Journal of Bionic Engineering (2023) 20:1852–1877
https://doi.org/10.1007/s42235-023-00340-2

RESEARCH ARTICLE

Chaotic Social Group Optimization for Structural Engineering Design Problems

Anima Naik[1]

Received: 14 September 2022 / Revised: 7 January 2023 / Accepted: 10 January 2023 / Published online: 20 February 2023

Abstract

Till now, several novel metaheuristic algorithms are proposed for global search. But only specific algorithms have become popular or attracted researchers, who are efficient in solving global optimization problems as well as real-world application problems. The Social Group Optimization (SGO) algorithm is a new metaheuristic bioinspired algorithm inspired by human social behavior that attracted researchers due to its simplicity and problem-solving capability. In this study, to deal with the problems of low accuracy and local convergence in SGO, the chaos theory is introduced into the evolutionary process of SGO. Since chaotic mapping has certainty, ergodicity, and stochastic property, by replacing the constant value of the self-introspection parameter with chaotic maps, the proposed chaotic social group optimization algorithm increases its convergence rate and resulting precision. The proposal chaotic SGO is validated through 13 benchmark functions and after that 9 structural engineering design problems have been solved. The simulated results have been noticed as competent with that of state-of-art algorithms regarding convergence quality and accuracy, which certifies that improved SGO with chaos is valid and feasible.

Keywords Chaos · Bionic algorithm · Constrained optimization · SGO · Design problem

1 Introduction

For simplicity and gradient-free mechanism, metaheuristic optimization algorithms are becoming popular among researchers globally. According to the no-free lunch (NFL) theorem [1], a single metaheuristic optimization algorithm cannot solve all optimization problems. It may solve some problems with high performance and some problems with low performance. Hence, researchers have invented many optimization algorithms, and every year new algorithms are being proposed. At the same time, the existing algorithms are also improved.

Till now, several novel metaheuristic algorithms are proposed for global search. These algorithms reveal improved performances in comparison to traditional optimization techniques, especially when applied to solve non-convex optimization problems [2]. Satapathy et al. have developed a promising metaheuristic algorithm, called social group optimization (SGO) in year 2016, which is inspired by humans social behavior to solve complex problems [3]. Preliminary studies suggest that the SGO demonstrates superior results when compared with other metaheuristics algorithms [4, 5].

The metaheuristic algorithms consist of two essential steps exploration and exploitation. Exploration refers to searching the whole search space of the algorithm. This factor shows the capability of a method in global search. Exploitation is the capability to find local optimum around different feasible solutions It has been seen that if an optimization algorithm has good exploration capability, then it will be lacking in good exploitation capability and vice versa [6]. Previously, researchers were using random walks and gradient descent methods for improving exploration and exploitation, respectively. But, increasing the overall computational cost of the algorithm, researchers are using chaotic maps to improve diversification and local exploitation of search space to find the optimal solutions [7, 8]. The interesting property of the systems is that when there is a minor change in the system, the whole system gets affected [9].

In the past, various metaheuristic optimization algorithms have been used together with chaotic sequences

✉ Anima Naik
animanaik@vignaniit.edu.in

1 Department of CSE, VIIT, Visakhapatnam, Andhra Pradesh 530049, India

such as Artificial Bee Colony (ABC) optimization [10], Harmony Search (HS) [11], Particle Swarm Optimization (PSO) [12], genetic algorithm (GA) [13], differential evolution (DE) [14], simulated annealing (SA) [15], firefly algorithm (FA) [16], krill herd (KH) [17], imperialist competitive algorithm (ICA)[18], biogeography-based optimization (BBO) [19], bat algorithm (BA) [20], gravitational search algorithm (GSA) [8], Bird swarm algorithms (BSA) [21], league championship algorithms (LCA) [22], and farmland fertility (FF) [23].

Based on the SGO algorithm, in this paper, a family of the chaotic algorithm is proposed, called Chaotic SGO (CSGO). The insertion of chaotic maps in the structure of the CSGO algorithm is motivated by the following arguments: (1) the SGO functions with various dimensions and characteristics (unimodal, multimodal, composite) [5]. Thus, it is expected that in the case of structural engineering problems this efficiency will be maintained. (2) Consulting several databases, we found that the SGO algorithm equipped with various chaotic maps has not been used in solving the structural engineering problem (3). In addition, CSGO algorithms are easy to implement and have the ability to maintain a good balance between exploration and exploitation, thus being able to generate promising solutions during the iterative process. Again to evaluate the proposed family of CSGO algorithms, 13 benchmark functions are utilized and their performances are compared: with 10 metaheuristics optimization algorithms.

Normally, metaheuristic algorithms show good results on benchmark functions, but they perform poorly on real-world problems. The practical problem is an actual test for checking the problem-solving capabilities of an optimization algorithm. Therefore, to further evaluate the validity of the proposed family of CSGO algorithms in real-world applications, these are used to solve nine structural engineering design problems. The results reveal that there is an improvement in the performance of the proposed algorithms due to the application of deterministic chaotic signals.

The rest of the paper is organized as follows. Section 2 presents the description of SGO. Section 3 outlines the chaotic maps that generate chaotic sequences in the SGO. Section 4 presents the proposed family of CSGO algorithms. Simulations and result analysis are presented in Sect. 5. Finally, the conclusions and directions for further research are drawn in Sect. 6.

2 Social Group Optimization (SGO) Algorithm

The SGO algorithm is based on human behavior towards society in solving complex problems. The person is a candidate solution and the person's knowledge is the fitness value of the problem. The human traits are designated as the design variable of the problem which corresponds to the dimension of the problem. The SGO algorithm goes through two phases, namely the improving and acquiring phases. In the group, each individual's knowledge level is improved based on the best individual influence in the improving phase. The best candidate solution is the one having the highest knowledge level and the ability to solve the problem under concern. The mutual interaction between individuals in the group and at the same time interaction with the best person through the acquiring phase improved each person's knowledge. For a detailed description of the SGO algorithm, please refer to the paper [3, 24]. The SGO algorithm, in short, is given as follows:

Let P_i, $i = 1,2,3,\ldots, N$, be the N persons of the social group and each person P_i is defined by $P_i = (p_{i1}, p_{i2}, p_{i3}, \ldots \ldots, p_{iD})$ where D is the number of traits assigned to a person and f_i's are their corresponding fitness value, respectively. For every iteration, each person has to undergo the "improving" and "acquiring" phase in the hope of finding a better solution.

2.1 Improving Phase

$$\text{Find best}_P = P_i \text{ such that } f_i \text{ isminmum in social group} \quad (1)$$

$$P\text{new}_i = c * P_i + \text{rand} * \left(\text{best}_P - P_i\right)$$

Accept Pnew if it gives better fitness than P

where rand is a random number, rand $\sim U(0, 1)$, and c is known as a self-introspection parameter in (0,1).

2.2 Acquiring Phase

$$\text{best}_P = P_i \text{ such that} f_i \text{ is minmum in social group} \quad (2)$$

Randomly select one person P_r, where $i \neq r$

If f $(P_i) <$ f (P_r)

$$P\text{new}_i = P_i + \text{rand}_1 * \left(P_i - P_r\right) + \text{rand}_2 * (\text{best}_P - P_i)$$

Else

$$P\text{new}_i = P_i + \text{rand}_1 * \left(P_r - P_i\right) + \text{rand}_2 * (\text{best}_P - P_i)$$

End If

Accept Pnew if it gives better fitness than P where rand_1 and rand_2 are two independent random numbers, $\text{rand}_1 \sim U(0, 1)$, and $\text{rand}_2 \sim U(0, 1)$. These random numbers are used to affect the stochastic nature of the algorithm.

3 Chaotic Map

Variety of chaotic maps are available in the optimization field [25]. In this study, 10 most widely used uni-dimensional chaotic maps have been employed [26]. The mathematical forms of chaotic maps employed are represented as follows:

Chebyshev map

$$x_{k+1} = \cos(k cos^{-1}(x_k)) \tag{3}$$

Circle map

$$x_{k+1} = x_k + b - (P/2\pi)\sin(2\pi x_k) mod(1) \tag{4}$$

where $P = 0.5$ and b = 0.2.

Gauss map

$$x_{k+1} = \begin{cases} 0 & x_k = 0 \\ \frac{1}{x_k mod(1)} & otherwise \end{cases} \tag{5}$$

$$\frac{1}{x_k mod(1)} = \frac{1}{x_k} - \left[\frac{1}{x_k}\right] \tag{6}$$

Iterative map

$$x_{k+1} = \sin\left(\frac{P\pi}{x_k}\right) \tag{7}$$

where $P \in (0, 1)$ is a suitable parameter.

Logistic map

$$x_{k+1} = P x_k (1 - x_k) \tag{8}$$

$P = 4$ is used for the experiments.

Piecewise map

$$x_{k+1} = \begin{cases} \frac{x_k}{P} & 0 \le x_k \le P \\ \frac{x_k - P}{0.5 - P} & 0 \le x_k \le 0.5 \\ \frac{1 - P - x_k}{0.5 - P} & 0.5 \le x_k \le 1 - P \\ \frac{1 - x_k}{P} & 1 - P \le x_k \le 1 \end{cases} \tag{9}$$

where $0 \le P \le 0.5$.

.

Sine map

$$x_{k+1} = \sin(\pi x_k) \tag{10}$$

Singer map

$$x_{k+1} = P\left(7.86x_k - 23.31x_k^2 + 28.75x_k^3 - 13.302875x_k^4\right) \tag{11}$$

where $P \in (0.9, 1.08)$.

Sinusoidal map

$$x_{k+1} = P x_k^2 \sin(\pi x_k) \tag{12}$$

where $P = 2.3$.

Tent map

$$x_{k+1} = \begin{cases} \frac{x_k}{0.7} & x_k < 0.7 \\ \frac{10}{3}(1 - x_k) & x_k \ge 0.7 \end{cases} \tag{13}$$

4 The Proposed Chaotic Social Group Optimization Algorithms

On consulting several databases (Scopus, Springer, Elsevier), it can be seen much latest research work has been done through the SGO algorithm. The SGO algorithm has been successfully applied to many research areas such as in the medical field [27, 28], civil engineering [29], optimization engineering [30], communication engineering [31], operation management [32, 33], and many more.

In metaheuristic algorithms, randomness is achieved through some probability distributions. Such randomness can be replaced with a chaotic map due to similar properties of randomness with better statistical and dynamic properties. Such dynamical mixing helps the algorithm to diverse enough to reach every mode in the multimodal objective landscape. Due to the ergodicity and mixing properties of chaos, algorithms perform the iterative search at higher speeds than standard stochastic searches with standard probability distributions.

While going through the literature on the SGO algorithm, we have found that there is only a single paper on SGO which is combined with chaotic concepts [34]. In this paper, the self-inspection parameter 'C' value is replaced by two chaotic strategies as Chaotic decreasing inertia weight and Chaotic random inertia weight where logistic maps are aggregated with two popular techniques i.e., liner decreasing inertia weight and random inertia weight. Here, authors proved that the chaotic maps do not significantly affect the convergence of SGO. Again the authors replaced the 'C' value with another adaptive chaotic inertia weight to adjust the weight with logistic maps to introduce chaotic sequence into iterations and proved that SGO with adaptive chaotic inertia weight performs better for some benchmark functions.

In the SGO algorithm, the self-introspection parameter $C = 0.2$ is made constant for all the persons in all generations, and this parameter is responsible for a person to improve his/her knowledge level from the current position towards the optimum position. In order to increase the searchability of the algorithm, the parameter C should be changed or redefined in a manner such that the improvement of the person should be done at a higher speed than the standard speed. This can be achieved by selecting different C values according to the chaotic function, as the insertion of chaotic maps in the structure of metaheuristic algorithms can increase the efficiency of the new algorithm [20, 35]. Equipping the SGO algorithm with chaotic maps aims to improve the capacity of the CSGO algorithm to avoid local minimums, increase stability and strengthen the global search. Hence if replacement happens the potential benefits of C are retained by chaotic numbers.

When the 'C' value in SGO is replaced by chaotic maps, then the CSGO can be an algorithm-specific parameter-free algorithm. Then, when we compared CSGO with other algorithms that have different parameter settings, we can declare for the CSGO algorithm no need to bother with parameter settings. The selected chaotic maps that produce chaotic numbers in (0, 1) have been listed in Sect. 3. The family of CSGO algorithms maybe simply classified and described as follows:

In Chaotic SGO1 (CSGO1), Chaotic SGO2 (CSGO2), Chaotic SGO3 (CSGO3), Chaotic SGO4 (CSGO4), Chaotic SGO5 (CSGO5), Chaotic SGO6 (CSGO6), Chaotic SGO7 (CSGO7), Chaotic SGO8 (CSGO8), Chaotic SGO9 (CSGO9), and Chaotic SGO10 (CSGO10) algorithm, the self-introspection parameter C is replaced by a chaotic number generated by the Chebyshev map, circle map, gauss map, iterative map, logistic map, piecewise map, sine map, singer map, sinusoidal map, and tent map, respectively.

Now, it can be said that both original SGO and chaotic SGOs algorithms have the same structure; the only difference between them is the self-introspection parameter that is replaced by chaotic maps in chaotic SGOs and all other conditions remain the same. If we carefully see, four random numbers have been used in SGO: the random numbers in the initialization phase, the improving phase, and two in acquiring phase, and these are not replaced by any chaotic maps in CSGOs. It can be seen from the literature that the chaotic maps replace the random numbers of the chaotic-based stochastic algorithm and even in population initialization. From paper [36], the author has experimentally proven that logistic map-based initialization is able to generate more uniformly distributed particles in the allowable search space to enhance the stability of the algorithm. Not replacing random numbers of the CSGO algorithm with any of the chaotic maps create uniqueness in the algorithm. Although there is no mathematical proof for enhancing the stability of the SGO algorithm still, it has been proved through our experiments that the proposed CSGO algorithms increase their convergence rate and the resulting precision than the SGO algorithm.

5 Simulation, Experimental Results, and Discussion

Every novel optimization algorithm must be subjected to well-defined benchmark functions to measure and test the performance. There are many benchmark functions available; however, there is no standardized set of benchmark functions that are agreed upon for validating new algorithms. To validate and benchmark the performance of the proposed CSGO family of algorithms, simulations on 13 benchmark functions are conducted. One of the main reasons for selecting these functions is that they are utilized in many papers [37–41]. Out of 13, 7 are unimodal benchmark functions and 6 are multimodal benchmarks. Detailed descriptions of these benchmark functions are given in papers [37–41]. After that, nine structural engineering design problems are considered, and the detailed descriptions of these design problems are given in their respective cited papers. All algorithms are implemented using MATLAB 2016a, under Microsoft Windows 10 operating system. Simulations are carried out on an Intel Core i5, 8 GB memory laptop.

5.1 Algorithm Validation

For validating the performance of the CSGO family of algorithms, 13 benchmarks are employed as described above and the results are compared with 10 different metaheuristics algorithms such as GSA [42], Whale Optimization Algorithm (WOA) [37], Henry Gas Solubility Optimization (HGSO) [43], Seagull Optimization Algorithm (SOA) [44], Marine Predators Algorithm (MPA) [45], Tunicate Swarm Algorithm (TSA) [46], Slime Mould Algorithm (SMA) [47], Sooty Term Optimization Algorithm (STOA) [48], Harris Hawks Optimization (HHO) [38], and Ground-Tour Algorithm (GTA) [49]. In experiment 1, the CSGO family of algorithms is compared with each other, and Table 2 illustrates the comparative results. Similar to experiment 2, the performance of the CSGO family is compared with the other ten algorithms and Table 3 illustrates the comparative results. In the experiments, the parameters max_FEs have been kept fixed at 10,000. Hence, the number of iterations and population size may vary for different algorithms. The algorithmic parameter settings are based on the parameters widely used by various researchers and these are mentioned in Table 1.

5.1.1 Experiment 1: The Performance Comparison of the CSGO Family of Algorithms

In this experiment, the performance of the proposed CSGO family of algorithms such as CSGO1, CSGO2, CSGO3, CSGO4, CSGO5, CSGO6, CSGO7, CSGO8, CSGO9, and CSGO10 are compared with each other. Statistical results of 30 repetitions in terms of the best (BEST), worst (WORST), average (MEAN), and standard deviation (SD) of fitness solutions are determined and reported in Table 2 to ensure stability and statistical significance with the best results are highlighted in bold. In the tables, the symbol '||' represents that its value is equal to the value of the above column.

It is seen from Table 2 that the CSGO3 algorithm reaches the global optimum for all the functions except F5–F7, F10, F12, and F13, and in 10 cases out of 13 finds the best solutions than others. CSGO4 in 6 cases, CSGO7 in 5 cases, and other algorithms except for CSGO9 in 4 cases out of 13 find the best solutions, whereas CSGO9 in 3 cases finds the best solutions. For the F10 function, all algorithms find an equivalent solution but not an optimal solution. Hence, it can be said that the CSGO3 algorithm outperformed all other CSGO family algorithms.

5.1.2 Experiment 2: The Performance Comparison with Other Metaheuristics Algorithms

From experiment 1, it can be examined that CSGO3 has shown superior performance in comparison to all algorithms of the CSGO family in terms of fitness function evaluation. Therefore, in this experiment, CSGO3 is compared with the other ten algorithms for performance validation. Statistical results of 30 repetitions in terms of the best (BEST), worst (WORST), average (MEAN), and standard deviation (SD) of fitness solutions are determined and reported in Table 3 to ensure stability and statistical significance with the best

Table 1 Parameter setting of algorithms

Sl. no.	Algorithms	Parameters	Values
1	Gravitational search algorithm	Gravitational constant	100
		Alpha constant	20
		Rnorm	2
		Rpower	1
2	Whale Optimization Algorithm	Control parameter (α_1)	[2, 0]
		Control parameter (α_2)	[− 2, − 1]
		b	1
3	Henry Gas Solubility Optimization	Cluster number	5
		M1	0.1
		M2	0.2
		Beta	1
		Alpha	1
		K	1
		L1	5e−03
		L2	100
		L3	1e−02
4	Harris Hawks Optimization	beta	1.5
5	Seagull Optimization Algorithm	Control parameter (A)	[2,0]
		f_c	2
6	Sooty Term Optimization Algorithm	f_c	2
7	G Ground-Tour Algorithm	G	9.81
		$m_{\min}$	45
		$m_{\max}$	95
		crr	0.002
		rho_ar	1.225
		Cd_A	0.307
		$C_{\min}$	0.5
		$C_{\max}$	1
8	Marine Predators Algorithm	FADs	0.2
		P	0.5
9	Slime Mould Algorithm	Parameter	0.03
10	Tunicate Swarm Algorithm	$P_{\min}$	1
		$P_{\max}$	4

Table 2 Results of CSGO family of algorithms

Algo/functions		F1	F2	F3	F4	F5	F6	F7
CSGO1	BEST	1.0683e−92	3.2308e−47	5.6021e−91	3.9926e−46	26.0415	2.6931e−05	3.4617e−05
	WORST	1.3538e−91	1.6891e−46	3.6616e−90	6.9955e−46	27.6446	0.2666	4.9832e−04
	MEAN	5.7815e−92	8.1931e−47	2.1116e−90	5.4908e−46	26.8969	0.0527	2.3371e−04
	SD	3.4688e−92	3.3245e−47	7.4547e−91	9.0649e−47	0.3824	0.0835	1.3319e−04
CSGO2	BEST	6.9669e−92	6.8360e−47	4.3436e−89	1.8208e−45	25.9082	5.5511e−05	7.0864e−06
	WORST	7.8927e−90	3.1121e−46	2.0284e−88	5.0291e−45	27.0863	0.0064	5.0886e−04
	MEAN	9.5072e−91	1.7078e−46	1.1019e−88	3.3616e−45	26.4021	0.0015	1.9760e−04
	SD	1.6173e−90	5.7342e−47	4.4095e−89	7.5651e−46	0.3211	0.0018	1.3103e−04
CSGO3	BEST	**0**	**0**	**0**	**0**	25.4235	2.9837e−04	**1.7330e−05**
	WORST	**0**	**0**	**0**	**0**	27.1020	0.0063	**4.7404e−04**
	MEAN	**0**	**0**	**0**	**0**	26.5388	0.0017	**1.2603e−04**
	SD	**0**	**0**	**0**	**0**	0.3496	0.0014	**1.5155e−04**
CSGO4	BEST	1.6304e−106	4.2533e−54	1.5325e−105	1.1600e−53	**25.3050**	**9.9073e−05**	6.0227e−06
	WORST	4.6757e−106	1.0801e−53	4.2147e−105	1.8466e−53	**27.3137**	**0.0032**	5.5482e−04
	MEAN	3.3332e−106	7.4070e−54	2.8588e−105	1.4491e−53	**26.3448**	**6.0213e−04**	2.1049e−04
	SD	8.8831e−107	1.5647e−54	6.9615e−106	1.9302e−54	**0.4628**	**6.8272e−04**	1.6184e−04
CSGO5	BEST	1.1732e−84	3.5204e−43	1.1670e−82	2.7343e−42	25.5748	7.8039e−05	7.6121e−06
	WORST	2.2132e−83	1.1257e−42	4.1043e−82	5.5994e−42	26.9888	0.0216	4.1000e−04
	MEAN	5.6495e−84	7.0591e−43	2.4175e−82	4.7346e−42	26.4354	0.0028	1.7562e−04
	SD	4.2043e−84	2.0575e−43	8.2278e−83	7.5436e−43	0.3896	0.0055	1.1653e−04
CSGO6	BEST	1.2168e−83	7.8863e−43	6.6060e−81	4.8814e−41	25.5439	9.0940e−05	4.5065e−06
	WORST	1.0647e−81	4.4936e−42	8.9854e−80	1.1354e−40	27.1496	0.0180	4.2095e−04
	MEAN	2.1726e−82	2.7235e−42	4.3211e−80	8.6495e−41	26.4801	0.0031	1.5705e−04
	SD	3.0157e−82	9.9014e−43	1.8345e−80	1.6118e−41	0.4494	0.0046	1.2955e−04
CSGO7	BEST	1.9384e−131	1.6142e−66	9.1665e−131	2.2333e−66	25.1802	1.6804e−04	2.4556e−05
	WORST	5.9034e−131	3.6146e−66	1.8822e−130	3.8591e−66	26.9885	0.1273	5.5495e−04
	MEAN	3.7236e−131	2.6495e−66	1.3517e−130	3.1869e−66	26.4430	0.0107	1.9425e−04
	SD	1.0540e−131	4.8397e−67	2.9019e−131	4.0613e−67	0.4475	0.0276	1.3754e−04
CSGO8	BEST	2.0472e−59	1.9574e−30	5.6915e−57	7.7435e−29	26.0117	1.3110e−04	2.5367e−05
	WORST	2.1062e−56	7.5073e−29	2.2897e−55	2.5021e−28	28.4936	0.3408	4.8607e−04
	MEAN	6.4538e−57	2.1280e−29	8.8489e−56	1.8842e−28	27.2073	0.0998	2.2333e−04
	SD	6.9756e−57	2.0236e−29	8.8489e−56	4.4614e−29	0.6222	0.1128	1.3667e−04
CSGO9	BEST	5.7660e−27	4.0550e−14	1.5323e−26	8.3977e−14	26.2668	3.5632e−04	2.6211e−05
	WORST	3.9755e−26	8.6943e−14	1.2625e−25	1.1371e−13	27.8802	0.2051	6.3431e−04
	MEAN	2.0281e−26	6.6514e−14	7.4925e−26	9.6614e−14	27.2533	0.0183	2.9170e−04
	SD	9.2565e−27	1.1227e−14	2.8668e−26	8.1737e−15	0.4334	0.0426	1.7539e−04
CSGO10	BEST	3.8364e−65	7.9023e−34	1.1472e−63	1.6130e−32	25.4118	7.7545e−05	7.2578e−06
	WORST	1.1357e−63	8.0502e−33	1.2105e−62	4.0668e−32	27.0872	0.1149	3.4833e−04
	MEAN	3.0789e−64	3.2396e−33	6.2703e−63	3.0951e−32	26.4356	0.0058	1.4095e−04
	SD	2.7512e−64	2.0573e−33	2.7445e−63	5.9278e−33	0.4326	0.0210	1.0161e−04

Algo/functions		F8	F9	F10	F11	F12	F13
CSGO1	BEST	**− 1.2569e + 04**	**0**	**8.8818e−16**	**0**	3.4247e−06	5.1351e−05
	WORST	**− 1.2569e + 04**	**0**	**8.8818e−16**	**0**	0.0721	2.9668
	MEAN	**− 1.2569e + 04**	**0**	**8.8818e−16**	**0**	0.0041	1.8857
	SD	**0**	**0**	**0**	**0**	0.0135	1.3776
CSGO2	BEST	||	||	||	||	2.5402e−05	2.8052e−04
	WORST					3.8921e−04	0.1913
	MEAN					1.2914e−04	0.0223
	SD					9.3119e−05	0.0408

Table 2 (continued)

Algo/functions		F8	F9	F10	F11	F12	F13
CSGO3	BEST	||	||	||	||	2.4603e−05	**1.4202e−04**
	WORST					3.0965e−04	**0.1370**
	MEAN					1.2233e−04	**0.0225**
	SD					7.0016e−05	**0.0438**
CSGO4	BEST	||	||	||	||	4.6680e−06	1.8100e−04
	WORST					4.5161e−04	2.9661
	MEAN					6.9008e−05	0.5903
	SD					9.6678e−05	1.0957
CSGO5	BEST	||	||	||	||	7.3677e−06	9.1983e−05
	WORST					1.5076e−04	0.1374
	MEAN					4.4260e−05	0.0180
	SD					3.8735e−05	0.0361
CSGO6	BEST	||	||	||	||	2.2879e−05	2.6560e−04
	WORST					2.8905e−04	0.5182
	MEAN					1.2341e−04	0.0416
	SD					7.1090e−05	0.0996
CSGO7	BEST	||	||	||	||	**3.4340e−06**	9.4175e−05
	WORST					**1.1076e−04**	0.1416
	MEAN					**3.4620e−05**	0.0236
	SD					**2.9680e−05**	0.0437
CSGO8	BEST	||	||	||	||	2.5923e−05	7.4171e−05
	WORST					0.1062	2.9671
	MEAN					0.0115	1.3721
	SD					0.0317	1.3888
CSGO9	BEST	||	**0**	**2.2204e−14**	||	3.8476e−05	2.0210e−04
	WORST		42.7113	**5.4179e−14**		0.1041	2.4780
	MEAN		15.6828	**4.1981e−14**		0.0076	0.4208
	SD		11.9999	**7.7322e−15**		0.0236	0.8018
CSGO10	BEST	||	**0**	||	||	1.6514e−05	8.9642e−05
	WORST		**0**			3.6342e−04	0.2510
	MEAN		**0**			8.2203e−05	0.0345
	SD		**0**			7.1402e−05	0.0710

results are highlighted in bold. Table 4 reports p values of the WRS test [50] obtained at a 5% significance level of CSGO3 vs. other approaches. The p values less than 0.05 indicate that the null hypothesis is rejected, and p values that are 'NaN' mean both the input values are the same in Table 4, "−", "+", and "≈" denote that the performance of other approaches is worse, better, and similar to CSGO3, respectively.

Table 3 illustrates that the CSGO3 algorithm has the best results in most of the cases than the other compared algorithms for the analyzed benchmarks. As can be seen from Table 4, out of 130 cases, only in 7 cases, CSGO3 finds equivalent results, in 9 cases, CSGO3 finds the same solution, in 4 cases, CSGO3 finds a worse solution and in 110 cases, CSGO3 finds best results than others.

5.2 Structural Engineering Design Optimization Problems and Result Analysis

In structural engineering, design optimization problems are Constrained Optimization Problem (COP) which are highly nonlinear and design variables are involved under complex constraints. Such nonlinearity often results in multimodal response landscape. Subsequently, metaheuristic global optimization algorithms are used to obtain optimal solutions.

5.2.1 Constrained Optimization

A COP comprises of an objective function together with some equality and inequality constraints. Lower and upper bounds of design variables are often specified. Considering

Table 3 Comparatives results of all the algorithm

Algo/functions		F1	F2	F3	F4	F5	F6	F7
CSGO3	BEST	**0**	**0**	**0**	**0**	25.3901	1.9837e−04	**1.7330e−05**
	WORST	**0**	**0**	**0**	**0**	26.8591	0.0044	**4.7404e−04**
	MEAN	**0**	**0**	**0**	**0**	26.2578	0.0014	**1.2603e−04**
	SD	**0**	**0**	**0**	**0**	0.3584	0.0011	**1.5155e−04**
GSA	BEST	436.9429	13.9984	766.0158	7.2154	2.4878e+03	225.4227	0.0384
	WORST	1.5687e+03	24.0684	2.7470e+03	15.6014	1.1464e+05	1.4569e+03	0.2033
	MEAN	903.0623	18.5126	1.7746e+03	11.3408	5.4996e+04	794.7375	0.1178
	SD	262.9562	2.6219	556.2787	1.8408	3.0013e+04	337.8156	0.0430
	p value	1.2118e−12	1.2118e−12	1.2118e−12	1.2118e−12	3.0199e−11	3.0199e−11	3.0199e−11
WOA	BEST	1.4248e−36	2.5533e−25	3.7377e+04	0.1006	27.6150	0.2801	2.9395e−04
	WORST	6.9506e−32	6.1535e−21	8.2242e+04	86.3280	28.7512	0.7901	0.0141
	MEAN	3.6350e−33	9.5057e−22	5.6921e+04	53.0049	28.3088	0.5040	0.0035
	SD	1.2743e−32	1.4700e−21	1.1924e+04	25.8779	0.3428	0.1590	0.0033
	p value	1.2118e−12	1.2118e−12	1.2118e−12	1.2118e−12	3.0199e−11	3.0199e−11	4.9752e−11
HGSO	BEST	1.3176e−44	8.6475e−23	1.3733e−40	1.9796e−22	28.1002	2.9690	5.3104e−05
	WORST	8.0886e−37	5.2336e−20	1.1816e−30	3.2606e−18	28.8903	5.2215	0.0014
	MEAN	4.7928e−38	4.1484e−21	6.7414e−32	3.7472e−19	28.5835	4.3948	5.5643e−04
	SD	1.5871e−37	9.7595e−21	2.2612e−31	8.5215e−19	0.1979	0.5995	3.8452e−04
	p value	1.2118e−12	1.2118e−12	1.2118e−12	1.2118e−12	3.0199e−11	3.0199e−11	1.1674e−05
SOA	BEST	6.8182e−04	0.0010	0.0173	0.0445	28.3808	2.4362	0.0017
	WORST	0.0135	0.0118	11.5964	3.5227	30.5146	4.1644	0.0210
	MEAN	0.0039	0.0064	3.2540	1.0136	29.1680	3.4433	0.0093
	SD	0.0028	0.0030	3.2276	0.7859	0.4275	0.5082	0.0048
	p value	1.2118e−12	1.2118e−12	1.2118e−12	1.2118e−12	3.0199e−11	3.0199e−11	3.0199e−11
MPA	BEST	8.6823e−08	4.2558e−05	0.0820	0.0023	26.0262	0.0117	0.0011
	WORST	6.5764e−07	2.2550e−04	12.2886	0.0056	27.5269	0.1271	0.0067
	MEAN	3.4184e−07	1.2261e−04	4.1587	0.0040	26.6481	0.0509	0.0031
	SD	1.5179e−07	4.7189e−05	3.0020	9.1343e−04	0.3191	0.0274	0.0016
	p value	1.2118e−12	1.2118e−12	1.2118e−12	1.2118e−12	1.0407e−04	3.0199e−11	3.0199e−11
TSA	BEST	1.0867e−08	1.4400e−06	0.0076	0.4260	26.5432	2.5777	0.0046
	WORST	1.2222e−06	1.1132e−04	5.7078	5.9403	28.8875	4.6151	0.0275
	MEAN	2.6827e−07	3.0485e−05	0.9064	2.0998	28.2306	3.7481	0.0155
	SD	3.1555e−07	2.6800e−05	1.3829	1.5630	0.7376	0.5572	0.0071
	p value	1.2118e−12	1.2118e−12	1.2118e−12	1.2118e−12	6.6955e−11	3.0199e−11	3.0199e−11
SMA	BEST	7.9861e−246	6.1818e−126	4.6324e−236	1.0034e−118	**0.0392**	0.0015	1.2680e−05
	WORST	3.8549e−178	1.2516e−104	1.0741e−126	3.0687e−86	**28.4892**	0.2100	8.3208e−04
	MEAN	1.8931e−179	8.8060e−106	3.5804e−128	1.2459e−87	**13.3071**	0.0749	2.7421e−04
	SD	0	2.7416e−105	1.9611e−127	5.6918e−87	**12.1549**	0.0614	2.0137e−04
	p value	1.2118e−12	1.2118e−12	1.2118e−12	1.2118e−12	0.0773	1.0937e−10	0.0519
STOA	BEST	0.0062	0.0124	33.7096	1.3346	32.5351	2.4143	0.0076
	WORST	1.8019	0.1430	464.9817	5.3936	132.2677	7.0952	0.0378
	MEAN	0.6876	0.0599	171.6655	2.9803	69.9200	4.8385	0.0213
	SD	0.5518	0.0342	115.0174	0.9095	29.1551	1.3178	0.0090
	p value	1.2118e−12	1.2118e−12	1.2118e−12	1.2118e−12	3.0199e−11	3.0199e−11	3.0199e−11
HHO	BEST	2.0414e−55	2.5442e−29	1.3946e−46	1.8486e−28	3.7719e−04	**7.1900e−09**	2.5621e−05
	WORST	1.4064e−45	1.0948e−22	2.3887e−34	4.9473e−23	0.1390	**0.0020**	5.2680e−04
	MEAN	1.2981e−46	8.4362e−24	9.0082e−36	7.1281e−24	0.0318	**3.5589e−04**	1.8200e−04
	SD	3.4572e−46	2.5946e−23	4.3583e−35	1.4556e−23	0.0390	**4.7640e−04**	1.4454e−04
	p value	1.2118e−12	1.2118e−12	1.2118e−12	1.2118e−12	3.0199e−11	4.4440e−07	0.8650

Table 3 (continued)

Algo/functions		F1	F2	F3	F4	F5	F6	F7
GTA	BEST	5.3971e−17	8.5670e−15	3.1423e−17	1.1227e−14	28.8400	4.0850	0.0011
	WORST	1.4599e−14	2.4318e−11	2.1144e−14	3.7380e−12	28.9924	7.2961	0.0183
	MEAN	3.5119e−15	2.2610e−12	3.3347e−15	8.7064e−13	28.9434	6.5305	0.0098
	SD	3.8949e−15	4.9836e−12	5.5902e−15	1.0195e−12	0.0433	0.6297	0.0056
	p value	1.2118e−12	1.2118e−12	1.2118e−12	1.2118e−12	3.0199e−11	3.0199e−11	3.0199e−11

Algo/functions		F8	F9	F10	F11	F12	F13
CSGO3	BEST	**− 1.2569e + 04**	**0**	**8.8818e−16**	**0**	2.5600e−05	2.0376e−04
	WORST	**− 1.2569e + 04**	**0**	**8.8818e−16**	**0**	6.3783e−04	0.1099
	MEAN	**− 1.2569e + 04**	**0**	**8.8818e−16**	**0**	1.5034e−04	0.0201
	SD	**0**	**0**2.2204e−14	**0**	**0**	1.5686e−04	0.0339
GSA	BEST	− 3.5062e + 03	58.9623	10.5204	2.6322	1.3065	27.9534
	WORST	− 2.3590e + 03	210.6226	15.3128	10.9486	12.2971	1.9369e + 03
	MEAN	− 2.8128e + 03	110.2984	13.4418	7.4664	5.9810	172.0101
	SD	338.5363	47.2230	1.0551	1.9688	2.5709	360.8752
	p value	3.0199e−11	1.2118e−12	1.2118e−12	1.2118e−12	3.0199e−11	3.0199e−11
WOA	BEST	− 1.2565e + 04	**0**	4.4409e−15	**0**	0.0069	0.1840
	WORST	− 8.1341e + 03	5.6843e−14	2.2204e−14	0.2705	0.0454	1.0653
	MEAN	− 1.0336e + 04	1.8948e−15	1.0362e−14	0.0166	0.0236	0.5366
	SD	1.6644e + 03	1.0378e−14	4.5068e−15	0.0633	0.0106	0.1912
	p value	0.0076	0.3337	6.5558e−13	0.0815	3.0199e−11	3.0199e−11
HGSO	BEST	− 4.4071e + 03	**0**	**8.8818e−16**	**0**	0.2994	1.8041
	WORST	− 2.3000e + 03	**0**	**8.8818e−16**	**0**	0.6773	2.9023
	MEAN	− 3.3158e + 03	**0**	**8.8818e−16**	**0**	0.5024	2.7293
	SD	564.9030	**0**	**0**	**0**	0.0977	0.2383
	p value	3.0199e−11	NaN	NaN	NaN	3.0199e−11	3.0199e−11
SOA	BEST	− 6.7380e + 03	1.8329e−04	19.9583	0.0013	0.1010	1.6028
	WORST	− 4.4390e + 03	44.3976	19.9635	0.1264	0.7223	2.7332
	MEAN	− 4.9316e + 03	16.1990	19.9614	0.0668	0.3125	2.1890
	SD	421.7500	9.8505	0.0015	0.0413	0.1214	0.2703
	p value	3.6897e−11	1.2118e−12	1.2118e−12	1.2118e−12	3.0199e−11	3.0199e−11
MPA	BEST	− 9.8527e + 03	1.5903e−06	6.5629e−05	2.7181e−07	6.6609e−04	0.0246
	WORST	− 7.4931e + 03	0.0015	1.7377e−04	2.1563e−06	0.0082	0.1567
	MEAN	− 8.3146e + 03	1.7403e−04	1.2064e−04	1.1734e−06	0.0031	0.0702
	SD	529.4453	3.3850e−04	2.8878e−05	5.6156e−07	0.0031	0.0395
	p value	1.6351e−05	1.2118e−12	1.2118e−12	1.2118e−12	3.0199e−11	3.2555e−07
TSA	BEST	− 6.8763e + 03	110.1055	1.1229e−04	6.8352e−09	1.3962	1.8352
	WORST	− 5.3556e + 03	237.5861	3.7481	0.0274	18.7804	4.4032
	MEAN	− 6.0460e + 03	189.5021	2.2329	0.0088	8.9181	3.0443
	SD	490.7734	33.7897	1.5160	0.0121	4.3787	0.5598
	p value	1.0937e−10	1.2118e−12	1.2118e−12	1.2118e−12	3.0199e−11	3.0199e−11
SMA	BEST	− 1.2569e + 04	**0**	**8.8818e−16**	**0**	1.6053e−05	5.2757e−04
	WORST	− 1.2564e + 04	**0**	**8.8818e−16**	**0**	0.0276	0.0683
	MEAN	− 1.2568e + 04	**0**	**8.8818e−16**	**0**	0.0076	0.0187
	SD	1.3005	**0**	**0**	**0**	0.0080	0.0186
	p value	0.0773	NaN	NaN	NaN	1.1737e−09	0.0451

Table 3 (continued)

Algo/functions		F8	F9	F10	F11	F12	F13
STOA	BEST	− 5.8237e+03	2.1453	0.7407	0.0921	0.0968	1.9846
	WORST	− 4.4813e+03	89.1702	19.9626	0.9397	2.2575	4.7258
	MEAN	− 5.0923e+03	42.6491	19.3197	0.5811	0.9978	3.2201
	SD	307.5817	23.9185	3.5090	0.2239	0.5344	0.7318
	p value	3.0199e−11	1.2118e−12	1.2118e−12	1.2118e−12	3.0199e−11	3.0199e−11
HHO	BEST	− 1.2569e+04	**0**	**8.8818e−16**	**0**	**1.8057e−09**	**2.2821e−06**
	WORST	− 1.2566e+04	**0**	**8.8818e−16**	**0**	**6.7745e−05**	**4.6801e−04**
	MEAN	− 1.2568e+04	**0**	**8.8818e−16**	**0**	**2.0373e−05**	**1.1551e−04**
	SD	1.1436	**0**	**0**	**0**	**2.1029e−05**	**1.3780e−04**
	p value	0.0773	NaN	NaN	NaN	4.9980e−09	5.5727e−10
GTA	BEST	− 6.0633e+03	**0**	1.8652e−14	**0**	0.5241	2.2501
	WORST	− 2.3456e+03	5.6843e−14	5.8771e−12	4.7184e−14	1.3979	3.1935
	MEAN	− 1.3501e+03	1.8948e−15	8.8386e−13	7.4533e−15	0.9688	2.9359
	SD	1.1853e+03	1.0378e−14	1.3506e−12	1.2642e−14	0.2434	0.1828
	p value	3.0199e−11	0.3337	1.2088e−12	1.9280e−09	3.0199e−11	3.0199e−11

The best results of the experiments are represented in bold

Table 4 WRS test results on Table 2

Algo/Functions	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13
GSA	−	−	−	−	−	−	−	−	−	−	−	−	−
WOA	−	−	−	−	−	−	−	−	≈	−	≈	−	−
HGSO	−	−	−	−	−	−	−	−	NaN	NaN	NaN	−	−
SOA	−	−	−	−	−	−	−	−	−	−	−	−	−
MPA	−	−	−	−	−	−	−	−	−	−	−	−	−
TSA	−	−	−	−	−	−	−	−	−	−	−	−	−
SMA	−	−	−	−	≈	−	≈	≈	NaN	NaN	NaN	−	+
STOA	−	−	−	−	−	−	−	−	−	−	−	−	−
HHO	−	−	−	−	−	+	≈	≈	NaN	NaN	NaN	+	+
GTA	−	−	−	−	−	−	−	−	≈	−	−	−	−
Total no. of '−' = 112, total no. of '+' = 2, total no. of '≈' = 7, total no. of '*NaN*' = 9													

"−", "+", and "≈" denote that the performance of other approaches are worse, better, and similar to CSGO3, respectively

that there are n design variables, then COP can be written in following form:

$$\left.\begin{array}{c} Minimize\ : f(X) \\ Subjectto\ : g_i(X) \leq 0, \quad i = 1, 2, \ldots\ldots, m \\ h_k(X) = 0 \quad k = 1, 2, \ldots\ldots, p \\ a_j \leq x_j \leq b_j \quad j = 1, \ldots\ldots, n \\ X = \left(x_1, x_2, \ldots, x_n\right) \end{array}\right\} \tag{14}$$

where the function $f(X)$ is objective function which is to be minimized. The functions $g_i(X)$ and $h_k(X)$ are inequality and equality constraint functions, respectively. There are m inequality constraints and p equality constraints in the above problem. This problem is a nonlinear optimization problem if at least one of the functions $f(X)$, $g_i(X)$ or $h_k(X)$ is nonlinear.

Most metaheuristic algorithms are normally designed to work on unconstrained search spaces. Solving COPs using metaheuristic algorithms requires additional mechanisms to incorporate the effects of constraints into their objective function. While solving COPs, it has become necessary to deal with both feasible and infeasible solutions, dealing with the latter having more concerns. It may be possible to ignore all the infeasible solutions but as metaheuristic algorithms are stochastic search methods, completely discarding the infeasible solutions may results in a loss of information about some promising regions of the function landscape.

To remove this confusion and to solve this problem, there is a traditional approach that imposes a penalty [51] for the infeasible solutions. A constraint violation is included for the penalized candidate solutions. Then, the penalized candidate solutions are handled as an unconstrained objective function that can be optimized using the unconstrained optimizing technique.

5.2.2 Constraint Violation

The constraint violation $V(X)$ is the measure that indicates by how much a candidate solution X violates the given constraints:

$$\left.\begin{array}{ll} V(X) = 0; & if\ X \in F \\ V(X) > 0; & if\ X \notin F \end{array}\right\}, \text{where} F \text{ is the feasible region} \tag{15}$$

Generally, evaluation of constraint violation in the COP is done using the following two equations:

$$V(X) = max\left\{ \max_i \{0, g_i(X)\}, \max_k |h_k(X)| \right\} \tag{16}$$

$$V(X) = \sum\nolimits_i \max\{0, g_i(X)\}^m + \sum\nolimits_k \{|h_k(X)|^m \tag{17}$$

In our study, we have used the approach (17) with m = 2.

5.2.3 Constraint Handling

In COP, the constraint handling technique is a necessary criterion to reach the optimal solution within the feasible region (if exists). This is mainly to exploit the infeasible candidate solutions and extract effective information for the stochastic search process. Depending on the constraint violation and the objective function value, Deb's rules [52] have been chosen for handling constraints.

While solving a COP, it is very difficult to handle the situation if some active constraint is present. All equality constraints are active constraints and for the inequality constraints those satisfy $g_i(X) = 0$ at the global optimum solution are called active constraints. Therefore, the problems with equality constraint should be handled evasively for a high-quality solution. The equality constraints can be altered into the inequality form and can easily be combined with the inequality constraint. Lots of techniques have been used for this particular operation. Here, we use a tolerance parameter (t_p) to for converting the equality constraints into inequality form. Therefore, the constraints of Eq. (14) can be written as

$$G_{ineq}(X) = \left\{ \begin{array}{ll} max\{g_i(X), 0\}, & i = 1, \ldots\ldots, m \\ max\{|h_i(X)| - t_{p,}, 0\}, & i = 1, \ldots\ldots, p \end{array} \right. \tag{18}$$

where $G_{\text{ineq}}(X)$ is the inequality constraints, and t_p is a tolerance parameter for the equality constraints.

Thus, the objective is to minimize the fitness function $f(X)$ such that the optimal solution obtained satisfies all the inequality constraints $G_{\text{ineq}}(X)$.

5.2.4 Structural Engineering Design Problems

The performance of the family of CSGO algorithms are demonstrated in this paper through solving nine structural engineering design problems and the performances are compared with many state-of-the-art as well as latest metaheuristic bionic algorithm algorithms of literature.

5.2.4.1 Parameter Settings and Evaluation Criterion

- *Stopping criterion:* Maximum number of function evaluations 20,000.
- *Runs*: 30 independent runs
- *Statistical results*: best (BEST), mean (MEAN), worst (WORST), and standard deviation (SD)
- *Constraints handling*: Deb's rules [52]
- Initial point in chaos theory set to 0.7 for all chaos maps

The parameter settings for all the algorithms considered for statistical results comparisons are kept the same as mentioned in their respective papers.

Here, we have applied a rule that the infeasible solutions containing slight violation of the constraints (from 0.01 in the first iteration to 0.001 in the last iteration) are considered as feasible solutions. For most structural optimization problems, the global minimum locates on or close to the boundary of a feasible design space. By applying this rule, the candidate solutions approach the boundaries and can reach the global minimum with a higher probability[53].

5.2.4.2 Tension/Compression Spring Design Problem The objective is the minimization of the fabrication cost of spring with three parameters and four constrains such as wire diameter (x_1), spring coil diameter (x_2), and a number of active coils (x_3), and deflection ($g_1(X)$), shear stress ($g_2(X)$), surge frequency ($g_3(X)$), and outer diameter limit ($g_4(X)$). The spring design pattern is shown in Fig. 1 and formulated optimization problem is referred from [54].

The optimization results and the statistical results are given in Tables 5 and 6, respectively. The result of the PO, EO, PRO, CSA, MFO and Emperor Penguin Optimizer (EPO) algorithms are imported from [55–59] and [46], respectively. The result for the HGSO, MPA, TSA, and STOA algorithms are reported from [43, 45, 46], and [48], respectively. For SHO, SCA, GWO, PSO, MVO, GSA, GA, and DE, the results are imported from [44]. For CS, WOA, EHO (elephant herbing behavior), and SA, the results are imported from [43]. The results for the CSGO family are achieved by us. In the tables, "−" represents "not given"

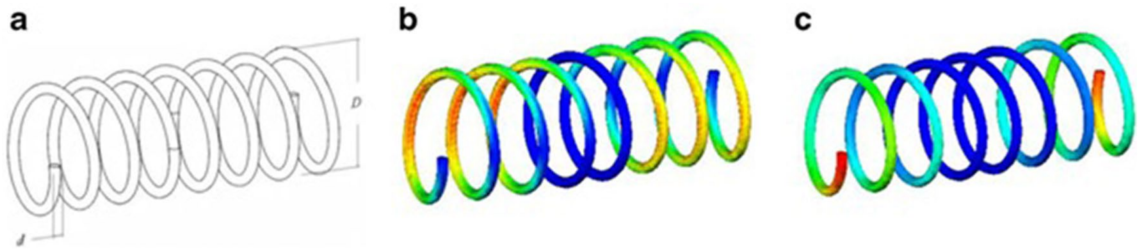

Fig. 1 **a** Schematic of the spring; **b** stress distribution evaluated at the optimum design and **c** displacement distribution evaluated at the optimum design

Table 5 Optimum results of tension/compression spring design problem by different algorithms

Algorithms	x_1	x_2	x_3	Optimal result
PRO	0.051819	0.359873	11.106372	0.012665
MFO	0.051994	0.364109	10.868421	0.012666
PO	0.05248	0.37594	10.24509	0.012678508650926
HGSO	0.0518	0.3569	11.2023	0.01265
EO	0.0516199100	0.355054381	11.38796759	0.012666132128342
MPA	0.051724477	0.35757003	11.2391955	0.012665283702960
TSA	0.051080	0.342890	12.0890	0.012655520
EPO	0.051087	0.342908	12.0898	0.012656987
STOA	0.051090	0.342910	12.0900	0.012656990
SHO	0.051144	0.343751	12.0955	0.012674000
GWO	0.050178	0.341541	12.07349	0.012678321
PSO	0.05	0.310414	15	0.013192580
MVO	0.05	0.315956	14.22623	0.012816930
SCA	0.050780	0.334779	12.72269	0.012709667
CSA	0.0516890284	0.3567169544	11.2890117993	0.0126652328
GSA	0.05000	0.317312	14.22867	0.012873881
GA	0.05010	0.310111	14.0000	0.013036251
DE	0.05025	0.316351	15.23960	0.012776352
CS	0.0518	0.3586	11.1808	0.0127
WOA	0.0520	0.3637	10.8938	0.0127
EHO	0.0580	0.5278	5.5820	0.0135
SA	0.0500	0.2500	9.3876	0.0178
CSGO1	0.051868506586171	0.361329001734979	11.010194291412466	0.012647191400445
CSGO2	0.052147921766084	0.368152633913749	10.635947794752985	0.012650559847160
CSGO3	0.051600716157639	0.354879861075208	11.383737998864271	0.012646496816940
CSGO4	0.051553871747610	0.353758785482019	11.450726981522376	0.012646651920237
CSGO5	0.051719671443382	0.357742166397375	11.215610175557737	0.012646456860643
CSGO6	0.051449941824867	0.351252263328912	11.603418508471799	0.012648440051873
CSGO7	0.051645763177484	0.355951578798280	11.320184517762158	0.012646506261597
CSGO8	0.051556912106367	0.353831660202976	11.446357218503547	0.012646639281262
CSGO9	0.051567172618997	0.354077252584201	11.431641386323564	0.012646599197901
CSGO10	0.051625142880021	0.355467751056912	11.348907315462579	0.012646459938708

for that particular algorithm. The best result is highlighted in bold. From Tables 5 and 6, it can be inferred that the CSGO family has found better optimal results than other. Again in CSGO family, CSGO5 finds better optimal result than others.

5.2.4.3 The Welded Beam Design Problem The objective is to minimize the manufacturing cost of the welded beam with four optimized variables and seven constrains such as thickness of the weld (x_1), length of clamped bar (x_2), the height of the bar (x_3), and thickness of the bar (x_4), and

Table 6 Statistical results of tension/compression spring design problem by different algorithms

Algorithms	BEST	WORST	MEAN	SD
PRO	–	–	–	–
MFO	–	–	–	–
PO	0.0127	0.0128	0.0127	0.0000
HGSO	0.01265	0.01278	0.0127	8.09E−07
EO	0.012666	0.013997	0.013017	3.91E-04
MPA	0.012665	0.012665	0.012665	5.55E-08
TSA	0.012655520	0.012667890	0.012677560	0.001010
EPO	0.012656987	0.012667902	0.012678903	0.001021
SOA	**0.01264522**	**0.012665417**	**0.012665871**	**0.001108**
SHO	0.01267400	0.012715185	0.012684106	0.000027
GWO	0.01267832	0.012720757	0.012697116	0.000041
PSO	0.01319258	0.017862507	0.014817181	0.002272
MVO	0.01281693	0.017839737	0.014464372	0.001622
CSA	0.0126652328	0.0126701816	0.0126659984	1.357079e−06
SCA	0.01270967	0.012998448	0.012839637	0.000078
GSA	0.01287388	0.014211731	0.013438871	0.000287
GA	0.01303625	0.016251423	0.014036254	0.002073
DE	0.01277635	0.015214230	0.013069872	0.000375
CS	0.0127	0.0127	0.0127	1.09E−06
WOA	0.0127	0.0178	0.0140	0.0014
EHO	0.0135	0.0189	0.0155	0.0011
SA	0.0178	0.0200	0.0184	5.90E−04
CSGO1	0.012647191400445	0.012705380460395	0.012671914370158	1.594855570095636e−05
CSGO2	0.012650559847160	0.012723177841050	0.012675734191303	2.039788119004384e−05
CSGO3	0.012646496816940	0.012723145234312	0.012670310798585	1.861631631708472e−05
CSGO4	0.012646651920237	0.012717738004471	0.012678120743460	2.042051003113493e−05
CSGO5	**0.012646456860643**	**0.012722315376422**	**0.012667662137365**	**1.979728990700402e−05**
CSGO6	0.012648440051873	0.012730907629073	0.012682782205489	2.400160579069954e−05
CSGO7	0.012646506261597	0.012698207423571	0.012668336535410	1.594780501287227e−05
CSGO8	0.012646639281262	0.012715944622341	0.012671778489487	1.789174157893117e−05
CSGO9	0.012646599197901	0.012709183712174	0.012677452370361	1.652317447164576e−05
CSGO10	0.012646459938708	0.012735913086394	0.012674051265553	2.142122522963287e−05

The best results of the experiments are represented in bold

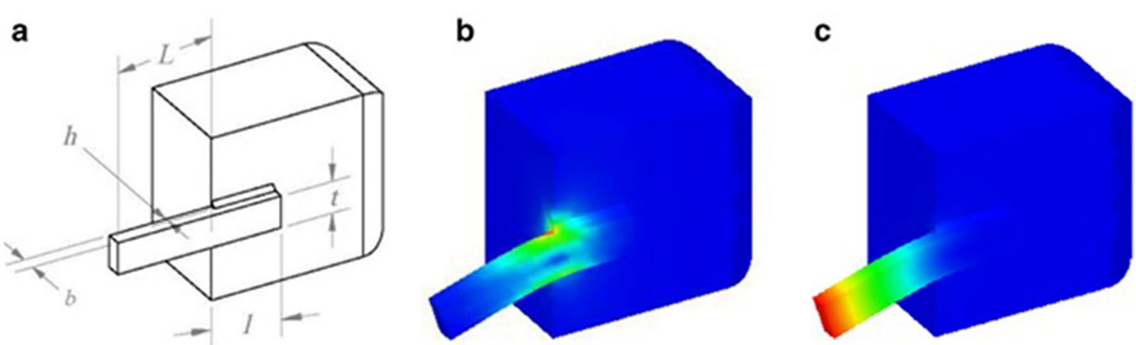

Fig. 2 Welded beam design problem: **a** Schematic of the weld, **b** stress distribution evaluated at the optimum design, **c** displacement distribution at the optimum design

Table 7 Optimum results of welded beam design by different algorithms

Algorithms	x_1	x_2	x_3	x_4	Optimal result
PO	0.205730	3.470472	9.036624	0.205730	1.724851
SMA	0.2054	3.2589	9.0384	0.2058	1.69604
MFO	0.2057	3.4703	9.0364	0.2057	1.72452
SSA	0.2057	3.4714	9.0366	0.2057	1.72491
HGSO	0.2054	3.4476	9.0269	0.2060	1.7260
EO	0.2057	3.4705	9.03664	0.2057	1.7249
MPA	0.205728	3.470509	9.036624	0.205730	1.724853
TSA	0.203290	3.471140	9.035100	0.201150	1.721020
EPO	0.205411	3.472341	9.035215	0.201153	1.723589
STOA	0.205415	3.472346	9.035220	0.201160	1.723590
SOA	0.205408	3.472316	9.035208	0.201141	1.723485
SHO	0.205563	3.474846	9.035799	0.205811	1.725661
GWO	0.205678	3.475403	9.036964	0.206229	1.726995
PSO	0.197411	3.315061	10.00000	0.201395	1.820395
MVO	0.205611	3.472103	9.040931	0.205709	1.725472
CSA	0.2057296398	3.4704886656	9.0366239104	0.2057296398	1.7248523086
SCA	0.204695	3.536291	9.004290	0.210025	1.759173
GSA	0.147098	5.490744	10.00000	0.217725	2.172858
GA	0.164171	4.032541	10.00000	0.223647	1.873971
DE	0.206487	3.635872	10.00000	0.203249	1.836250
CS	0.2057	3.4705	9.0366	0.2057	1.7289
WOA	0.1876	3.9298	8.990	0.2308	1.9428
EHO	0.4834	2.4950	4.4538	0.8488	2.3234
SA	0.4834	3.4751	9.0417	0.2063	1.7306
CSGO1	**0.20573671085387**	**3.2529941267793**	**9.0366239103577**	**0.205729639786075**	**1.695240471336402**
CSGO2	**0.20573671085387**	**3.2529941267793**	**9.0366239103577**	**0.205729639786075**	**1.695240471336402**
CSGO3	**0.20573671085387**	**3.2529941267793**	**9.0366239103577**	**0.205729639786075**	**1.695240471336402**
CSGO4	**0.20573671085387**	**3.2529941267793**	**9.0366239103577**	**0.205729639786075**	**1.695240471336402**
CSGO5	**0.20573671085387**	**3.2529941267793**	**9.0366239103577**	**0.205729639786075**	**1.695240471336402**
CSGO6	**0.20573671085387**	**3.2529941267793**	**9.0366239103577**	**0.205729639786075**	**1.695240471336402**
CSGO7	**0.20573671085387**	**3.2529941267793**	**9.0366239103577**	**0.205729639786075**	**1.695240471336402**
CSGO8	**0.20573671085387**	**3.2529941267793**	**9.0366239103577**	**0.205729639786075**	**1.695240471336402**
CSGO9	**0.20573671085387**	**3.2529941267793**	**9.0366239103577**	**0.205729639786075**	**1.695240471336402**
CSGO10	**0.20573671085387**	**3.2529941267793**	**9.0366239103577**	**0.205729639786075**	**1.695240471336402**

The best results of the experiments are represented in bold

shear stress (τ), and bending stress in the beam (θ), buckling load (Pc), end deflection of beam (δ), normal stress (σ), and boundary. The design of welded beam is shown in Fig. 2 and formulated optimization problem is referred from [54].

The optimization results and statistical results are given in Tables 7 and 8, respectively. The result of the PO, EO, PRO, CSA, MFO and Emperor Penguin Optimizer (EPO) algorithms are imported from [55–59] and [56], respectively. The result for the HGSO, SOA, MPA, TSA, and STOA algorithms are imported from [43–46], and [48], respectively. For SMA, SSA the result is imported from [57]. For SHO, GWO, PSO, MVO, SCA, GSA, GA, and DE, the results are imported from [44]. For CS, WOA, EHO (elephant herbing behavior), and SA, the results are imported from [43]. The results for the CSGO family are achieved by us. In the tables, "–" represents "not given" for that particular algorithm. The best result is highlighted in bold. From Tables 7 and 8, it can be stated that the CSGO family of algorithms outperforms all other algorithms. Here, all algorithms of the CSGO family perform equally well in finding optimal results.

5.2.4.4 Cantilever Beam Design Problem It is made up of five hollow square cross-sections as Fig. 3 [54]. Regarding this problem, detailed description and formulated optimization problem is referred from [54].

The optimization results and the statistical results of algorithms are given in Tables 9 and 10, respectively. The result SMA, MFO, SOS (Symbiotic Organisms Search), CS,

Table 8 Statistical results of welded beam design problem by different algorithms

Algorithms	BEST	WORST	MEAN	SD
PO	1.724851	1.724852	1.724851	2.53E-07
SMA	–	–	–	–
MFO	–	–	–	–
SSA	–	–	–	–
HGSO	1.7260	1.7325	1.7265	7.66E−03
EO	1.724853	1.736725	1.726482	0.003257
MPA	1.724853	1.724873	1.724861	6.41E-06
STA	1.721020	1.727205	1.725021	0.003316
EPO	1.723589	1.727211	1.725124	0.004325
STOA	1.723590	1.727215	1.725126	0.004330
SOA	1.723485	1.727102	1.724251	0.005967
SHO	1.725661	1.726064	1.725828	0.000287
GWO	1.726995	1.727564	1.727128	0.001157
PSO	1.820395	3.048231	2.230310	0.324525
MVO	1.725472	1.741651	1.729680	0.004866
CSA	1.7248523086	1.7248523086	1.7248523086	1.19450917e−15
SCA	1.759173	1.873408	1.817657	0.027543
GSA	2.172858	3.003657	2.544239	0.255859
GA	1.873971	2.320125	2.119240	0.034820
DE	1.836250	2.035247	1.363527	0.139485
CS	1.7289	1.7250	1.7276	2.89E−05
WOA	1.9428	5.9905	3.3865	0.8251
EHO	2.3234	4.8541	3.5058	0.5536
SA	1.7288	1.7332	1.7380	0.0026
CSGO1	1.695240471336402	1.695240471336402	1.695240471336402	6.115790034619237e–16
CSGO2	1.695240471336402	1.695240471336402	1.695240471336402	6.334279577469838e–16
CSGO3	1.695240471336402	1.69524047133640	1.695240471336402	6.467088141258292e–16
CSGO4	1.695240471336402	1.695240471336402	1.695240471336402	6.239631870765690e–16
CSGO5	1.695240471336402	1.695240471336402	1.695240471336402	6.973073502539061e–16
CSGO6	1.695240471336402	1.695240471336404	1.695240471336402	6.762657090795278e–16
CSGO7	1.695240471336402	1.695240471336402	1.695240471336402	6.825217693107653e–16
CSGO8	1.695240471336402	1.695240471336403	1.695240471336402	6.825217693107653e–16
CSGO9	1.695240471336402	1.695240471336403	1.695240471336402	6.911851303463225e–16
CSGO10	1.695240471336402	1.695240471336402	1.695240471336402	7.105905885894476e–16

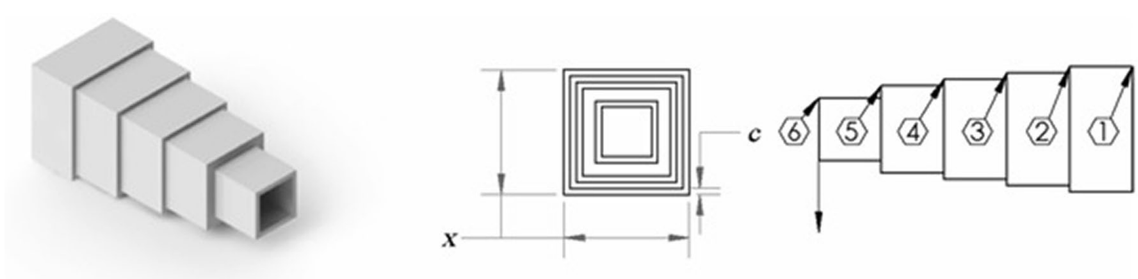

Fig. 3 Cantilever beam design

MMA (Method of Moving Asymptotes), and GCA (Generalised Convex Approximation) are imported from [47]. The results for the CSGO family are achieved by us. The "−" represents "not given" for that particular algorithm. The best result is highlighted in bold. From Tables 9 and 10, it can be inferred that the CSGO family of algorithm outperform all other algorithms. Among the CSGO family of algorithms, CSGO1 finds the best optimal solution, whereas CSGO8 finds the best solution in terms of BEST, WORST, MEAN, and SD solution among others.

Table 9 Optimum results of cantilever beam design problem by different algorithms

Algorithms	x_1	x_2	x_3	x_4	x_5	Optimal result
SMA	6.017757	5.310892	4.493758	3.501106	2.150159	1.339957* 1.339957c
MFO	5.9830	5.3167	4.4973	3.5136	2.1616	1.33998* 1.33998c
SOS	6.0188	5.3034	4.4959	3.4990	2.1556	1.33996* 13.3996c
CS	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999* 13.3999c
MMA	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400* 13.400c
GCA	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400* 13.400c
CSGO1	6.0143069125645	5.3080639580584	4.4936917574974	3.5006821291728	2.1518561741050	**13.3620572197022**
CSGO2	6.0150727517961	5.3073755398209	4.4931808855624	3.5006393848484	2.1523324379117	13.3620572623622
CSGO3	6.0145326449192	5.3076950955757	4.4931185679383	3.5011268561112	2.1521279556988	13.3620573372395
CSGO4	6.0138874235950	5.3079214693168	4.4932750108494	3.5013219117263	2.1521954499936	13.3620574276354
CSGO5	6.0149990750964	5.3075027372209	4.4934538417099	3.5002586185344	2.1523867492680	13.3620572759868
CSGO6	6.0151106132899	5.3084991455319	4.4926672877461	3.5003628791237	2.1519613721669	13.36205744778727
CSGO7	6.0142224790303	5.3079146553709	4.4931574308163	3.5008169029355	2.1524895464754	13.36205727150465
CSGO8	6.0148823577978	5.3072968983588	4.4933942605634	3.5007556408336	2.1522719285524	13.36205731599240
CSGO9	6.0145090595848	5.3071963634624	4.4935926982223	3.5007769877979	2.1525259502884	13.36205729934305
CSGO10	6.0140122013280	5.308292289245	4.4935032881358	3.5002309588785	2.1525624547352	13.36205738210221

*Represents wrongly put

cRepresents corrected value

Table 10 Statistical results of cantilever beam design problem by different algorithms

Algorithms	BEST	WORST	MEAN	SD
SMA	–	–	–	–
MFO	–	–	–	–
SOS	–	–	–	–
CS	–	–	–	–
MMA	–	–	–	–
GCA	–	–	–	–
CSGO1	**13.362057219702205**	13.362060325929614	13.362058423000358	8.593615875833658e–07
CSGO2	13.362057262362285	13.362061867644758	13.362058785952119	1.299590279055970e–06
CSGO3	13.362057337239426	13.362061311531937	13.362058932062419	9.909115197252335e–07
CSGO4	13.362057427635454	13.362060428644744	13.362058414070182	8.861240402167021e–07
CSGO5	13.362057275986803	13.362061591038888	13.362058600135557	1.028567170707533e–06
CSGO6	13.362057447787276	13.362061960283475	13.362058518321360	9.275908881267436e–07
CSGO7	13.362057271504650	13.362061236823928	13.362058766562427	1.103333599646931e–06
CSGO8	13.362057315992399	**13.362060253167591**	**13.362058364060077**	**7.399711676054027e–07**
CSGO9	13.362057299343050	13.362060339061168	13.362058575093503	9.682425313271179e–07
CSGO10	13.362057382102218	13.362063301098216	13.362058788800560	1.304095221610504e–06

The best results of the experiments are represented in bold

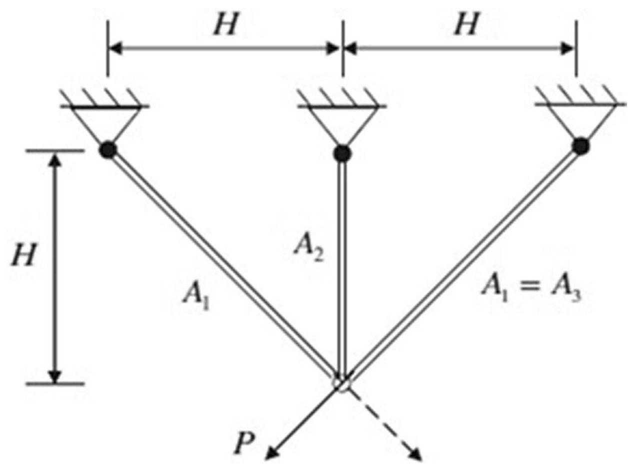

Fig. 4 Three-bar truss design

5.2.4.5 Three-Bar Truss Design Problem The objective is to design a truss with a minimum weight that does not violate constraints. Regarding this problem, detailed description is given in [54]. Figure 4 [54] shows the structural parameters of this problem. The formulated design problem is referred from [42].

The optimization result of algorithms with constraint function values is given in Table 11, and the statistical result is given in Table 12. The result PRO (Poor and Rich Optimization), GOA, MFO, PSO-DE, ALO, and MVO are imported from [57], and for CSA, the result is imported from [58]. The results for the CSGO family are achieved by us. The constraint value in Table 13 is calculated by us using optimization results as imported by the authors in their respective papers. The "–" represents "not given" for that particular algorithm. The best result is highlighted in bold. From Tables 11 and 12, it can be stated that the CSGO family of algorithms outperforms all other algorithms. Among the CSGO family of algorithms, CSGO4 finds the best optimal solution, whereas CSGO1 finds the best solution in terms of worst, mean, and standard deviation solution among others.

5.2.4.6 I-Beam Design Problem The objective is to minimize the vertical deflection of an I-beam with four optimization variables and two optimization constraints such as cross-section area ($g_1(X)$), and bending stress ($g_2(X)$) as Fig. 5 [59]. This case is modified from the original problem reported in [60]. The formulated optimization problem is referred from [59].

The optimization result of algorithms with constraint function values is given in Table 13, and the statistical result is given in Table 14. The result of ARSM, Improved ARSM, and CS are imported from [59]. The results for the CSGO family are achieved by us. "–" represents "not given" for that particular algorithm. The best result is highlighted in bold From Tables 13 and 14, it can be deduced that the CSGO family of algorithms outperforms all other algorithms. Here, all algorithms of CSGO family find equivalent results. Hence, all CSGO algorithm finds the best optimal solution

Table 11 Optimum results of three-bar truss design with constraint values by different algorithms

Algorithms	x_1	x_2	Optimal result
PRO	0.7886475	0.4083262	263.8958439
GOA	0.7888975	0.4076195	263.8958814
MFO	0.7882447	0.4094669	263.8959796
PSO-DE	0.7886751	0.4082482	**263.8958433**
ALO	0.7886628	0.4082831	263.8958434
MVO	0.7886027	0.4084530	263.8958499
CSA	0.7886751284	0.4082483080	263.8958433765
CSGO1	0.785910302125472	0.406771162753979	**2.629661178903138e+02**
CSGO2	0.785902703879913	0.406792652722559	2.629661177787879e+02
CSGO3	0.785910438238492	0.406770779370732	2.629661180505648e+02
CSGO4	0.785897853448955	0.406806371549532	2.629661177524364e+02
CSGO5	0.785899014791685	0.406803086881755	2.629661177629863e+02
CSGO6	0.785889546201876	0.406829868373055	2.629661177904916e+02
CSGO7	0.785900217084501	0.406799686270454	2.629661177616174e+02
CSGO8	0.785889712034763	0.406829399279497	2.629661177857593e+02
CSGO9	0.785906353496849	0.406782330489170	2.629661178230024e+02
CSGO10	0.785892039451910	0.406822817331165	2.629661178839051e+02

The best results of the experiments are represented in bold

Table 12 Statistical results of three-bar toss design by different algorithms

Algorithms	BEST	WORST	MEAN	STD
PRO	–	–	–	–
GOA	–	–	–	–
MFO	–	–	–	–
PSO-DE	–	–	–	–
ALO	–	–	–	–
MVO	–	–	–	–
CSA	263.8958433770	263.8958433770	263.8958433770	1.0122543402e−10
CSGO1	2.629661178903138e+02	**2.629661311442690e+02**	**2.629661217343362e+02**	**3.426468153983831e−06**
CSGO2	2.629661177787879e+02	2.629661351706924e+02	2.629661224914092e+02	4.479771727973893e−06
CSGO3	2.629661180505648e+02	2.629661524424530e+02	2.629661262757790e+02	8.898346072064122e−06
CSGO4	**2.629661177524364e+02**	2.629661395505302e+02	2.629661239386700e+02	6.429962565899163e−06
CSGO5	2.629661177629863e+02	2.629661462439137e+02	2.629661238138859e+02	7.489793389126740e−06
CSGO6	2.629661177904916e+02	2.629661407947427e+02	2.629661246397652e+02	6.876621127163010e−06
CSGO7	2.629661177616174e+02	2.629661495027468e+02	2.629661256394791e+02	8.120192773574671e−06
CSGO8	2.629661177857593e+02	2.629661475561125e+02	2.629661258534684e+02	9.245721043128937e−06
CSGO9	2.629661178230024e+02	2.629661400469813e+02	2.629661236594902e+02	6.698724976365791e−06
CSGO10	2.629661178839051e+02	2.629661417584532e+02	2.629661275134199e+02	7.425143328221043e−06

The best results of the experiments are represented in bold

Fig. 5 I-beam design problem

Table 13 Optimum results of I-beam design problem with value of constraints function by different algorithms

Algorithms	x_1	x_2	x_3	x_4	Optimal result
ARSM	80	37.05	1.71	2.31	0.0157
Improved ARSM	79.99	48.42	0.90	2.40	0.0131
CS	80	50	0.9000	2.3216715	0.0130747
CSGO1	80	50	1.764718360707903	5	**0.006620471665250**
CSGO2	80	50	1.764718360707903	5	**0.006620471665250**
CSGO3	80	50	1.764718360707903	5	**0.006620471665250**
CSGO4	80	50	1.764718360707903	5	**0.006620471665250**
CSGO5	80	50	1.764718360707903	5	**0.006620471665250**
CSGO6	80	50	1.764718360707903	5	**0.006620471665250**
CSGO7	80	50	1.764718360707903	5	**0.006620471665250**
CSGO8	80	50	1.764718360707903	5	**0.006620471665250**
CSGO9	80	50	1.764718360707903	5	**0.006620471665250**
CSGO10	80	50	1.764718360707903	5	**0.006620471665250**

The best results of the experiments are represented in bold

Table 14 Statistical results of I-beam design problem by different algorithms

Algorithms	BEST	WORST	MEAN	STD
ARSM	–	–	–	–
Iproved ARSM	–	–	–	–
CS	0.0130747	0.01353646	0.0132165	0.0001345
CSGO1	**0.006620471665250**	**0.006620471665250**	**0.006620471665250**	**3.528758033802314e−18**
CSGO2	**0.006620471665250**	**0.006620471665250**	**0.006620471665250**	**3.528758033802314e−18**
CSGO3	**0.006620471665250**	**0.006620471665250**	**0.006620471665250**	**3.528758033802314e−18**
CSGO4	**0.006620471665250**	**0.006620471665250**	**0.006620471665250**	**3.528758033802314e−18**
CSGO5	**0.006620471665250**	**0.006620471665250**	**0.006620471665250**	**3.528758033802314e−18**
CSGO6	**0.006620471665250**	**0.006620471665250**	**0.006620471665250**	**3.528758033802314e−18**
CSGO7	**0.006620471665250**	**0.006620471665250**	**0.006620471665250**	**3.528758033802314e−18**
CSGO8	**0.006620471665250**	**0.006620471665250**	**0.006620471665250**	**3.528758033802314e−18**
CSGO9	**0.006620471665250**	**0.006620471665250**	**0.006620471665250**	**3.528758033802314e−18**
CSGO10	0.006620471665250	0.006620471665250	0.006620471665250	3.528758033802314e−18

The best results of the experiments are represented in bold

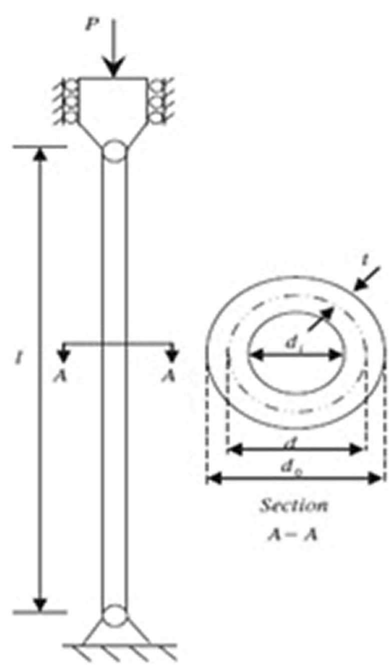

Fig. 6 The tabular column design

Table 15 Optimum results of tabular column design problem by different algorithms

Algorithms	x_1	x_2	Optimal result
Fuzzy PDCOE	5.4507	0.292	**25.5316*** **26.49912312**[C]
Rao	5.44	0.293	26.5323
CS	5.45139	0.29196	26.53217
CSGO1	5.451941	0.29174226	26.4913864882
CSGO2	5.451941	0.29174226	26.4913864882
CSGO3	5.451941	0.29174226	26.4913864882
CSGO4	5.451941	0.29174226	26.4913864884
CSGO5	5.451941	0.29174226	26.4913864882
CSGO6	5.451941	0.29174227	26.4913864882
CSGO7	5.451941	0.29174227	**26.4913864881**
CSGO8	5.451942	0.29174222	26.4913864884
CSGO9	5.451941	0.29174228	26.4913864882
CSGO10	5.451941	0.29174227	26.4913864882

The best results of the experiments are represented in bold

*Represents wrongly put

[c]Represents corrected value

as well as the best solution in terms of mean and standard deviation solution among others.

5.2.4.7 Tabular Column Design The objective of the tabular column design problem is to minimize the cost of designing a uniform column of the tabular section that includes material and construction costs as Fig. 6 [59]. The column is made of material of length (L) with a yield stress (S), a modulus of elasticity (E), and a density (D) carry a compressive load (P). This optimization problem has two optimized variables such as mean diameter of the column (x_1) and tube thickness (x_2), and six constraints such as the stress included in the column should be less then the buckling stress (($g_1(X)$), and the yield stress ($g_2(X)$), the mean diameter of the column is restricted between 2 and 14 cm ($g_3(X)$ and $g_4(X)$), and columns with thickness outside the range 0.2–0.8 cm is not commercially available ($g_5(X)$ and $g_6(X)$). The formulated optimization problem is referred from [59].

The optimization result of algorithms with constrains value is given in Table 15, and the statistical result is given in Table 16. The result of Rao, fuzzy proportional-derivative

Table 16 Statistical results of tabular column design problem by different algorithms

Algorithms	BEST	WORST	MEAN	SD
Fuzzy PDCOE	–	–	–	–
Rao	–	–	–	–
CS	26.53217	26.53972	26.53504	0.00193
CSGO1	26.491386488153360	26.491386495333025	26.491386490856410	2.363396596882364e−09
CSGO2	26.491386488165571	26.491386498136841	26.491386490233214	2.281681440373964e−09
CSGO3	26.491386488158895	26.491386497496833	26.491386490513413	2.511529340998990e−09
CSGO4	26.491386488390233	26.491386498026579	26.491386490660908	2.452381904843001e−09
CSGO5	26.491386488172985	26.491386501565771	26.491386490788571	3.436161611017448e−09
CSGO6	26.491386488156394	26.491386496288538	**26.491386489778495**	**1.843842257939578e−09**
CSGO7	**26.491386488145615**	26.491386509030100	26.491386492059771	4.395248054678416e−09
CSGO8	26.491386488359122	26.491386503520090	26.491386492538890	4.499242203936305e−09
CSGO9	26.491386488150628	26.491386497031133	26.491386490765663	2.795759587125070e−09
CSGO10	26.491386488156930	26.491386507603700	26.491386491513428	4.242503751531444e−09

The best results of the experiments are represented in bold

controller optimization engine (fuzzy PDCOE) and CS are reported from [60, 61] and [59], respectively. The results for the CSGO family are achieved by us. The "−" represents "not given" for that particular algorithm. The best result is highlighted in bold. From Tables 15 and 16, it can be stated that the CSGO family of algorithms outperforms all other algorithms. Among the CSGO family of algorithms, CSGO7 finds the best optimal solution, whereas CSGO6 finds the best solution in terms of mean and standard deviation among others.

5.2.4.8 Piston Lever Design Problem The objective is to minimize the oil volume when the lever of the piston is lifted up from 0^o to 45^o as shown in Fig. 7 [59]. This problem has four optimization variables and four constraints such as force equilibrium, maximum bending moment of the lever,

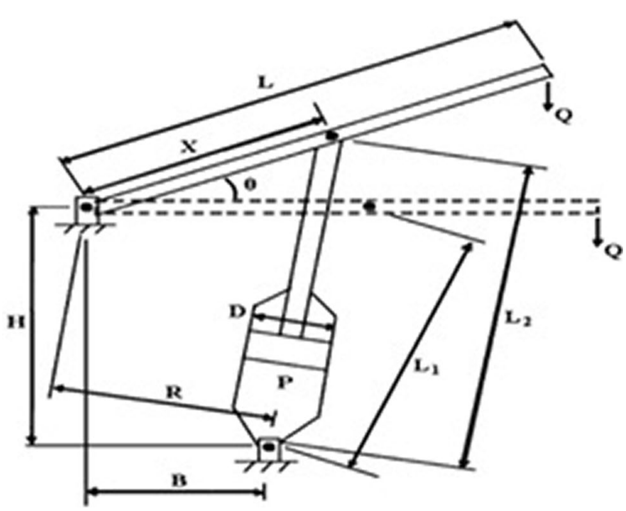

Fig. 7 Piston problem

minimum piston stroke, and geographical conditions. The formulated optimization problem is referred from [59].

The optimization result of algorithms is given in Table 17, and the statistical result is given in Table 18. The result of PSO, DE, GA, HPSO, HPSO with Q-learning, and CS are reported from [59]. The results for the CSGO family of algorithms are achieved by us. "−" represents "not given" for that particular algorithm. The best result is highlighted in bold. From Tables 17 and 18, it can be stated that the CSGO1 algorithm finds the best optimal solution as well as worst solution and CS algorithm achieve best mean solution among others.

5.2.4.9 Multi-plate Disc Clutch Brake Design Problem The objective is to minimize the total weight of the multi-plate disc clutch brake as Fig. 8 [61]. This problem has five optimized variables as driving force F(x_4), inner redius ri(x_1), outer redius ro(x_2), friction surface number Z(x_5), and disc thickness t(x_3). Since the problem contains eight different constraints, the feasible region in the solution space only accounts for 70%, which makes it more difficult to solve the problem. The formulated optimization problem is referred from [61].

The optimization results of algorithms are given in Table 19, and the statistical results are given in Table 20. The result of HHO, PVS, WCA, TLBO, and WSOA are reported from [38, 62–65], respectively. Similarly, the result of hHHO–SCA, NSGA-II, and AMDE are reported from [66] and for teaching learning-based pathfinder algorithm (TLPFA), the result is reported from [67]. The results for the CSGO family are achieved by us. "−" represents "not given" for that particular algorithm. The best result is highlighted in bold. From Tables 19 and 20, it can be stated that the TLPFA, CSGO1-CSGO6, CSGO9 and CSGO10 algorithms

Table 17 Optimum results of piston lever design problem by different algorithms

Algorithms	x_1	x_2	x_3	x_4
CS	0.0500	2.0430	120	4.0851
PSO	–	–	–	–
DE	–	–	–	–
GA	–	–	–	–
HPSO	–	–	–	–
HPSO with Q-learning	–	–	–	–
CSGO1	**0.0544705199829**	**1.9727488561405**	**119.9999999997065**	**3.2093948910144**
CSGO2	1000	1000	59.999999999356	1.273651425739
CSGO3	1000	1000	59.999999999356	1.273651425739
CSGO4	1000	1000	59.999999999352	1.273651425739
CSGO5	1000	1000	59.999999999352	1.273651425739
CSGO6	1000	1000	59.999999999352	1.273651425739
CSGO7	1000	1000	59.999999999352	1.273651425739
CSGO8	1000	1000	59.999999999352	1.273651425739
CSGO9	1000	1000	59.999999999356	1.273651425739
CSGO10	1000	1000	59.999999999356	1.273651425739

The best results of the experiments are represented in bold

Table 18 Statistical results of piston lever design problem by different algorithms

Algorithms	BEST	WORST	MEAN	SD
CS	8.4271	168.5920	**40.2319**	59.0552
PSO	122	294	166	51.7
DE	159	199	187	14.2
GA	161	216	185	18.2
HPSO	162	197	187	13.4
HPSO with Q-learning	129	168	151	13.4
CSGO1	**8.049336450844930**	**72.337509930474823**	70.599881973007484	10.568892070914794
CSGO2	72.337390896854473	72.337390896854686	72.337390896854629	7.741981830734082e−14
CSGO3	72.337390896854515	72.337390896854828	72.337390896854629	7.819310489593108e−14
CSGO4	72.337390896854473	72.337390896854686	72.337390896854615	8.700334948321627e−14
CSGO5	72.337390896854473	72.337390896854671	72.337390896854629	8.972224205222919e−14
CSGO6	72.337390896854487	72.337390896854771	72.337390896854629	8.787937783981305e−14
CSGO7	72.337390896854473	72.337390896854714	72.337390896854629	7.982368035972391e−14
CSGO8	72.337390896854473	72.337390896854728	72.337390896854629	8.064819254916418e−14
CSGO9	72.337390896854475	72.337390896854679	72.337390896854629	7.972224205222919e−14
CSGO10	72.337390896854477	72.337390896854777	72.337390896854629	8.797937783981305e−14

The best results of the experiments are represented in bold

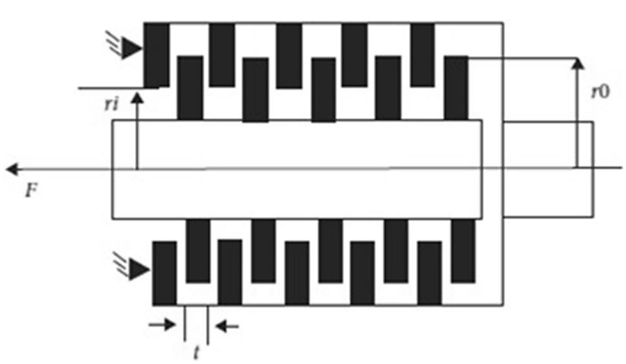

Fig. 8 Multi-plate disc clutch brake problem

find the best optimal solution than all other algorithms. Here, CSGO1 and CSGO4 algorithm outperforms all other algorithms and finds the best optimal solution in terms of best (BEST), mean (MEAN), worst (WORST) and standard deviation (SD) solution among others.

5.2.4.10 Corrugated Bulkhead Design Problem The objective is to minimize the weight of the corrugated bulkhead for a tanker [68]. This problem has four optimized variables such as width (x_1), depth (x_2), length (x_3), and plate thickness (x_4) and six constraints. The formulated optimization problem is referred from [67].

Table 19 Optimum results of multi-plate disc clutch brake design by different algorithms

Algorithms	x_1	x_2	x_3	x_4	x_5	Optimal result
HHO	69.999999999249	90	1	1000	3	**0.2597689930*** **0.313656610544706**[C]
PVS	70	90	1	980	3	0.31366
WCA	70	90	1	910	3	0.313656
TLBO	70	90	1	810	3	0.313656
WSOA	69.9996	90	1	600	2	0.23525
hHHO-SCA	70	90	2.312785	1000	1.5	0.389653842
NSGA-II	70	90	3	1000	1.5	0.4704
AMDE	70	90	3	810	1	0.3136566
TLPFA	69.9999	90.0002	1	680.0484	2	**0.23524**
CSGO1	70.0000070710678	90	1	270.7039721371302	2	**0.235242385126378**
CSGO2	70.0000070710678	90	1	167.9137205406554	2	**0.235242385126378**
CSGO3	70.0000070710678	90	1	61.266402700497608	2	**0.235242385126378**
CSGO4	70.0000070710678	90	1	350.5520670511875	2	**0.235242385126378**
CSGO5	70.0000070710678	90	1	564.7464174658709	2	**0.235242385126378**
CSGO6	70.0000070710678	90	1	745.7030521963353	2	**0.235242385126378**
CSGO7	60	90	1	72.498081261659237	2	0.330809706423005
CSGO8	60	90	1	329.7724327669219	2	0.330809706423005
CSGO9	70.0000070710678	90	1	968.9001314507637	2	**0.235242385126378**
CSGO10	70.0000070710678	90	1	571.8343784399032	2	**0.235242385126378**

The best results of the experiments are represented in bold

*Represents wrongly put

[c]Represents corrected value

Table 20 Statistical results of multi-plate disc clutch brake design by different algorithms

Algorithms	BEST	WORSE	MEAN	STD
HHO	–	–	–	–
PVS	–	–	–	–
WCA	–	–	–	–
TLBO	–	–	–	–
WSOA	–	–	–	–
hHHO—SCA	–	–	–	–
NSGA-II	–	–	–	–
AMDE	–	–	–	–
TLPFA	–	–	–	–
CSGO1	**0.235242385126378**	**0.235242385126378**	**0.235242385126378**	**5.646012854083702e−17**
CSGO2	0.235242385126378	0.330809706423005	0.327624129046451	0.017448125878169
CSGO3	0.235242385126378	0.330809706423005	0.324438551669897	0.024246206537976
CSGO4	**0.235242385126378**	**0.235242385126378**	**0.235242385126378**	**5.646012854083702e−17**
CSGO5	0.235242385126378	0.330809706423005	0.327624129046451	0.017448125878169
CSGO6	0.235242385126378	0.330809706423005	0.321252974293342	0.029160320719521
CSGO7	0.330809706423005	0.330809706423005	0.330809706423005	0.330809706423005
CSGO8	0.330809706423005	0.330809706423005	0.330809706423005	0
CSGO9	0.235242385126378	0.330809706423005	0.324438551669897	0.024246206537976
CSGO10	0.235242385126378	0.330809706423005	0.327624129046451	0.017448125878169

The best results of the experiments are represented in bold

Table 21 Optimum results of corrugated bulkhead design problem by different algorithms

Algorithms	x_1	x_2	x_3	x_4	Optimal result
CSGO1	57.644819523344992	34.150109951178024	57.644939799015262	1.049257917114013	6.839596279899362
CSGO2	57.644820772896942	34.150109887652107	57.644939345910650	1.049257917252038	6.839596279826666
CSGO3	57.644819962796369	34.150109901928758	57.644939421910124	1.049257916614736	6.839596279845521
CSGO4	57.644820464996570	34.150109692435414	57.644937754918580	1.049257912968599	6.839596279822398
CSGO5	57.644818850940318	34.150109536692320	57.644936416305917	1.049257907999835	6.839596279986681
CSGO6	57.644820863171816	34.150109774692190	57.644938438225104	1.049257915078914	6.839596279813460
CSGO7	57.644820566162814	34.150109615000055	57.644937135495880	1.049257911534023	6.839596279859407
CSGO8	57.644819703038927	34.150109750974714	57.644938190258316	1.049257913273896	6.839596279825124
CSGO9	57.644819026305839	34.150109700474118	57.644937749466969	1.049257911489912	6.839596279881082
CSGO10	57.644819123092731	34.150110099522784	57.644940980144021	1.049257919679909	6.839596280080818

Table 22 Statistical results corrugated bulkhead design problem by different algorithms

Algorithms	BEST	WORST	MEAN	SD
CSGO1	6.839596279899362	6.839596289558670	6.839596282867553	2.530597150337215e−09
CSGO2	6.839596279826666	6.839596287568882	6.839596282290350	2.586440627141579e−09
CSGO3	6.839596279845521	6.839596288078614	6.839596282066320	1.879211473605989e−09
CSGO4	6.839596279822398	6.839596289306188	6.839596282008946	2.722561976382350e−09
CSGO5	6.839596279986681	6.839596285057616	6.839596282035327	1.313951699882326e−09
CSGO6	6.839596279813460	6.839596290113129	6.839596282042369	2.268237453036341e−09
CSGO7	6.839596279859407	6.839596286421157	6.839596281907713	1.803297997945493e−09
CSGO8	6.839596279825124	6.839596292945716	6.839596292945716	3.839693220809969e−09
CSGO9	6.839596279881082	6.839596286671987	6.839596281940447	1.887799099707682e−09
CSGO10	6.839596280080818	6.839596288903122	6.839596282445204	2.080643690725033e−09

The optimization result of CSGO family algorithms is given in Table 21 and the statistical result is given in Table 22. For this problem, Ravindran et al. [69] reported the minimum value of 6.84241 using the random search method. A comparison of the results clearly shows that the CSGO family notably improves the results were obtained by the random search method.

In the present study, the CSGO family of algorithms performance is compared against other state-of-art as well as latest metaheuristic algorithms in solving structural optimization problems. The extensive comparative study conducted reveals that the CSGO family performs superior to different existing algorithms. This is partly due to the fact that there is one parameter, i.e., self-introspection parameter is replaced by the chaotic maps than in other algorithms. Other algorithms such as GA, DE, PSO, MBA, and CS require the tuning of at least one specific algorithm parameter. While simpler and more robust than competing algorithms, the CSGO family is able to resolve a wide variety of problems. Moreover, it avoids the risk of compromised performance due to proper parameter tuning.

6 Conclusion

In this paper, the family of CSGO algorithms is proposed as an improved version of the SGO algorithm to solve optimization problems. The family of ten CSGO algorithms is designed using ten chaotic maps in place of self-introspection parameters which improves the performance of the SGO algorithm. The performance of the CSGO family of algorithms is validated through 13 benchmark functions and to evaluate effectiveness, extensive experiments are conducted using 9 structural engineering problems and results are compared with many popular optimization algorithms. The extensive experiment and the promising results indicate the superiority of the proposed CSGO family for providing acceptable results in a wide range of problems. Again, although not yet mathematically proven, these experimental studies have shown that using chaotic maps is generally more useful than a constant value of 0.2 for the self-introspection parameter in SGO. Since in CSGO, any of the random numbers are not replaced by a chaotic map, then replacing random numbers with a chaotic map in SGO is further research. Furthermore, the mapping strategies can also be used to generate a high-quality initial population for obtaining rapid and better solutions. Proposing different hybridizing methods for chaotic mapping, and solving the problem of clustering or classification, large-scale optimizations and multiobjective optimizations using these chaotic concepts with SGO can be further researched.

Data Availability Statement Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Conflict of Interest The author declares that she has no conflict of interest in the publication of this paper.

References

1. Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 1*(1), 67–82.
2. Talbi, E. G. (2009). *Metaheuristics: From design to implementation*. Wiley.
3. Satapathy, S., & Naik, A. (2016). Social group optimization (SGO): A new population evolutionary optimization technique. *Complex & Intelligent Systems, 2*(3), 173–203.
4. Naik, A., Satapathy, S. C., Ashour, A. S., & Dey, N. (2018). Social group optimization for global optimization of multimodal functions and data clustering problems. *Neural Computing and Applications, 30*(1), 271–287.
5. Naik, A., & Satapathy, S. C. (2021). A comparative study of social group optimization with a few recent optimization algorithms. *Complex & Intelligent Systems, 7*(1), 249–295.
6. Eiben, A. E., & Schippers, C. A. (1998). On evolutionary diversification and intensification. *Fundamenta Informaticae, 35*(1/4), 35–50.
7. Gandomi, A. H., Yun, G. J., Yang, X. S., & Talatahari, S. (2013). Chaos-enhanced accelerated particle swarm optimization. *Communications in Nonlinear Science and Numerical Simulation, 18*(2), 327–340.
8. Mirjalili, S., & Gandomi, A. H. (2017). Chaotic gravitational constants for the gravitational search algorithm. *Applied Soft Computing, 53*, 407–419.
9. Rather, S. A., & Bala, P. S. (2020). Swarm-based chaotic gravitational search algorithm for solving mechanical engineering design problems. *World Journal of Engineering., 17*, 97–114.
10. Alatas, B. (2010). Chaotic bee colony algorithms for global numerical optimization. *Expert Systems with Applications, 37*(8), 5682–5687.
11. Alatas, B. (2010). Chaotic harmony search algorithms. *Applied Mathematics and Computation, 216*(9), 2687–2699.
12. Alatas, B., Akin, E., & Ozer, A. B. (2009). Chaos embedded particle swarm optimization algorithms. *Chaos, Solitons & Fractals, 40*(4), 1715–1734.
13. Coello, C. A. C., & Montes, E. M. (2002). Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics, 16*(3), 193–203.
14. Guo, H., Li, Y., Liu, X., Li, Y., & Sun, H. (2016). An enhanced self-adaptive differential evolution based on simulated annealing for rule extraction and its application in recognizing oil reservoir. *Applied Intelligence, 44*(2), 414–436.
15. Mingjun, J., & Huanwen, T. (2004). Application of chaos in simulated annealing. *Chaos, Solitons & Fractals, 21*(4), 933–941.
16. Gandomi, A. H., Yang, X. S., Talatahari, S., & Alavi, A. H. (2013). Firefly algorithm with chaos. *Communications in Nonlinear Science and Numerical Simulation, 18*(1), 89–98.
17. Wang, G. G., Guo, L., Gandomi, A. H., Hao, G. S., & Wang, H. (2014). Chaotic krill herd algorithm. *Information Sciences, 274*, 17–34.
18. Talatahari, S., Azar, B. F., Sheikholeslami, R., & Gandomi, A. H. (2012). Imperialist competitive algorithm combined with chaos for global optimization. *Communications in Nonlinear Science and Numerical Simulation, 17*(3), 1312–1319.
19. Saremi, S., Mirjalili, S., & Lewis, A. (2014). Biogeography-based optimisation with chaos. *Neural Computing and Applications, 25*(5), 1077–1097.
20. Gandomi, A. H., & Yang, X. S. (2014). Chaotic bat algorithm. *Journal of Computational Science, 5*(2), 224–232.
21. Varol Altay, E., & Alatas, B. (2020). Bird swarm algorithms with chaotic mapping. *Artificial Intelligence Review, 53*(2), 1373–1414.
22. Bingol, H., & Alatas, B. (2016). Chaotic league championship algorithms. *Arabian Journal for Science and Engineering, 41*(12), 5123–5147.
23. Gharehchopogh, F. S., Nadimi-Shahraki, M. H., Barshandeh, S., Abdollahzadeh, B., & Zamani, H. (2022). Cqffa: A chaotic quasi-oppositional farmland fertility algorithm for solving engineering optimization problems. *Journal of Bionic Engineering, 20*, 1–26.
24. Naik, A., Satapathy, S. C., & Abraham, A. (2020). Modified Social Group Optimization—A meta-heuristic algorithm to solve short-term hydrothermal scheduling. *Applied Soft Computing, 95*, 106524.
25. He, D., He, C., Jiang, L. G., Zhu, H. W., & Hu, G. R. (2001). Chaotic characteristics of a one-dimensional iterative map with infinite collapses. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 48*(7), 900–906.
26. Tavazoei, M. S., & Haeri, M. (2007). Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms. *Applied Mathematics and Computation, 187*(2), 1076–1085.
27. Singh, A. K., Kumar, A., Mahmud, M., Kaiser, M. S., & Kishore, A. (2021). COVID-19 infection detection from chest X-ray images using hybrid social group optimization and support vector classifier. *Cognitive Computation*. https://doi.org/10.1007/s12559-021-09848-3
28. Dey, N., Rajinikanth, V., Fong, S. J., Kaiser, M. S., & Mahmud, M. (2020). Social group optimization–assisted Kapur's entropy and morphological segmentation for automated detection of COVID-19 infection from computed tomography images. *Cognitive Computation, 12*(5), 1011–1023.
29. Das, S., Saha, P., Satapathy, S. C., & Jena, J. J. (2021). Social group optimization algorithm for civil engineering structural health monitoring. *Engineering Optimization, 53*(10), 1651–1670.
30. Kalananda, V. K. R. A., & Komanapalli, V. L. N. (2021). A combinatorial social group whale optimization algorithm for numerical and engineering optimization problems. *Applied Soft Computing, 99*, 106903.
31. Swathi, A. V. S., Chakravarthy, V. V. S. S. S., & Krishna, M. V. (2021). Circular antenna array optimization using modified social group optimization algorithm. *Soft Computing, 25*(15), 10467–10475.
32. Akhtar, M., Manna, A. K., & Bhunia, A. K. (2023). Optimization of a non-instantaneous deteriorating inventory problem with time and price dependent demand over finite time horizon via

hybrid DESGO algorithm. *Expert Systems with Applications, 211*, 118676.

33. Naik, A., & Chokkalingam, P. (2011). Binary social group optimization algorithm for solving 0–1 knapsack problem. *Decision Science Letters, 11*(1), 55–72.
34. Jena, J. J., & Satapathy, S. C. (2021). A new adaptive tuned Social Group Optimization (SGO) algorithm with sigmoid-adaptive inertia weight for solving engineering design problems. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-021-11266-4
35. Yu, J., Kim, C. H., Wadood, A., Khurshiad, T., & Rhee, S. B. (2018). A novel multi-population based chaotic JAYA algorithm with application in solving economic load dispatch problems. *Energies, 11*(8), 1946.
36. Tian, D., & Shi, Z. (2018). MPSO: Modified particle swarm optimization and its applications. *Swarm and Evolutionary Computation, 41*, 49–68.
37. Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95*, 51–67.
38. Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems, 97*, 849–872.
39. Nematollahi, A. F., Rahiminejad, A., & Vahidi, B. (2017). A novel physical based meta-heuristic optimization method known as lightning attachment procedure optimization. *Applied Soft Computing, 59*, 596–621.
40. Nematollahi, A. F., Rahiminejad, A., & Vahidi, B. (2020). A novel meta-heuristic optimization method based on golden ratio in nature. *Soft Computing, 24*(2), 1117–1151.
41. Moghdani, R., & Salimifard, K. (2018). Volleyball premier league algorithm. *Applied Soft Computing, 64*, 161–185.
42. Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences, 179*(13), 2232–2248.
43. Hashim, F. A., Houssein, E. H., Mabrouk, M. S., Al-Atabany, W., & Mirjalili, S. (2019). Henry gas solubility optimization: A novel physics-based algorithm. *Future Generation Computer Systems, 101*, 646–667.
44. Dhiman, G., & Kumar, V. (2019). Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. *Knowledge-Based Systems, 165*, 169–196.
45. Faramarzi, A., Heidarinejad, M., Mirjalili, S., & Gandomi, A. H. (2020). Marine predators algorithm: A nature-inspired metaheuristic. *Expert Systems with Applications, 152*, 113377.
46. Kaur, S., Awasthi, L. K., Sangal, A. L., & Dhiman, G. (2020). Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. *Engineering Applications of Artificial Intelligence, 90*, 103541.
47. Li, S., Chen, H., Wang, M., Heidari, A. A., & Mirjalili, S. (2020). Slime mould algorithm: A new method for stochastic optimization. *Future Generation Computer Systems, 111*, 300–323.
48. Dhiman, G., & Kaur, A. (2019). STOA: A bio-inspired based optimization algorithm for industrial engineering problems. *Engineering Applications of Artificial Intelligence, 82*, 148–174.
49. Meirelles, G., Brentan, B., Izquierdo, J., & Luvizotto, E., Jr. (2020). Grand tour algorithm: Novel swarm-based optimization for high-dimensional problems. *Processes, 8*(8), 980.
50. Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation, 1*(1), 3–18.
51. Koziel, S., & Michalewicz, Z. (1999). Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. *Evolutionary Computation, 7*(1), 19–44.
52. Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering, 186*(2–4), 311–338.
53. Kaveh, A., & Talatahari, S. (2009). A particle swarm ant colony optimization for truss structures with discrete variables. *Journal of Constructional Steel Research, 65*(8–9), 1558–1568.
54. Belegundu, A.D. (1983) Study of mathematical programming methods for structural optimization. *Diss Abstr Int Part B Sci Eng* 43.
55. Askari, Q., Younas, I., & Saeed, M. (2020). Political optimizer: A novel socio-inspired meta-heuristic for global optimization. *Knowledge-Based Systems, 195*, 105709.
56. Faramarzi, A., Heidarinejad, M., Stephens, B., & Mirjalili, S. (2020). Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Systems, 191*, 105190.
57. Moosavi, S. H. S., & Bardsiri, V. K. (2019). Poor and rich optimization algorithm: A new human-based and multi populations algorithm. *Engineering Applications of Artificial Intelligence, 86*, 165–181.
58. Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Computers & Structures, 169*, 1–12.
59. Gandomi, A. H., Yang, X. S., & Alavi, A. H. (2013). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with Computers, 29*(2), 245–245.
60. Rao, S. S. (2019). *Engineering optimization: Theory and practice*. Wiley.
61. Hsu, Y. L., & Liu, T. C. (2007). Developing a fuzzy proportional–derivative controller optimization engine for engineering design optimization problems. *Engineering Optimization, 39*(6), 679–700.
62. Che, Y., & He, D. (2021). A Hybrid whale optimization with seagull algorithm for global optimization problems. *Mathematical Problems in Engineering, 2021*, 1–31.
63. Savsani, P., & Savsani, V. (2016). Passing vehicle search (PVS): A novel metaheuristic algorithm. *Applied Mathematical Modelling, 40*(5–6), 3951–3978.
64. Zheng, Y. J. (2015). Water wave optimization: A new nature-inspired metaheuristic. *Computers & Operations Research, 55*, 1–11.
65. Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-aided Design, 43*(3), 303–315.
66. Kamboj, V. K., Nandi, A., Bhadoria, A., & Sehgal, S. (2020). An intensify Harris Hawks optimizer for numerical and engineering optimization problems. *Applied Soft Computing, 89*, 106018.
67. Tang, C., Zhou, Y., Tang, Z., & Luo, Q. (2021). Teaching-learning-based pathfinder algorithm for function and engineering optimization problems. *Applied Intelligence, 51*(7), 5040–5066.
68. Kvalie, D. (1967). *Optimization of plane elastic grillages*. Doctoral dissertation, PhD Thesis, Norges Teknisk Naturvitenskapelige Universitet, Norway.

69. Ravindran, A., Reklaitis, G. V., & Ragsdell, K. M. (2006). *Engineering optimization: methods and applications*. Wiley.