple structured, non-recursive, highly efficient, fast, and practically parameter-free [39]. The BSD algorithm proposed a bijective mutation and crossover operators without any control-parameter tuning process. The crossover process is controlled randomly by using Bernstein polynomials. This algorithm uses different random number generators to generate evolutionary step size and produce efficient trial vectors using the span pattern vector and the best-obtained solution. The step-wise procedure for the implementation of the BSD algorithm is given as follows.

– Step 1: Randomly distributing N individuals in the search space

In the BSD algorithm, the candidate solutions are assumed as pattern vectors in the D dimension search space. As the Eq. (1) N distinct individuals are randomly distributed in a D dimensional search space between the upper (up) and lower (low) bounders.

$$P_{i,j} = \mathbf{U}\,(low_j\,,up_j\,), \qquad 1 \le i \le N, \qquad 1 \le j \le D \tag{1}$$

– Step 2: Evaluate the fitness (objective) function of the pattern vectors

The fitness value for each pattern vector fit_{Pi} is computed by inserting the decision variable values into the fitness function of the corresponding problem.

– Step 3: Set the best pattern vector

The best pattern vector in individuals is considered a global solution and defined by Eq. (2). In this equation, the parameters sol_P and $best_P$ are the best pattern vector's solution and best fitness function, respectively.

$$[sol_P, best_P\,] = [\text{ fit P}(\gamma)\,, \text{P}\,(\gamma)\,] \mid \text{fit P}(\gamma) = \min\,(fit_P\,) \mid 1 \le \gamma \le \text{N} \tag{2}$$

– Step 4: Compute the crossover ratio

The crossover ratio is determined using Eq. (3). In this equation, the initial value of parameter $M_{(i=1:N, j=1:D)}$ is equal to 0. The parameter ρ is computed using 2-nd degree Bernstain polynomials defined by Eq. (4).

$$M_{(i,u\ (1:\lceil\rho.D\rceil))} = 1 \qquad u = Permutation\ ([1 : D]) \tag{3}$$

$$\begin{aligned} &\text{Switch} \quad \kappa_0 \\ &\quad \text{Case } 1 \quad \rho = (1-\beta)^2 \\ &\quad \text{Case } 2 \quad \rho = 2 \times \beta \times (1-\beta) \\ &\quad \text{Case } 3 \quad \rho = 2 \times \beta \times (1-\beta) \\ &\text{end} \end{aligned} \tag{4}$$

$\beta \sim U\ (0, 1)$, $\kappa_0 = \lceil 3.\kappa_1^3 \rceil$, $\kappa_1 \sim U\ [0, 1]$, $\kappa_0 \in U\ (1 : 3)$

– Step 5: Compute the evolutionary step size

The evolutionary step size is defined using parameter F and is calculated by Eq. (5). In this equation, the parameters $\eta \sim \mathbf{U}\ (0, 1)$, and $\lambda \sim \mathbf{N}\ (0, 1)$ generate a new value using uniform and normal distribution in each dimension, respectively, and $(\cdot, \cdot)$ sized all-ones matrix $Q\ (\cdot, \cdot) = 1$.

$$F = \begin{cases} \left(\left[\eta^3_{(1,1:D)} \times \left|\lambda^3_{(1,1:D)}\right|\right]' \times Q_{(1,1:D)}\right) & \kappa_2 < \kappa_3 \\ \lambda^3_{(N,1)} \times Q_{(1,:D)} & \kappa_2 \geq \kappa_3 \end{cases} \tag{5}$$

– Step 6: Generate the BSD's trial pattern vector

The trial pattern vector (T_i) for the i-th individual is generated using Eq. (6). In this equation, the parameter E is computed by Eq. (7)

$$T = P + F \times M \times ((W^*)^3 \times E + (1-(W^*)^3) \times Best_P - P) \qquad W^*_{(1:N,1)} \sim \mathrm{U}\ (0,1) \tag{6}$$

$$E = w\ .P_{L1} + (1-W)\ .P_{L2} \qquad W_{(1:N,1:D)} \sim \mathrm{U}\ (0,1) \tag{7}$$

$$L_1 = Permutation\ ([1 : N]), \quad L_2 = Permutation\ ([1 : N]), \quad L_1 \neq L_2$$

– Step 7: Check the feasibility of the new trial pattern vector

In each generation, the feasibility of the new trial pattern vector of each individual is checked. If the new position is feasible, the individual updates its position. Otherwise, the trial pattern vector of individuals is updated using Eq. (8).

$$T_{ij} = low_j + \delta\,.\left(up_j - low_j\right) \qquad T_{ij} < low_j \quad or \quad T_{ij} < up_j, \qquad \delta \sim U(0,1) \tag{8}$$

IV. Enhancement of Bernstain-Search Differential Evolution (EBSD) Algorithm

In this section, the enhancement of bernstain-search differential evolution (EBSD) is presented to solve engineering design optimization (EDO) problems. The BSD algorithm suffers from a loss of diversity that leads to premature convergence to the local optima. Therefore, the EBSD algorithm is proposed to overcome the premature convergence of the canonical BSD algorithm. In the proposed EBSD algorithm, the trial pattern vector is computed by Eq. (9) enriched by our proposed parameters G, β and P_r. The Chebyshev chaotic map determines the value of G by Eq. (10), the parameter β is computed by Eq. (11), and P_r is the random position in the search space. The flowchart of the proposed EBSD algorithm is shown in Fig.1. In Eq. (11), the parameters t and MaxIt are the current and maximum iteration, respectively.

$$T = \mathrm{G} \times (Best_P - P) + F \times M \times (\beta \times E + (1-\beta) \times Best_P + P_r - P) \tag{9}$$

$$\mathrm{G} = \cos(\mathrm{i} \times cos^{-1}(\mathrm{G}_i)) \tag{10}$$

$$\beta = 2 \times \exp(-(\frac{4 \times t}{MaxIt})^2) \tag{11}$$

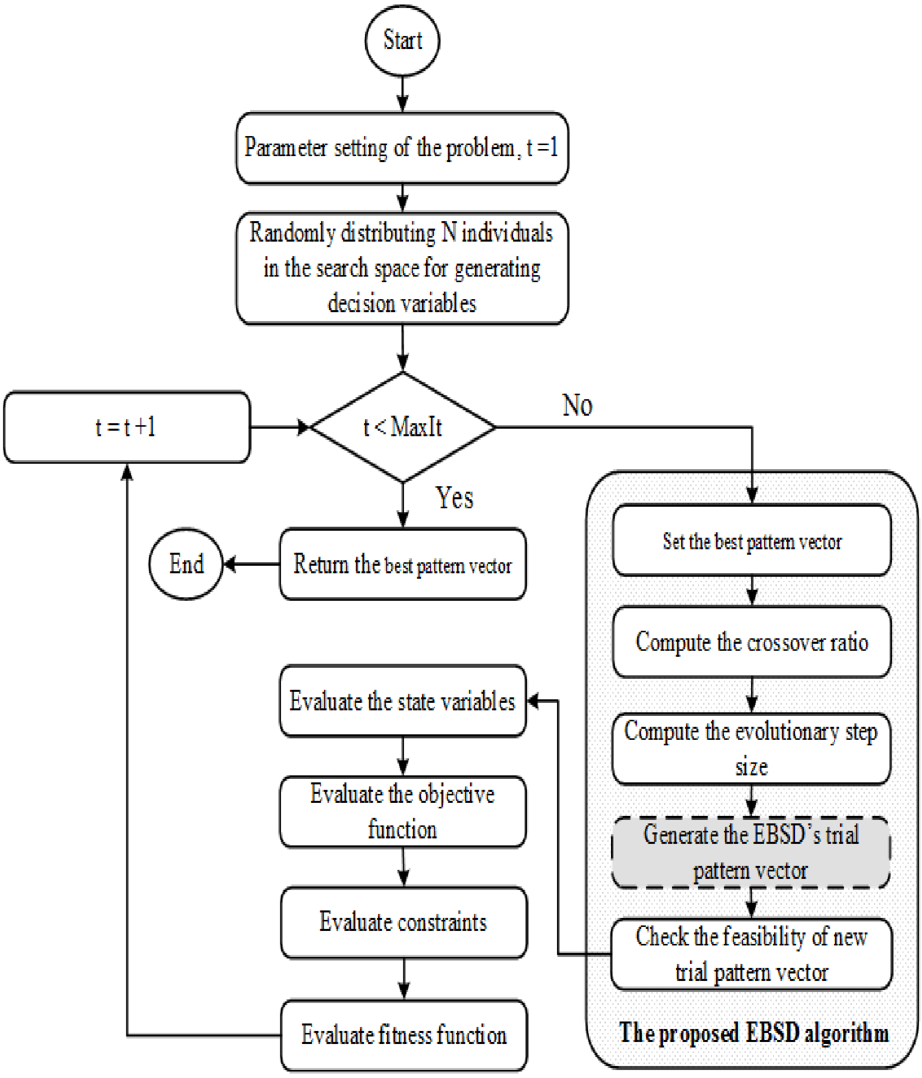

Figure 1. Flowchart of the proposed EBSD algorithm for solving constrained engineering design optimization problems

V. Experimental Evaluation

In this section, the efficiency of the EBSD algorithm is evaluated by four minimization constrained engineering design optimization problems, pressure vessel, welded beam, tension/Compression spring, and three-bar truss problem. Furthermore, the proposed EBSD algorithm was compared with the state-of-the-art swarm intelligence and evolutionary algorithms: comprehensive learning particle swarm optimizer (CLPSO) [66], hybrid differential evolution with biogeography-based optimization (DE/BBO) [67], exploration-enhanced GWO (EEGWO) [21], weighted differential evolution algorithm (WDE) [40], chimp optimization algorithm (ChOA) [20] and bernstain-search differential evolution (BSD) [39]. All experiments are run on a personal computer with the following features: Intel(R) Core(TM) i7-3770 CPU, 3.4 GHz, and 8 GB RAM, Windows 7, the 64-bit operating system using the version R2016b of MATLAB programming. Due to the random nature of the algorithms, the EBSD and contender algorithms independently were run 30 time, and the common parameters such as population size (N) and the maximum number of iterations (MaxIt) were set to 200 and 2000, respectively.

A. Tension/Compression Spring Problem

The tension/compression spring design problem has three nonlinear and linear constraints with three design variables wire diameter (d), mean coil diameter (D), and the number of active coils (N). This problem is designed to minimize the weight of tension/compression spring by handling the constraints defined in Eq. (12). The schematic of this problem shows in Fig.2, and also the obtained results during this competition are reported in Table I. These results indicate that the EBSD algorithm outperforms all other algorithms for finding optimal values of variables d, D, and N. The convergence rate of the EBSD algorithm for solving tension/compression spring design problem is shown in Fig. 3.

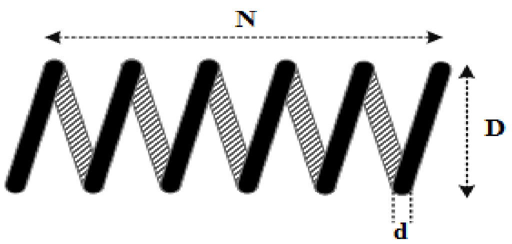

Figure 2. The schematic of the tension/compression spring problem

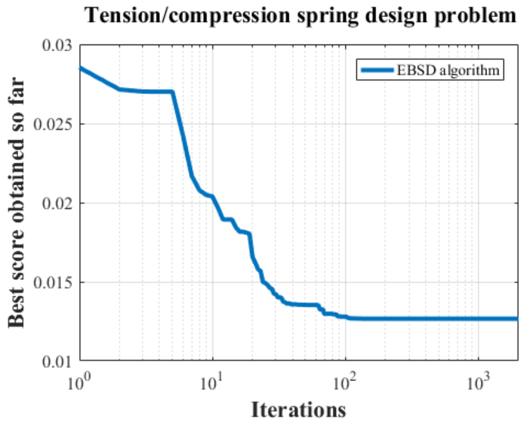

Figure 3. The convergence curves of the EBSD algorithm in tension/compression spring design problem

Consider $x = [x_1 x_2 x_3] = [d, D, N]$,

Minimize $f(x) = (x_3 + 2)x_2 x_1^2$,

Subject to $g_1(x) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0$,

$$g_2(x) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{510 x_1^2} \leq 0, \quad (12)$$

$g_3(x) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0$,

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

Variable range $0.05 \leq x_1 \leq 2.00$, $0.25 \leq x_2 \leq 1.30$, $2.00 \leq x_3 \leq 15.0$

TABLE I. RESULTS FOR TENSION/COMPRESSION SPRING DESIGN PROBLEM

Algorithms	Optimal values for variables			Optimum weight
	d	D	N	
CLPSO	0.0521866216727594	0.368762152224704	10.6314879388538	0.012685839712429
DE/BBO	0.0671321803394291	0.586277295673415	8.16975255260144	0.026870451690880
EEGWO	0.0614938814945025	0.505195423311179	8.17993166957979	0.019447692706730
WDE	0.0636599093934690	0.653637012142558	5.86359900227856	0.020830036326148
ChOA	0.0515584533307910	0.353524600837209	11.5136153275889	0.012699626676001
BSD	0.0517699866230997	0.358660942938636	11.1772435935890	0.012666737362859
EBSD	0.0516454381086322	0.355668856807088	11.3507403444742	**0.012665289472899**

B. Pressure Vessel Design (PVD) Problem

The pressure vessel design problem is well-known in the industrial field, with four parameters and four constraints. The constrained decision variables in the pressure vessel design problem are included the thickness of the shell (T_s or x_1), the thickness of the head (T_h or x_2), inner radius (R or x_3), and length of the cylindrical section of the vessel (L or x_4). In this problem, the objective function should be minimized by satisfying four constraints: three linear and one nonlinear, and this function is formulated in Eq. (13). The schematic of the constrained PVD problem is shown in Fig. 4. The pressure vessel is optimized with the EBSD algorithm, and the obtained results are compared to CLPSO [66], DE/BBO [67], EEGWO [21], WDE [40], ChOA [20], and BSD [39] and reported in Table II. This Table shows that the EBSD algorithm can find a low-cost design. The convergence rate of the EBSD algorithm is illustrated in Fig. 5.

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

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

Subject to $g_1(\vec{x}) = -x_1 + 0.0193 x_3 \leq 0$, (13)

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

	$g_3(\vec{x}) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0,$
	$g_4(\vec{x}) = x_4 - 240 \le 0,$
Variable range	$0 \le x_1 \le 99,$
	$0 \le x_2 \le 99,$
	$10 \le x_3 \le 200,$
	$10 \le x_4 \le 200,$

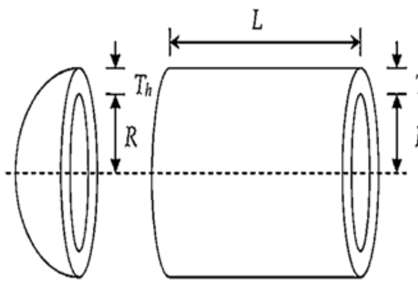

Figure 4. The pressure vessel design problem.

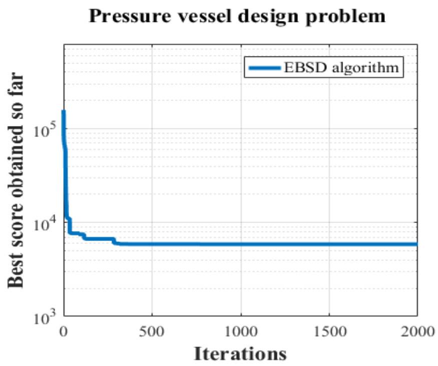

Figure 5. The convergence curves of the EBSD algorithm in pressure vessel design problem.

TABLE II. RESULTS FOR PRESSURE VESSEL DESIGN PROBLEM

Algorithms	Optimal values for variables				Optimum cost
	T_s	T_h	R	L	
CLPSO	0.791745815950072	0.39362390343283	40.8974203635550	192.147531065319	5.933103607984365E+03
DE/BBO	1.30086305203045	1.64796631688201	62.9757926913573	26.0280486087595	1.520217225187381E+04
EEGWO	2.58999204267615	1.00701187435643	97.9097487108382	10	3.198625537262918E+04
WDE	48.4868106513100	88.3483630690864	106.232866002143	19.5114059497321	1.581680402411724E+04
ChOA	1.04375805524499	0.54814029437827	53.2363735879272	77.3302047573049	6.854064418325173E+03
BSD	0.801848403601477	0.39638217623266	41.5320000747373	183.785327667544	5.929041946994261E+03
EBSD	0.780881241079924	0.38599922293360	40.4601679361572	198.052665808410	**5.890012647906080e+03**

C. *Three-bar Truss Problem*

This minimization problem is defined in Eq. (14) with two constrained decision variables x_1 and x_2. The schematic of this problem is shown in Fig. 6. The experimental results are shown in Table III, in which the bold value shows the winning algorithm with the best solutions. These results indicate the proposed EBSD algorithm can optimize the three-bar truss problem with a minimum solution than the contender algorithms. The convergence rate of the EBSD algorithm is plotted in Fig. 7.

Min	$f(\vec{x}) = \left(2\sqrt{2x_1} + x_2\right) \times l,$	
Subject to	$g_1(\vec{x}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \le 0,$	
	$g_2(\vec{x}) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \le 0,$	
	$g_3(\vec{x}) = \frac{1}{\sqrt{2}x_2 + x_1} P - \sigma \le 0,$	(14)
Variable range	$0 \le x_1 \le 1,$	
	$0 \le x_2 \le 1,$	
	$l = 100\ cm, P = 2\frac{kN}{cm^2}, and\ \sigma = 2\ kN/cm^2$	

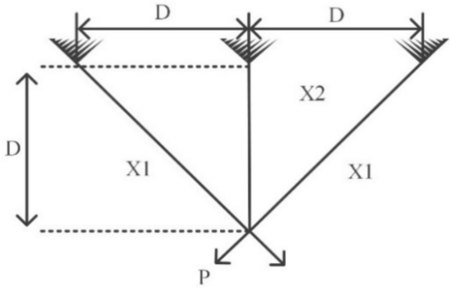

Figure 6. Three bar truss problem

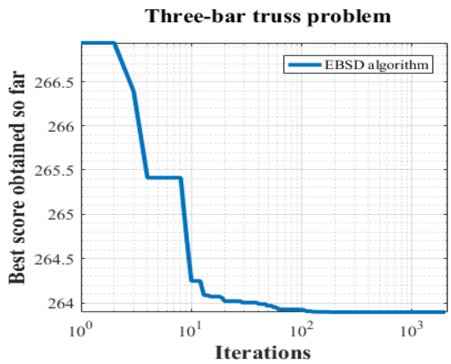

Figure 7. The convergence curves of the EBSD algorithm in three-bar truss problem

TABLE III. RESULTS FOR THE THREE-BAR TRUSS PROBLEM

Algorithms	Optimal values for variables		Optimal weight
	x_1	x_2	
CLPSO	0.788682377347302	0.408227829082155	2.638958457980573E+02
DE/BBO	0.781073415199283	0.431779109746533	2.640988343714328E+02
EEGWO	0.790761722154339	0.402632303723429	2.639244207875771E+02
WDE	0.515535107819326	0.0156341500434795	2.639297829829848E+02
ChOA	0.789283868232206	0.406532014487667	2.638963916520235E+02
BSD	0.788677477421088	0.408241664062184	2.638958433876389E+02
EBSD	0.788675387136042	0.408247576169568	**2.638958433765372E+02**

D. The Welded Beam Design Problem

This well-known problem is formulated by minimizing the overall cost of welded beam fabrication using three constrained decision variables, i.e., the thickness of the weld (h or x_1), length of the clamped bar (l or x_2), the height of the bar (t or x_3), and thickness of the bar (b or x_4). The fabrication cost of a welded beam is considered as an objective and defined in Eq. (15), and also the schematic of this problem is illustrated in Fig. 8. The results are tabulated in Table IV and indicate that the EBSD algorithm is superior to the compared algorithms. The convergence rate of the EBSD algorithm is illustrated in Fig. 9.

$$\begin{aligned}
&\text{Consider} && \vec{x} = [x_1 x_2 x_3 x_4] = [hltb],\\
&\text{Minimize} && (f\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2),\\
&\text{Subject to} && g_1(\vec{x}) = \tau(\vec{x}) - \tau_{max} \le 0,\\
& && g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{max} \le 0,\\
& && g_3(\vec{x}) = \delta(\vec{x}) - \delta_{max} \le 0,\\
& && g_4(\vec{x}) = x_1 - x_4 \le 0,\\
& && g_5(\vec{x}) = P - P_c(\vec{x}) \le 0,\\
& && g_6(\vec{x}) = 0.125 - x_1 \le 0,\\
& && g_7(\vec{x}) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \le 0,\\
&\text{Variable range} && 0.1 \le x_1 \le 2,\\
& && 0.1 \le x_2 \le 10,\\
& && 0.1 \le x_3 \le 10,\\
& && 0.1 \le x_4 \le 2,
\end{aligned} \tag{15}$$

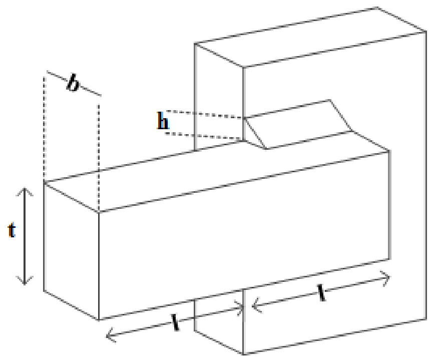

Figure 8. The welded beam problem

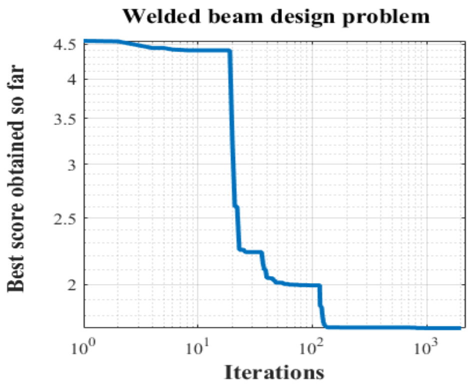

Figure 9. The convergence curves of the EBSD algorithm in the welded beam problem

TABLE IV. RESULTS OF THE WELDED BEAM DESIGN PROBLEM

Algorithms	Optimal values for variables				Optimum cost
	h	l	t	b	
CLPSO	0.205648373679504	3.47313938334828	9.03790267620040	0.205723791915902	1.725261786070104
DE/BBO	0.313333731813621	2.79799463025649	7.78407926889794	0.362471191915633	2.583668745036021
EEGWO	0.189579634875795	8.06858666751934	8.39710631128266	0.288452725805553	2.892016948788617
WDE	0.243496501605723	3.46444980272299	7.50783436531566	0.305771508536195	2.155783092880342
ChOA	0.201035484557912	3.59691466012680	9.12453451381718	0.205814056229098	1.750447414674333
BSD	0.205346187751607	3.46980771492620	9.05989449594751	0.205613967305163	1.727298276704067
EBSD	0.205796274208058	3.46947443466481	9.03553427600789	0.205796464889623	**1.725138336562238**

VI. CONCLUSION

Finding an effective solution to solve a constrained engineering optimization problem is a challenging task, in which a few of the current metaheuristic algorithms can cope with this challenge. This paper proposes an enhancement of the BSD algorithm named EBSD for handling constraints in engineering optimization problems. In the EBSD, the trial pattern vector is improved, and the Chebyshev chaotic map is used to increase the diversity of the population. The experimental evaluation was designed to assess the EBSD and compare its results in solving constrained engineering optimization problems with the state-of-the-art algorithms. The experiments are conducted by four constrained engineering design optimization problems, including pressure vessel, welded beam, tension/compression spring, and three-bar truss with various constraints. The experimental results proved that the EBSD algorithm is superior to competitors in this paper to solve these constrained problems.

REFERENCES

[1] S. Al-Muhaideb and M. E. B. Menai, "Hybrid metaheuristics for medical data classification," in Hybrid Metaheuristics, ed: Springer, 2013, pp. 187-217.
[2] N. Zeng, H. Qiu, Z. Wang, W. Liu, H. Zhang, and Y. Li, "A new switching-delayed-PSO-based optimized SVM algorithm for diagnosis of Alzheimer's disease," Neurocomputing, vol. 320, pp. 195-202, 2018.
[3] S. Taghian and M. H. Nadimi-Shahraki, "Binary Sine Cosine Algorithms for Feature Selection from Medical Data," arXiv preprint arXiv:1911.07805, 2019.
[4] Z. Zhang, S. Ding, and W. Jia, "A hybrid optimization algorithm based on cuckoo search and differential evolution for solving constrained engineering problems," Engineering Applications of Artificial Intelligence, vol. 85, pp. 254-268, 2019.
[5] N. B. Guedria, "Improved accelerated PSO algorithm for mechanical engineering optimization problems," Applied Soft Computing, vol. 40, pp. 455-467, 2016.
[6] M. H. Nadimi-Shahraki, S. Taghian, and S. Mirjalili, "An improved grey wolf optimizer for solving engineering problems," Expert Systems with Applications, vol. 166, p. 113917.
[7] E. Mezura-Montes and C. A. C. Coello, "A simple multimembered evolution strategy to solve constrained optimization problems," IEEE Transactions on Evolutionary computation, vol. 9, pp. 1-17, 2005.
[8] M. B. Dezfouli, M. H. Nadimi-Shahraki, and H. Zamani, "A Novel Tour Planning Model using Big Data," in 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), 2018, pp. 1-6, https://doi.org/10.1109/IDAP.2018.8620933.
[9] I. BoussaïD, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," Information sciences, vol. 237, pp. 82-117, 2013.
[10] E.-G. Talbi, Metaheuristics: from design to implementation vol. 74: John Wiley & Sons, 2009.
[11] P. J. Van Laarhoven and E. H. Aarts, "Simulated annealing," in Simulated annealing: Theory and applications, ed: Springer, 1987, pp. 7-15.

[12] F. Glover and M. Laguna, "Tabu search," in Handbook of combinatorial optimization, ed: Springer, 1998, pp. 2093-2229.
[13] H. R. Lourenço, O. C. Martin, and T. Stützle, "Iterated local search," in Handbook of metaheuristics, ed: Springer, 2003, pp. 320-353.
[14] C. Voudouris, E. P. Tsang, and A. Alsheddy, "Guided local search," in Handbook of metaheuristics, ed: Springer, 2010, pp. 321-361.
[15] O. Bozorg-Haddad, M. Solgi, and H. A. Loï, Meta-heuristic and evolutionary algorithms for engineering optimization: John Wiley & Sons, 2017.
[16] H. Zamani, M. H. Nadimi-Shahraki, and A. H. Gandomi, "CCSA: Conscious Neighborhood-based Crow Search Algorithm for Solving Global Optimization Problems," Applied Soft Computing, vol. 85, p. 105583, 2019.
[17] A. Askarzadeh, "Bird mating optimizer: an optimization algorithm inspired by bird mating strategies," Communications in Nonlinear Science and Numerical Simulation, vol. 19, pp. 1213-1228, 2014.
[18] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm," Computers & Structures, vol. 169, pp. 1-12, 2016.
[19] S. Mirjalili and A. Lewis, "The whale optimization algorithm," Advances in engineering software, vol. 95, pp. 51-67, 2016.
[20] M. Khishe and M. R. Mosavi, "Chimp optimization algorithm," Expert Systems with Applications, p. 113338, 2020.
[21] W. Long, J. Jiao, X. Liang, and M. Tang, "An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization," Engineering Applications of Artificial Intelligence, vol. 68, pp. 63-80, 2018.
[22] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," Advances in engineering software, vol. 69, pp. 46-61, 2014.
[23] M. Dorigo and G. Di Caro, "Ant colony optimization: a new meta-heuristic," in Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406), 1999, pp. 1470-1477.
[24] S. Mirjalili, "The ant lion optimizer," Advances in engineering software, vol. 83, pp. 80-98, 2015.
[25] E. S. Fard, K. Monfaredi, and M. H. Nadimi-Shahraki, "Application methods of ant colony algorithm," Am. J. Softw. Eng. Appl, vol. 3, pp. 12-20, 2014, https://doi.org/10.11648/j.ajsea.20140302.11.
[26] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," Neural Computing and Applications, vol. 27, pp. 1053-1073, 2016.
[27] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proceedings of ICNN'95-International Conference on Neural Networks, 1995, pp. 1942-1948.
[28] E. S. Fard, K. Monfaredi, and M. H. Nadimi-Shahraki, "An area-optimized chip of ant colony algorithm design in hardware platform using the address-based method," International Journal of Electrical and Computer Engineering, vol. 4, p. 989, 2014.
[29] R. Agrawal, B. Kaur, and S. Sharma, "Quantum based whale optimization algorithm for wrapper feature selection," Applied Soft Computing, vol. 89, p. 106092, 2020.
[30] H. G. Arjenaki, M. H. Nadimi-Shahraki, and N. Nourafza, "A low cost model for diagnosing coronary artery disease based on effective features," International Journal of Electronics Communication and Computer Engineering, vol. 6, pp. 93-97, 2015.
[31] H. Zamani and M. H. Nadimi-Shahraki, "Swarm Intelligence approach for breast cancer diagnosis," International Journal of Computer Applications, vol. 151, 2016.
[32] A. Shalikar, M. R. Ashouri, and M. H. Nadimi-Shahraki, "A CAD system for automatic classification of brain strokes in CT images," International Journal of Mechatronics, Electrical and Computer Technology, vol. 4, pp. 67-85, 2014.
[33] S. Taghian, M. H. Nadimi-Shahraki, and H. Zamani, "Comparative analysis of transfer function-based binary Metaheuristic algorithms for feature selection," in 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), 2018, pp. 1-6, https://doi.org/10.1109/IDAP.2018.8620828.
[34] S. Taghian and M. H. Nadimi-Shahraki, "A Binary Metaheuristic Algorithm for Wrapper Feature Selection," International Journal of Computer Science Engineering (IJCSE), vol. 8, pp. 168-172, 2019.
[35] H. Zamani and M.-H. Nadimi-Shahraki, "Feature selection based on whale optimization algorithm for diseases diagnosis," International Journal of Computer Science and Information Security, vol. 14, p. 1243, 2016.
[36] J. H. Holland, "Genetic algorithms," Scientific american, vol. 267, pp. 66-73, 1992.
[37] I. Rechenberg, "Evolution strategy: Optimization of technical systems by means of biological evolution," Fromman-Holzboog, Stuttgart, vol. 104, pp. 15-16, 1973.
[38] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," Journal of global optimization, vol. 11, pp. 341-359, 1997.
[39] P. Civicioglu and E. Besdok, "Bernstain-search differential evolution algorithm for numerical function optimization," Expert Systems with Applications, vol. 138, p. 112831, 2019.
[40] P. Civicioglu, E. Besdok, M. A. Gunen, and U. H. Atasever, "Weighted differential evolution algorithm for numerical function optimization: a comparative study with cuckoo search, artificial bee colony, adaptive differential evolution, and backtracking search optimization algorithms," Neural Computing and Applications, vol. 32, pp. 3923-3937, 2020.
[41] M. H. Nadimi-Shahraki, S. Taghian, S. Mirjalili, and H. Faris, "MTDE: An effective multi-trial vector-based differential evolution algorithm and its applications for engineering design problems," Applied Soft Computing, vol. 97, p. 106761, 2020.
[42] M. Yang, C. Li, Z. Cai, and J. Guan, "Differential evolution with auto-enhanced population diversity," IEEE transactions on cybernetics, vol. 45, pp. 302-315, 2014.
[43] W.-l. Xiang, Y.-z. Li, X.-l. Meng, C.-m. Zhang, and M.-q. An, "A grey artificial bee colony algorithm," Applied Soft Computing, vol. 60, pp. 1-17, 2017.
[44] H. Wang, H. Sun, C. Li, S. Rahnamayan, and J.-S. Pan, "Diversity enhanced particle swarm optimization with neighborhood search," Information Sciences, vol. 223, pp. 119-135, 2013.
[45] Q. He and L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems," Engineering applications of artificial intelligence, vol. 20, pp. 89-99, 2007.
[46] H. K. Zahrani, M. H. Nadimi-Shahraki, and H. R. Sayarshad, "An intelligent social-based method for rail-car fleet sizing problem."
[47] Y. Yongbin, S. A. Bagherzadeh, H. Azimy, M. Akbari, and A. Karimipour, "Comparison of the artificial neural network model prediction and the experimental results for cutting region temperature and surface roughness in laser cutting of AL6061T6 alloy," Infrared Physics & Technology, vol. 108, p. 103364, 2020.
[48] X. Mei, Z. Li, S. A. Bagherzadeh, A. Karimipour, M. Bahrami, and A. Karimipour, "Development of the ANN–KIM composed model to predict the nanofluid energetic thermal conductivity via various types of nano-powders dispersed in oil," Journal of Thermal Analysis and Calorimetry, pp. 1-6, 2020.
[49] H. Arasteh, R. Mashayekhi, D. Toghraie, A. Karimipour, M. Bahiraei, and A. Rahbari, "Optimal arrangements of a heat sink partially filled with multilayered porous media employing hybrid nanofluid," Journal of Thermal Analysis and Calorimetry, vol. 137, pp. 1045-1058, 2019.
[50] H. R. Sayarshad and H. O. Gao, "Optimizing dynamic switching between fixed and flexible transit services with an idle-vehicle relocation strategy and reductions in emissions," Transportation Research Part A: Policy and Practice, vol. 135, pp. 198-214, 2020.

[51] H. Ershadi and A. Karimipour, "Present a multi-criteria modeling and optimization (energy, economic and environmental) approach of industrial combined cooling heating and power (CCHP) generation systems using the genetic algorithm, case study: a tile factory," Energy, vol. 149, pp. 286-295, 2018.
[52] J. Zhang, M. Xiao, L. Gao, and Q. Pan, "Queuing search algorithm: A novel metaheuristic algorithm for solving engineering optimization problems," Applied Mathematical Modelling, vol. 63, pp. 464-490, 2018.
[53] 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, pp. 2592-2612, 2013.
[54] S. Gupta, K. Deep, H. Moayedi, L. K. Foong, and A. Assad, "Sine cosine grey wolf optimizer to solve engineering design problems," Engineering with Computers, pp. 1-27, 2020.
[55] T.-S. Du, X.-T. Ke, J.-G. Liao, and Y.-J. Shen, "DSLC-FOA: improved fruit fly optimization algorithm for application to structural engineering design optimization problems," Applied Mathematical Modelling, vol. 55, pp. 314-339, 2018.
[56] C.-l. Sun, J.-c. Zeng, and J.-s. Pan, "An improved vector particle swarm optimization for constrained optimization problems," Information Sciences, vol. 181, pp. 1153-1163, 2011.
[57] H. R. Sayarshad, "Using bees algorithm for material handling equipment planning in manufacturing systems," The International Journal of Advanced Manufacturing Technology, vol. 48, pp. 1009-1018, 2010.
[58] T. Ray and K. Liew, "A swarm with an effective information sharing mechanism for unconstrained and constrained single objective optimisation problems," in Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546), 2001, pp. 75-80.
[59] L. C. Cagnina, S. C. Esquivel, and C. A. C. Coello, "Solving engineering optimization problems with the simple constrained particle swarm optimizer," Informatica, vol. 32, 2008.
[60] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," IEEE Transactions on evolutionary computation, vol. 4, pp. 284-294, 2000.
[61] S. B. Hamida and M. Schoenauer, "ASCHEA: new results using adaptive segregational constraint handling," in Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600), 2002, pp. 884-889.
[62] W. He, S. A. Bagherzadeh, H. Shahrajabian, A. Karimipour, H. Jadidi, and Q.-V. Bach, "Controlled elitist multi-objective genetic algorithm joined with neural network to study the effects of nano-clay percentage on cell size and polymer foams density of PVC/clay nanocomposites," Journal of Thermal Analysis and Calorimetry, vol. 139, pp. 2801-2810, 2020.
[63] S. A. Bagherzadeh, M. T. Sulgani, V. Nikkhah, M. Bahrami, A. Karimipour, and Y. Jiang, "Minimize pressure drop and maximize heat transfer coefficient by the new proposed multi-objective optimization/statistical model composed of “ANN+ Genetic Algorithm” based on empirical data of CuO/paraffin nanofluid in a pipe," Physica A: Statistical Mechanics and its Applications, vol. 527, p. 121056, 2019.
[64] H. Wu, S. A. Bagherzadeh, A. D’Orazio, N. Habibollahi, A. Karimipour, M. Goodarzi, et al., "Present a new multi objective optimization statistical Pareto frontier method composed of artificial neural network and multi objective genetic algorithm to improve the pipe flow hydrodynamic and thermal properties such as pressure drop and heat transfer coefficient for non-Newtonian binary fluids," Physica A: Statistical Mechanics and its Applications, vol. 535, p. 122409, 2019.
[65] H. Garg, "A hybrid PSO-GA algorithm for constrained optimization problems," Applied Mathematics and Computation, vol. 274, pp. 292-305, 2016.
[66] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," IEEE transactions on evolutionary computation, vol. 10, pp. 281-295, 2006.
[67] W. Gong, Z. Cai, and C. X. Ling, "DE/BBO: a hybrid differential evolution with biogeography-based optimization for global numerical optimization," Soft Computing, vol. 15, pp. 645-665, 2010.