Knowledge-Based Systems 222 (2021) 106926

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

SSC: A hybrid nature-inspired meta-heuristic optimization algorithm for engineering applications

Gaurav Dhiman

Department of Computer Science, Government Bikram College of Commerce, Punjabi University, Patiala 147001, India

ARTICLE INFO

Article history:
Received 16 October 2020
Received in revised form 24 February 2021
Accepted 2 March 2021
Available online 25 March 2021

Keywords:
Chimp Optimization Algorithm (*choA*)
Spotted Hyena Optimizer (*SHO*)
Sine–cosine
Meta-heuristics
Optimization
Swarm-intelligence
Engineering design

ABSTRACT

Chimp Optimization Algorithm (*ChoA*) is a recently developed meta-heuristic approach which is inspired by the individual intelligence and sexual motivation of chimps. It is designed for trapping the local optima to alleviate the slow convergence speed. In this paper, a hybrid algorithm is developed which is based on the sine–cosine functions and attacking strategy of Spotted Hyena Optimizer (*SHO*). This hybrid algorithm is termed as Sine–cosine and Spotted Hyena-based Chimp Optimization Algorithm (*SSC*). This algorithm is used to find the best optimal solutions of real-life complex problems. The sine–cosine and attacking strategy of *SHO* algorithm is responsible for better exploration and exploitation. These strategies are applied to update the equations of chimps during the searching process to overcome the drawbacks of the *ChoA* algorithm such as slow convergence and local minima. Experimental results based on *IEEE CEC'17* and six real-life engineering problems such as welded beam design, tension/compression spring design, pressure vessel design, multiple disk clutch brake design, gear train design, and car side crashworthiness, demonstrate the robustness, effectiveness, efficiency, and convergence analysis of the proposed *SSC* algorithm in comparison with other competitor approaches.
Note that the source codes are available at http://www.dhimangaurav.com.
© 2021 Elsevier B.V. All rights reserved.

1. Introduction

Optimization has been a burning issue in every area of science over the last few decades. Researchers are designing best optimization algorithms for solving complex engineering, mathematics, medical fields, statistics, computer science, and more problems [1–7]. The optimization is commonly defined as a method to select the ideal optimal solutions for a particular problem/function, globally or almost globally. Majority of the real-world problems can be perceived as optimization issues [8–14]. The market in each region faces a range of complex problems related to real-world optimization every day [15]. Frankly speaking, high-performance algorithms are expected to meet potential complicated problems. Hence, for such issues the researchers are attempting to make highly successful optimization techniques. So these algorithms satisfy future complex problem specifications. Overall, the optimization approach has been used to find the most accurate and best solution to deterministic optimization problems that suffer from a variety of drawbacks or major issues such as the need to derive search space, unbalanced exploitation or discovery, local optima, premature convergence, and sluggish convergence [9,16–26].

Due to these disadvantages, the researcher's interest in the different fields has increased in the last few years in bio-inspired techniques. Sometimes, mostly real-life applications in diverse fields/domain, such as digital filtering, bio-medical, statistics, pattern recognition, tuning machine-learning parameters, image processing, engineering, clustering, applied mathematics, and computer science, etc., are severely linear, non-linear, non-continuous, with several design variables included and are having complex constraints [27–31]. In addition, the best and feasible solution in the search area is not guaranteed and these complicated functions can often not be solved [32]. Such are the real-world challenges that motivate scientists to create a highly stable and effective algorithm of optimization to find their solutions. In this respect, various robust optimization algorithms are disadvantaged when looking at solutions to complex features. and therefore, bio-inspired algorithms have been considered the most widely used in recent times to solve problems of optimization. For techniques like various designs, merge/modify two or more approaches while increasing their limitations. Numerous algorithms have worked to show improved convergence efficiency without alteration or hybridization to implement complicated function and improve effectiveness. Moreover, according to No Free Lunch (*NFL*) theorem [33], there is no any optimization algorithm which can solve all the real-life complex problems. So these facts are the

E-mail address: gdhiman0001@gmail.com.

https://doi.org/10.1016/j.knosys.2021.106926
0950-7051/© 2021 Elsevier B.V. All rights reserved.

motivation of this research to design a novel hybrid optimization algorithm.

Chimp Optimization Algorithm (*ChoA*) is the newly developed bio-inspired optimization algorithm [34]. Chimp optimization algorithm (*ChoA*) has contented the convergence results for proper equilibrium between exploitation and exploration. But during the search process this version cannot solve some of the challenging problems. This algorithm also faces many limitations such as slow convergence rate, position of local minima rather than global minima, premature convergence, low equilibrium among exploitation and exploration, etc. Considering the above to mask its shortcomings, the *ChoA* algorithm must be improved. Thus, during this study, the proposed approach is implemented from its original Chimp Optimization Algorithm by modifying and refining the attacking strategy of Spotted Hyena Optimizer and different mathematical equations via sine–cosine functions(*SHO*) [35], which are capable of improving the efficiency of the optimum outputs, improving the iteration rate, and is also able to balance the original *ChoA* algorithm's exploitation and exploration phases. This principle is useful in optimization, as they help produce accurate solutions. The use of these functions in optimization methods allows to broaden the range of solutions available, avoiding the use of local solutions.

This research proposes an updated version of Chimp Optimizer that incorporates sine–cosine function and attacking feature of *SHO* algorithm with the original *ChoA* algorithm. This hybrid approach is termed as Sine–cosine and Spotted Hyena Optimizer based Chimp Optimization Algorithm (*SSC*). The aim is to refine the original version of *ChoA's* search system in order to find optimal solutions.

1. With the shortcomings of the original *ChoA* algorithm, Sine–cosine functions and attacking strategy of *SHO* algorithm are applied to improve its convergence efficiency.
2. *SSC* is designed to solve complex single-objective optimization problems.
3. The dominance of the proposed algorithm in solving single objective complex optimization issues are indicated by experimental solutions and convergence graphs.

Different comparisons and tests are made between different well-known algorithms to pick the one that provides the most reliable solutions. It is noted that the *SSC* algorithm trapped the best optima solutions for numerical efficiency, statistical measurements and convergence curves that were verified with the competitor methods.

The rest of the paper is structured as follows: Section 2 presents the literature review. Section 3 presents the proposed *SSC* algorithm and analyzes the experimental findings. Section 4 validates the proposed algorithm on engineering problems. Finally, Section 5 accounts for this paper's conclusion and potential work.

2. Literature review

To meet the demand for complex issues in various fields the researchers create a robust modified or hybrid, nature-inspired methodologies, such as Genetic Algorithm (*GA*) [36], Particle Swarm Optimization (*PSO*) [37], and several others, [38]. In addition, a newly updated and modified version has been reported.

A new way of dealing with premature convergence in *SCA* has been suggested by Zou et al. [39] and called as Chaos Cultural Sine–Cosine Algorithm (*CCSCA*). The approach proposed used a cultural algorithm as population structure and used chaotic methods for search management. The results of the benchmarking functions showed that in terms of global search functionality the *CCSCA* obtained better results than the basic *SCA* algorithm.

In order to address the profit based unit commitment (*PBUC*) problem, Reddy et al. [40] contemplated a binary *SCA* approach. This approach used modified sigmoid features to map continuous search space. The findings showed how efficient the proposed binary *SCA* is compared to alternative methods for *PBUC* problem.

Gholizadeh and Sojoudizadeh [41] suggested the discrete version of *SCA* to solve the complex discrete problems with optimization. Multiple benchmark functions illustrate the efficiency of the proposed discrete *SCA*. Compared to other related approaches, the proposed algorithm obtained better results.

In order to solve optimization problems, Pasandideh and Khalilpourazari [42] contemplated the latest hybrid algorithm based on *SCA* approach. The combination of ideas and operators of each algorithm enables a strong exchange of exploitation and exploitation skills between the proposed hybrid method. Moreover, to measure the efficiency seven benchmark functions are used. The results showed that, relative to other similar approaches, the proposed hybrid method would provide a highly competitive solution.

El-Shorbagy et al. [43] suggested a new hybrid algorithm to solve engineering design problems, called the *SSGA*, based on hybridization of *SCA* using a steady state genetic algorithm. This incorporates the advantages of *SCA* discovery and genetic algorithm's ability to prevent early convergence. The findings portrays that the proposed *SSGA* approach enhanced the performance of two complex engineering design problems in comparison to other comparable optimization approaches.

Guo et al. [44] suggested a new updated method of learning with *SCA* to initialize people to solve many problems of optimization. In addition, the plan is coupled with an almost opposition learning strategy and opposition-based learning methods to increase the efficiency and convergence rate of global exploration. The proposed method efficiency was evaluated using 23 benchmark functions. The findings show that, as compared to alternative approaches, the proposed algorithm greatly overwhelms others.

In order to address the optimization problems, Liu et al. [45] projected a new improved technique using *SCA* called *HSCA*. The findings reveal that in multiple cases the proposed *HSCA* has improved results and can yield scheduling results with higher quality than other related approaches

The *SSA* and sine–cosine (*SCA*) algorithms combine the methods for enhanced *MOPS* techniques (*MOSSASCA*) to locate an acceptable solution for the problem of the virtual position of machines [46]. The *MOSSASCA* proposed is intended primarily to alleviate service quality infringements, to minimize power consumption and rise the total time before terminating the agent and to eliminate the conflicts between three objectives. A local search technique is used at *SCA* to increase SSA efficiency so that convergence speeds are increased and an optimal solution is not stuck locally. There were various physical and digital devices in a series of demonstration to assess the efficiency of the combined algorithm. Well-known *MOP* methods have been contrasted with the results of *MOSSASCA*. The findings suggest a equilibrium between the achievement of the three objectives.

Qu et al. [47] proposed a linear weight loss method to help match the global search and the algorithm's exponential decrease conversion parameter as well as the local capacity for growth. Based on simulation performance, current approaches will show that they have faster convergence speed and can efficiently prevent declining into local optimum and greater optimization precision.

Dhiman and Kaur [13] proposed the hybrid optimization algorithm named as Hybrid Particle Swarm and Spotted Hyena Optimizer (*HPSSHO*). Four meta-heuristic algorithms are compared to the proposed algorithm. The results of the experiments show that the algorithm is better than other meta-heuristic algorithms.

Luo et al. [48] proposed the lateral inhibition-based (*LI*) spotted hyena optimizer (*SHO*) to address complicated image matching issues. The lateral mechanism of inhibition is employed for image pre-processing, which improves the intensity gradient of the picture contrast and improves image characters to increase image matching accuracy. The proposed *LI-SHO* image matching approach combined the benefits of *SHO* to the mechanism of lateral inhibition. The test shows that the proposed lateral inhibition algorithm suits the other comparative algorithm more efficaciously and feasible.

Moayedi et al. [49] developed the *SHO* and ant lion optimization (*ALO*) method to simulate the strength of soil shear. These algorithms have been used to counteract the measuring disadvantages of the artificial neural network (*ANN*) model. The best fitting structures have been defined through a method of test and error following the creation of the *ALO–ANN* and *SHO–ANN*. The results showed the efficiency of both algorithms because *ALO* and *SHO* respectively reduced the ANN's prediction error by about 35% and 18%.

Jia et al. [50] proposed the feature selection based *SHO* method. The experiments reveal that the proposed algorithm outperformed than the others and proved the better performance in spatial search and feature selection.

3. Proposed algorithm

3.1. Short definitions

3.1.1. Sine–Cosine algorithm

This algorithm was originally designed and developed by Mirjalili [51]. In *SCA* algorithm, the positions are updated by using the following equations:

$$SX_i^{k+1} = SX_i^k + R1 \times sin(R2) \times |R3 \cdot PX_i^k - SX_i^t| \tag{1}$$

$$SX_i^{k+1} = SX_i^k + R1 \times cos(R2) \times |R3 \cdot PX_i^k - SX_i^t| \tag{2}$$

where SX_i^t is the current position at kth iteration with PX_i population. $R1, R2, R3$ are random variables.

- The *SCA* SCA generates and improves a random set of solutions for a given problem, resulting in higher exploration and local optimal avoidance in comparison to other single-solution algorithms.
- Various areas of the search field are searched When the sine and cosine functions return values greater than 1 or smaller than -1.
- The promising areas of the search field are used if sine and cosine return values between -1 and 1.
- The *SCA* algorithm moves smoothly from discovery to exploitation by modifying the sine and cosine functions using adaptive range.
- The best path to the global optimum is saved in a number of variables as the destination point and is never lost during optimization.
- Since solutions are often updated to the best solution so far, there is a trend towards optimizing the best areas of the search space.
- As the proposed algorithm regards optimization problems as black-boxes, problems in various fields that can be properly formulated are readily incorporated.

The pseudo-code of *SCA* algorithm is described in Algorithm 1.

Algorithm 1 Pseudo-code of SCA algorithm

```
Inputs: Population size
Population initialization
Set k ← 0
while k < Max_Itr do
    for each search agent do
        Update search agent by using sine–cosine functions
    end for
    Update the position of the current search agent
    k ← k + 1
end while
```

3.1.2. Spotted Hyena Optimizer (SHO)

Spotted Hyena Optimizer (*SHO*) [35] is focused on spotted hyenas' social activity and communal relationships. It primarily focuses on the trusted group of spotted hyenas' attacking, encircling, hunting, and searching activity. *SHO* algorithm effectively mimicked spotted hyena behavior to find the best possible optimal solution.

Encircling behavior in *SHO* is apprehended by subsequent series of equations [35]:

$$\vec{X}_h = | \vec{A} \cdot \vec{C}_p(x) - \vec{C}(x) | \tag{3}$$

$$\vec{C}(x+1) = \vec{C}_p(x) - \vec{B} \cdot \vec{X}_h \tag{4}$$

where $\vec{X}_h$ is the distance a spotted hyena would travel to meet its prey. The running iteration at a given instance is indicated by x. $\vec{C}_p$ and $\vec{C}$, represents the location vectors for prey and spotted hyena, respectively. The multiplication vectors and absolute value are represented by the $\cdot$ and $||$ symbols, respectively. The coefficient vectors $\vec{A}$ and $\vec{B}$ are calculated as follows:

$$\vec{A} = 2 \cdot \vec{xd}_1 \tag{5}$$

$$\vec{B} = 2\vec{h} \cdot \vec{xd}_2 - \vec{h} \tag{6}$$

$$\vec{h} = 5 - \left(Itr \times \frac{5}{Max_{Itr}} \right) \tag{7}$$
$$\text{where}\ ,\ Itr = 0, 1, 2, \ldots, Max_{Itr}$$

Here, in each iteration the value of $\vec{h}$ is reduced from 5 to 0. The random vectors $\vec{xd}_1$ and $\vec{xd}_2$ have values in the range [0, 1]. Changing the place vectors $\vec{A}$ and $\vec{B}$ helps one to reach various locations.

Furthermore, using the equations below, the hunting activity of spotted hyenas is mapped, and possible hunting regions are identified:

$$\vec{X}_h = | \vec{A} \cdot \vec{C}_h - \vec{C}_k | \tag{8}$$

$$\vec{C}_k = \vec{C}_h - \vec{B} \cdot \vec{X}_h \tag{9}$$

$$\vec{O}_h = \vec{C}_k + \vec{C}_{k+1} + \cdots + \vec{C}_{k+N} \tag{10}$$

where N is the number of iterations, which can be calculated as follows:

$$N = count_{nos}(\vec{C}_h, \vec{C}_{h+1}, \vec{C}_{h+2}, \ldots, (\vec{C}_h + \vec{M})) \tag{11}$$

$$\vec{C}(x+1) = \frac{\vec{O}_h}{N} \tag{12}$$

Here, *nos* denotes the number of solutions that are at least very close to the optimally best solution for the search space under consideration. The cluster of optimal solutions is described by $\vec{O}_h$.

Within range [0.5, 1], $\vec{M}$ denotes randomly initialized vectors. The best optimal solution is identified using $\vec{C}(x+1)$ which further helps in updating the search agents.

The exploration is assured by using a vector $\vec{B}$ of random values. The search agent will shift away from prey if the randomly assigned values are > 1 or < 1. Exploration is also supported by $\vec{A}$, which further assigns random values in the range $[0, 5]$ that serves as prey weight.

The exploitation of *SHO* algorithm starts when $|\vec{B}| < 1$ and has a random initialization of $\vec{B}$, which is in the range $[-1, 1]$. In *SHO*, the optimization process begins with the generation of a population of random solutions. At first, search agents form clusters by defining the locations of the best search agents and further update their places. Each iteration linearly decrements the values of parameters h and E. After each iteration has been completed successfully, the best locations corresponding to search agents are fetched.

Algorithm 2 describes the pseudo-code of *SHO* algorithm.

Algorithm 2 Pseudo-code of SHO algorithm

```
Inputs: Population size
Population initialization
Set Itr ← 0
while Itr < Max_Itr do
    for each search agent do
        Update search agent by using group selection strategy
    end for
    Update the position of the current search agent
    Itr ← Itr + 1
end while
```

3.1.3. Chimp Optimization Algorithm (ChoA)

Khishe et al. [34] developed the bio-inspired optimization algorithm named the Chimp Optimization Algorithm (*ChoA*). This technique is motivated by chimpanzees' individual intellect and sexual inspiration in group hunting. It is distinct among other carnivores in society. Four different phases were used during this methodology to model different intelligence. The initial solution is hereby presumed to be better aware of the position of the target by the chaser, driver, attacker, and barrier. In the next stage four other still achieved optimum solutions are stored and the other chimpanzees are forced to update their own position to the best places of chimpanzees.

The mathematical model of the algorithm proposed was defined by as follows:

$$D_r = \left|c.a_{prey}\,(n_i) - ma_{chimp}\,(n_i)\right| \tag{13}$$

$$a_{chimp}\,(n_i+1) = a_{prey} - a.d \tag{14}$$

The total number of iterations are defined by n_i, and the coefficient vectors are: c, m, and a. These coefficients c, m, and a are calculated by the following equations:

$$a = 2.l.r_1 - l \tag{15}$$

$$c = 2.r_2 \tag{16}$$

$$m = chaotic_{value} \tag{17}$$

where r_1 and r_2 are random numbers in range $[0, 1]$, m is the chaotic vector, and during the iteration process l is linearly reduced from 2.5 to 0.

$$\begin{aligned} d_{attacker} &= |c_1 a_{attacker} - m_1.x| \\ d_{barrier} &= |c_2 a_{barrier} - m_2.x| \\ d_{chaser} &= |c_3 a_{chaser} - m_3.x| \\ d_{driver} &= |c_4 a_{driver} - m_4.x| \end{aligned} \tag{18}$$

When the random vectors are in the $[-1, 1]$ range, then the chimp's next position can be somewhere between where it is now and where the target or prey is.

$$\begin{aligned} x_1 &= a_{attacker} - a_1.d_{attacker} \\ x_2 &= a_{barrier} - a_2.d_{barrier} \\ x_3 &= a_{chaser} - a_3.d_{chaser} \\ x_4 &= a_{driver} - a_4.d_{driver} \end{aligned} \tag{19}$$

The following mathematical equation is used to change the chimps' location during the search:

$$x_{n_i+1} = \frac{x_1 + x_2 + x_3 + x_4}{4} \tag{20}$$

The following mathematical equation has been applied for updating chimps location during the search process in the search domain.

$$a_{chimp}\,(n_i+1) = \begin{cases} a_{prey}\,(n_i) - x.d, & if \quad \phi < 0.5 \\ chaotic_{value} & if \quad \phi > 0.5 \end{cases} \tag{21}$$

Algorithm 3 describes the pseudo-code of *ChoA* Algorithm.

Algorithm 3 Pseudo-code of ChoA algorithm

```
Inputs: Population size N and U ← 1
Population initialization X_i(i = 1, 2, ..., N)
while U < Max_iterations do
    for each chimps do
        Define the group of chimp's
        Update search agent by using group strategy
    end for
    for each search climb do
        if U < 1 then
            Update the position of the current search chimp
        else if U > 1 then
            Select the random search agent (i.e., chimp)
        end if
        Update the position of the current search chimp
    end for
    Update Attacker, Barrier, Driver and Chaser
    U ← U + 1
end while
```

3.2. Proposed algorithm (SSC)

The complex optimization application is a challenge for metaheuristic optimization. According to the literature, each optimization approach cannot show the best solution to any complex problem. All algorithms which have some inconvenience so they may not find a solution for complex functions because of these limitations. Due to the current competitive environment, therefore, we need the most efficient optimizing strategies such that complex problems can be easily addressed. However, these techniques might easily deal with complicated functions if the discovery and exploitation process of these approaches is too high. Here we try and present a newly updated algorithm with the help of the sine–cosine function and attacking strategy of *SHO* algorithm for the complex optimization functions after the initial step of chimp optimization algorithm. The algorithm allows the sine and cosine function to fluctuate or find optimal solutions. These functions help neglect local optimum and easily push the algorithm to trap global optimum. The *ChoA* algorithm, on the other hand, will solve a number of complex optimization functions, but it has some inconveniences in the search domain as it is captured locally, globally and explores optimum values. It cannot cope with huge difficult optimizations problems and cannot fix various inconveniences, such as premature convergence, slow

Table 1
Summary of IEEE CEC'17 benchmark functions.

Type	No.	Description	Fi*
Unimodal functions	1	Shifted and Rotated Bent Cigar Function	100
	2	Shifted and Rotated Sum of Different Power Function	200
	3	Shifted and Rotated Zakharov Function	300
Simple multimodal functions	4	Shifted and Rotated Rosenbrock's Function	400
	5	Shifted and Rotated Rastrigin's Function	500
	6	Shifted and Rotated Expanded Scaffer's Function	600
	7	Shifted and Rotated Lunacek Bi- Rastrigin Function	700
	8	Shifted and Rotated Non-Continuous Rastrigin's Function	800
	9	Shifted and Rotated Levy Function	900
	10	Shifted and Rotated Schwefel's Function	1000
Hybrid functions	11	Hybrid Function (N = 3)	1100
	12	Hybrid Function (N = 3)	1200
	13	Hybrid Function (N = 3)	1300
	14	Hybrid Function (N = 4)	1400
	15	Hybrid Function (N = 4)	1500
	16	Hybrid Function (N = 4)	1600
	17	Hybrid Function (N = 5)	1700
	18	Hybrid Function (N = 5)	1800
	19	Hybrid Function (N = 5)	1900
	20	Hybrid Function (N = 6)	2000
Composition Functions	21	Composition Function (N = 3)	2100
	22	Composition Function (N = 3)	2200
	23	Composition Function (N = 3)	2300
	24	Composition Function (N = 3)	2400
	25	Composition Function (N = 3)	2500
	26	Composition Function (N = 3)	2600
	27	Composition Function (N = 3)	2700
	28	Composition Function (N = 3)	2800
	29	Composition Function (N = 3)	2900

diversity, and slow convergence etc. This algorithm has been modified in the search domain in order to determine the precise and best global optimum for complex optimization functions. The chimp optimization algorithm (*ChoA*) phases works in the direction of discovery of the optimum solution and altered phases have been used to exploit optima in the search domain. The move has helped to rapidly achieve the best global performance and to neglect the local optima. Because of this we have improved by this modification the convergence rate of the regular *ChoA* algorithm.

In addition, the *SSC* algorithm's mathematical formulation is demonstrated with the following steps:

- Initialization: Firstly, at this point, we initialize the search agent population, which is randomly initialized by the given functions. Where the algorithms allocate a n dimensional random vector to ith chimp; X_i $(i = 1, 2, \ldots, n)$.
- Evaluation: In this step the population search participants are assessed by the supremacy of the search space in the search process. Then the fitness value of each search member is checked by the provided objective function, and during the search, each search member of the crowd uses these fitness values to position the new search positions.
- Exploration Phase: In this condition, chimpanzees are updating their location to seek their next nearest prey position. With the aid of this update, chimps may in at least a number of iterations easily be stuck in optima solutions in the complex space. It also helps to minimize computing time when looking for the next best search agent location.

$$r_2 = (2\pi) \times rand \tag{22}$$

$$\begin{aligned} x_1 &= a_{attacker} - cos(r_2) \times a_1.d_{attacker} \\ x_2 &= a_{barrier} - sin(r_2) \times a_2.d_{barrier} \\ x_3 &= a_{chaser} - cos(r_2) \times a_3.d_{chaser} \\ x_4 &= a_{driver} - sin(r_2) \times a_4.d_{driver} \end{aligned} \tag{23}$$

where r_2 is the random number lies between the range of [0, 1].

- Stopping conditions: Eventually, the stop conditions were introduced for searching in the search domain for the best possible optimal value. These conditions were used to determine the selection process of all community search agents and substitute them according to the roles of the best search agent. It is replicated again and again until it meets the preventive criterion, for example, until it hits the maximum possible iterations or the solution is found as soon as possible.

The pseudocode of *SSC* algorithm is described in Algorithm 4.

3.3. Computational complexity

The *SSC* initialization takes $\mathcal{O}(n \times d)$ time, where n is the number of iterations and d is the dimension of a test function used to change the solutions within the boundary. In the next step, The fitness measurement of each search agent takes $\mathcal{O}(Max_{iterations} \times n \times d)$ time, where $Max_{iterations}$ is the maximum number of iterations needed to simulate the proposed *SSC* algorithm. Thus, $\mathcal{O}(Max_{iterations} \times n \times d \times N)$ is the overall time complexity of *SSC* algorithm.

3.4. Parameter settings

The *SSC* algorithm and the competitive algorithms are tested with 30 independent runs, over ten runs of experiments for fair benchmark comparison. The maximum number of iterations number is 1000 for each benchmark test function. For comparison purpose, Seagull Optimization Algorithm (*SOA*) [52], Spotted Hyena Optimizer (*SHO*) [35], Tunicate Swarm Algorithm (*TSA*) [53], Emperor Penguin Optimizer (*EPO*) [15], Rat Swarm Optimization (*RSO*) [19], Emperor penguin and Salp swarm algorithm (*ESA*) [6], Differential Evolution (*DE*) [54], and Genetic Algorithm (*GA*) [55] approaches are used. The parameter values of these techniques are chosen according to the literature. Experimentation is verified on *MATLAB* R2019a version using *Microsoft*

Algorithm 4 Pseudocode of SSC algorithm

Inputs: Population size N and $U \leftarrow 1$
Initialize the population $x_i (i = 1, 2, \ldots, N)$
while $U < Max_{iterations}$ **do**
  for each chimps **do**
    Define the group of chimp's
    Update m, c, and l
  end for
  for each search climb **do**
    if $\phi < 0.5$ **then**
      if $|x| < 1$ **then**
        Update the position of the current search agent or chimp by Eq. (14)
      else if $|x| > 1$ **then**
        Select a random search chimp
      end if
    else if $\phi > 0.5$ **then**
      Update the position of the current search agent or chimp by the Eq. (21)
    end if
  end for
  Again update m, c, l, and a
  Apply attacking strategy of SHO algorithm for parameters m, c, l, and a
  Update attacker, chaser, barrier and driver
  $t \leftarrow t + 1$
end while
Return $x_{attacker}$

Windows 10 environment with 3.2GHZ *Intel* i5 processor. To validate the algorithm's efficiency, this analysis uses both qualitative and quantitative measures.

3.5. Experimentation on IEEE CEC'17 test suite

The *IEEE CEC'17* test suite [56] was selected as a test-bed in this analysis as it is complicated and customized for global optimization. The *IEEE CEC'17* test suite contained 30 challenging functions [56] (see Table 1). The *IEEE CEC'17* test suite functions are classified into four categories, i.e., unimodal functions, multimodal functions, composition functions, and hybrid functions. Also, the mathematical formulation and properties of the *IEEE CEC'17* test suite are available in [56]. Table 2 shows the obtained optimal results by contemplated and comparative techniques. It can be seen from this Table that the efficiency of *SSC* is superior to others in terms of average and standard deviation on most of the benchmark test functions, as seen in this Table.that on most of the benchmark test functions.

3.6. Convergence behavior analysis

For a selected set of test functions, Fig. 1 shows the convergence curves of proposed *SHO*, *TSA*, *EPO*, *SOA*, *RSO*, *ESA*, *DE*, and *GA* for selected number of test functions. For all functions, the proposed algorithm reached a stable point. This behavior indicates that the suggested algorithm is convergent. Furthermore, for most functions, the proposed *SSC* algorithm achieves the lowest average of the best so far solutions the fastest. This fast convergence to the (near)-optimal solution makes the proposed *SSC* algorithm a promising tool to solve the problems that require fast computation, such as online optimization problems. The trajectory, average fitness, and convergence analysis of proposed *SSC* algorithm on different test functions are shown in Fig. 2.

3.7. Statistical results on IEEE CEC'17 test suite

For statistical significance a Wilcoxon ranksum test [57] is applied to the proposed *SSC* algorithm and the obtained findings are compared with the state-of-art technique. A meaning level of $p-$ value is set to 5% level of significance for comparison. Table 3 tabulates the Wilcoxon test and reveals that *SSC* efficiency is superior than competitor algorithms in most of the cases on *IEEE CEC '17* test functions.

3.8. Scalability analysis of SSC algorithm

The efficiency of the projected algorithm was checked in this sub-section using it scalability analysis as shown in Fig. 3. In this figure, the contemplated algorithm is tested on standard benchmark functions. It can be evaluated virtually empty, indicating that the *SSC* approach is consistent for the same functions executed. Moreover, the algorithms' lowest standard scores indicate their rapid convergence performance. Thus, It is obvious from this example that the *SSC* algorithm outperforms the lowest standard score in full test suites. As a consequence, it can be inferred that the *SSC* algorithm's convergence speed is quicker than any other for finding the best global optimum value with different dimensions during search.

4. SSC for complex engineering problems

The effectiveness of the contemplated *SSC* algorithm is tested on four engineering applications such as the welded beam, the multiple disk clutch brake, the car side impact of crash worthiness, and the pressure vessel. In order to optimize the limited problems of these functions, the contemplated approach has been compared with several recently developed algorithms.

4.1. Welded beam design

The goal of this design is to decrease the cost of producing a welded beam, as illustrated in Fig. 4. A load is added at the end of the member. The values for sold thickness (h), bar connection length (l), bar height (t), and bar thickness (b) for determining the minimum cost of this problem. The mathematical description of this problem is described as:

Consider $\vec{x} = [x_1 x_2 x_3 x_4] = [h\, l\, t\, b]$

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

Subject to:

$$g_1\left(\vec{x}\right) = \tau\left(\vec{x}\right) - 13600 \le 0$$
$$g_2\left(\vec{x}\right) = \sigma\left(\vec{x}\right) - 30000 \le 0$$
$$g_3\left(\vec{x}\right) = x_1 - x_4 \le 0$$
$$g_4\left(\vec{x}\right) = 0.10471\left(x_1^2\right) + 0.04811 x_3 x_4 \left(14 + x_2\right) - 5.0 \le 0$$
$$g_6\left(\vec{x}\right) = \delta\left(\vec{x}\right) - 0.25 \le 0$$
$$g_7\left(\vec{x}\right) = 6000 - p_c\left(\vec{x}\right) \le 0$$

where

$$\tau\left(\vec{x}\right) = \sqrt{(\tau') + (2\tau'\tau'')\frac{x_2}{2R} + (\tau'')^2}$$
$$\tau' = \frac{6000}{\sqrt{2} x_1 x_2}$$
$$\tau'' = \frac{MR}{J}$$
$$M = 6000\left(14 + \frac{x_2}{2}\right)$$
$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$$
$$j = 2\left\{x_1 x_2 \sqrt{2}\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

Table 2
The mean and standard deviation of fitness values for 30 runs obtained with the different algorithms on the IEEE CEC'17 functions.

F	Measures	SHO	TSA	EPO	SOA	RSO	ESA	DE	GA	SSC
CEC-1	Mean	1.01E+04	2.21E+09	7.07E+03	1.34E+05	5.38E+04	4.34E+04	7.77E+04	4.00E+05	**5.99E+03**
	STD	8.01E+03	1.54E+09	**3.95E+03**	2.69E+05	5.12E+04	2.52E+05	6.25E+04	8.83E+05	4.90E+03
CEC-2	Mean	1.24E+04	2.21E+09	9.41E+03	1.36E+05	5.61E+04	4.57E+04	8.00E+04	4.02E+05	**8.33E+03**
	STD	1.04E+04	1.54E+09	6.29E+03	2.71E+05	5.35E+04	2.54E+05	6.48E+04	8.85E+05	**2.24E+03**
CEC-3	Mean	1.29E+04	2.21E+09	9.91E+03	1.37E+05	5.66E+04	4.62E+04	8.05E+04	4.03E+05	**8.83E+03**
	STD	**1.09E+02**	1.54E+09	6.79E+03	2.72E+05	5.40E+04	2.55E+05	6.53E+04	8.86E+05	7.74E+03
CEC-4	Mean	1.21E+04	2.21E+09	9.11E+03	1.36E+05	5.58E+04	4.54E+04	7.97E+04	4.02E+05	**8.03E+03**
	STD	1.01E+04	1.54E+09	**5.99E+02**	2.71E+05	5.32E+04	2.54E+05	6.45E+04	8.85E+05	6.94E+03
CEC-5	Mean	2.01E+04	2.21E+09	1.71E+04	1.44E+05	6.38E+04	5.34E+04	8.77E+04	4.10E+05	**1.60E+04**
	STD	1.81E+04	1.54E+09	1.40E+04	2.79E+05	6.12E+04	**2.62E+02**	7.25E+04	8.93E+05	1.49E+04
CEC-6	Mean	1.22E+04	2.21E+09	9.21E+03	1.36E+05	5.59E+04	4.55E+04	7.98E+04	4.02E+05	**8.13E+03**
	STD	1.02E+04	1.54E+09	**6.09E+03**	2.71E+05	5.33E+04	2.54E+05	6.46E+04	8.85E+05	7.04E+03
CEC-7	Mean	9.12E+04	2.21E+09	8.82E+04	2.15E+05	1.35E+05	1.25E+05	1.59E+05	4.81E+05	**8.71E+04**
	STD	8.91E+04	1.54E+09	8.51E+04	3.50E+05	1.32E+05	3.33E+05	**1.44E+02**	9.64E+05	8.60E+04
CEC-8	Mean	1.91E+05	2.21E+09	1.88E+05	3.15E+05	2.35E+05	2.25E+05	2.59E+05	5.81E+05	**1.87E+05**
	STD	1.89E+05	1.54E+09	1.85E+05	4.50E+05	2.32E+05	4.33E+05	2.44E+05	1.06E+06	**1.86E+02**
CEC-9	Mean	1.44E+05	2.21E+09	1.41E+05	2.68E+05	1.87E+05	1.77E+05	2.11E+05	5.34E+05	**1.40E+05**
	STD	1.42E+05	1.54E+09	1.37E+05	4.03E+05	1.85E+05	3.86E+05	1.96E+05	1.02E+06	**1.38E+04**
CEC-10	Mean	2.21E+05	2.21E+09	2.18E+05	3.45E+05	2.65E+05	2.55E+05	2.89E+05	6.11E+05	**2.17E+05**
	STD	2.19E+05	1.54E+09	2.15E+05	4.80E+05	2.63E+05	4.63E+05	2.74E+05	1.09E+06	**2.16E+04**
CEC-11	Mean	1.87E+05	2.21E+09	1.84E+05	3.11E+05	2.31E+05	2.20E+05	2.54E+05	5.77E+05	**1.83E+05**
	STD	1.85E+05	1.54E+09	1.81E+05	**4.46E+04**	2.28E+05	4.29E+05	2.39E+05	1.06E+06	1.82E+05
CEC-12	Mean	1.19E+06	2.21E+09	1.18E+06	1.31E+06	1.23E+06	1.22E+06	1.25E+06	1.58E+06	**1.18E+06**
	STD	1.18E+06	1.54E+09	1.18E+06	1.45E+06	1.23E+06	1.43E+06	1.24E+06	2.06E+06	**1.18E+05**
CEC-13	Mean	1.42E+06	2.21E+09	1.42E+06	1.55E+06	1.47E+06	1.45E+06	1.49E+06	1.81E+06	**1.42E+06**
	STD	1.42E+06	1.54E+09	1.42E+06	1.68E+06	1.46E+06	**1.66E+05**	1.47E+06	2.29E+06	1.42E+06
CEC-14	Mean	2.00E+06	2.21E+09	1.32E+06	1.45E+06	1.37E+06	1.35E+06	1.39E+06	1.71E+06	**1.32E+06**
	STD	1.32E+06	1.54E+09	1.32E+06	1.58E+06	1.36E+06	**1.56E+05**	1.37E+06	2.19E+06	1.32E+06
CEC-15	Mean	9.19E+04	2.21E+09	8.89E+04	2.16E+05	1.36E+05	1.25E+05	1.60E+05	4.82E+05	**8.78E+03**
	STD	8.98E+04	1.54E+09	8.58E+04	3.51E+05	1.33E+05	3.34E+05	1.44E+05	9.65E+05	**8.67E+03**
CEC-16	Mean	1.01E+05	2.11E+07	5.07E+03	1.24E+05	5.38E+05	2.44E+04	7.78E+05	2.08E+05	**5.99E+02**
	STD	7.01E+04	1.44E+08	**3.95E+02**	3.59E+04	4.12E+04	2.42E+04	5.25E+04	7.73E+04	2.70E+02
CEC-17	Mean	1.24E+05	2.21E+07	7.41E+04	1.36E+04	5.61E+04	3.57E+05	7.00E+05	2.02E+05	**8.33E+02**
	STD	1.04E+05	1.54E+08	5.29E+03	2.71E+05	4.35E+05	2.54E+05	8.48E+05	2.85E+05	**1.21E+03**
CEC-18	Mean	2.29E+05	2.21E+09	8.71E+04	1.37E+05	4.76E+03	4.62E+04	8.75E+03	4.73E+04	**8.83E+02**
	STD	**1.09E+01**	1.74E+08	6.79E+04	2.52E+05	4.40E+05	2.45E+05	6.53E+05	6.66E+05	7.44E+03
CEC-19	Mean	1.21E+05	2.21E+08	7.11E+04	1.36E+05	6.58E+04	4.54E+08	8.97E+04	8.02E+06	**8.03E+02**
	STD	1.01E+05	2.54E+06	**5.99E+02**	2.71E+06	8.32E+04	2.54E+08	7.45E+04	8.85E+08	7.94E+04
CEC-20	Mean	2.01E+04	2.21E+09	1.71E+04	1.44E+05	6.48E+04	5.34E+04	8.47E+04	4.15E+05	**5.11E+03**
	STD	1.81E+04	1.54E+07	1.40E+04	2.59E+07	6.12E+04	**2.62E+02**	7.55E+04	8.53E+07	1.49E+07
CEC-21	Mean	1.62E+05	2.21E+09	9.61E+05	1.36E+05	5.89E+05	4.75E+05	7.48E+04	4.52E+05	**5.11E+02**
	STD	4.02E+05	1.54E+04	**6.09E+02**	4.71E+05	5.33E+05	4.44E+05	6.46E+05	4.45E+05	4.04E+05
CEC-22	Mean	5.12E+06	2.21E+09	3.82E+06	2.25E+05	5.35E+06	3.15E+05	5.29E+07	5.31E+05	**1.34E+04**
	STD	3.41E+05	1.54E+08	3.41E+04	3.50E+07	1.42E+05	4.33E+07	**1.44E+02**	5.74E+05	4.80E+06
CEC-23	Mean	1.91E+06	4.21E+09	1.88E+06	4.15E+05	2.35E+06	4.25E+05	2.59E+06	4.81E+05	**1.46E+04**
	STD	1.89E+06	1.34E+09	1.85E+06	4.30E+05	2.32E+06	3.53E+06	2.44E+05	1.56E+07	**1.11E+02**
CEC-24	Mean	3.44E+06	2.21E+09	1.41E+06	6.68E+05	1.87E+06	6.77E+05	2.11E+06	1.34E+06	**2.10E+05**
	STD	1.22E+06	1.44E+06	1.37E+05	4.03E+05	1.45E+06	3.86E+05	1.46E+06	1.02E+06	**1.12E+04**
CEC-25	Mean	2.41E+06	3.21E+06	2.48E+05	3.45E+06	3.45E+05	2.55E+06	2.49E+05	3.41E+06	**1.12E+04**
	STD	2.49E+06	1.54E+09	2.45E+06	4.80E+05	2.43E+06	4.63E+05	2.44E+06	1.09E+07	**3.13E+03**
CEC-26	Mean	1.67E+05	2.21E+07	1.64E+05	3.11E+05	2.61E+07	2.60E+05	2.54E+07	5.37E+05	**1.13E+04**
	STD	1.83E+04	1.54E+09	1.81E+04	**2.15E+02**	2.23E+04	4.29E+05	2.33E+04	1.06E+06	1.83E+04
CEC-27	Mean	1.19E+06	2.11E+07	1.18E+06	1.31E+06	1.53E+07	1.22E+06	1.55E+07	1.38E+07	**1.02E+05**
	STD	1.18E+06	1.44E+07	1.18E+06	1.35E+07	1.43E+06	1.43E+07	1.44E+06	2.46E+07	**1.12E+04**
CEC-28	Mean	1.42E+06	2.21E+06	1.72E+06	1.55E+06	1.77E+07	1.45E+06	1.49E+07	1.71E+06	**1.52E+05**
	STD	1.42E+07	1.54E+09	1.62E+06	1.68E+07	1.46E+06	**1.50E+05**	1.67E+06	2.29E+07	1.62E+06
CEC-29	Mean	2.00E+05	2.11E+09	1.52E+05	1.35E+06	1.87E+06	1.35E+05	1.99E+06	1.41E+05	**1.11E+05**
	STD	1.32E+06	1.54E+09	1.31E+06	1.58E+06	1.26E+06	1.56E+05	**1.37E+04**	2.79E+06	1.82E+06
CEC-30	Mean	1.13E+04	1.26E+09	5.84E+04	6.18E+05	3.33E+05	6.23E+05	7.61E+05	8.89E+05	**7.28E+03**
	STD	8.38E+04	1.24E+06	8.18E+04	3.61E+04	1.33E+05	3.64E+04	1.74E+05	9.35E+05	**7.52E+03**

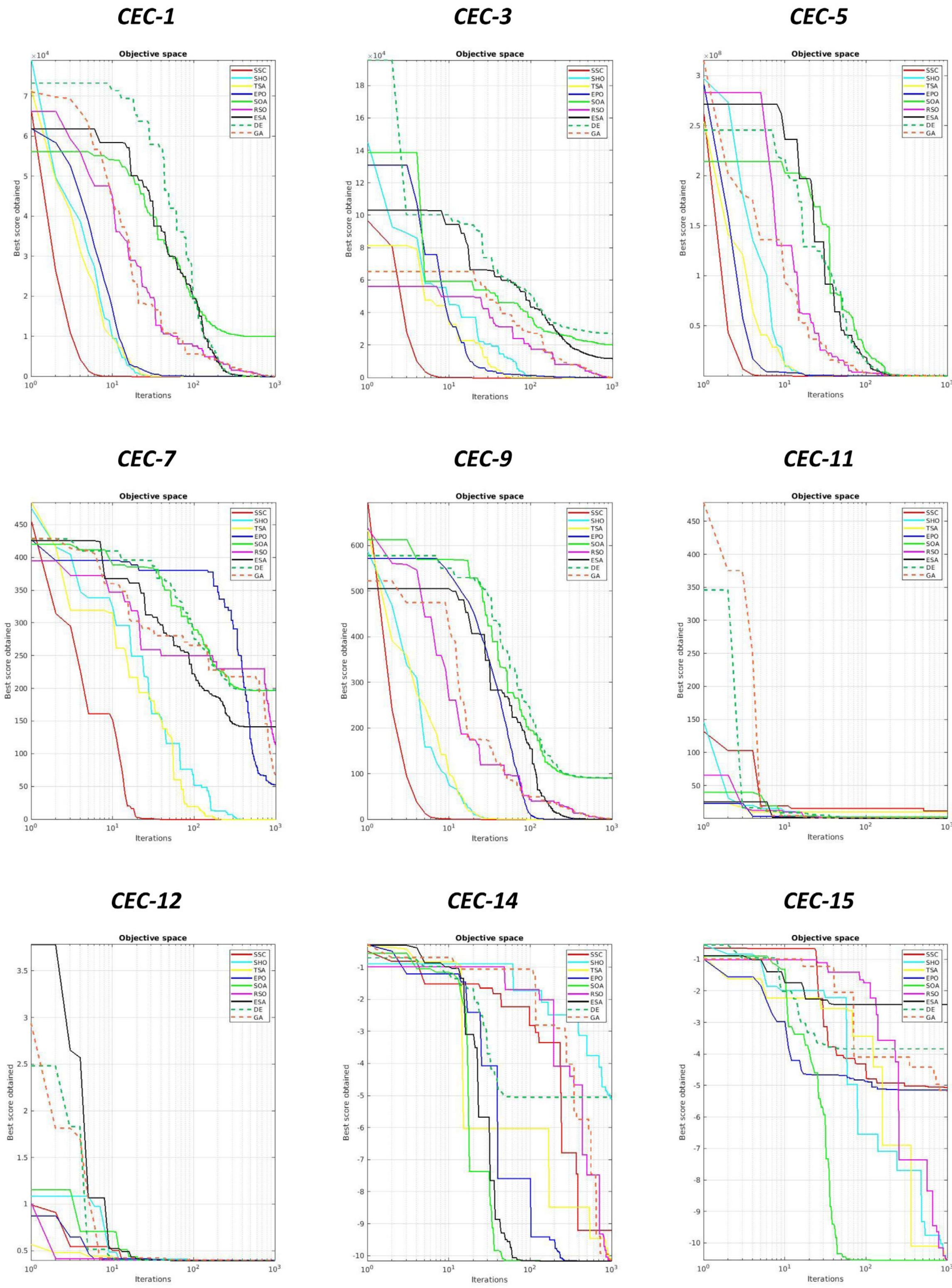

Fig. 1. Convergence analysis of proposed and other competitor algorithms.

$\sigma\left(\vec{x}\right) = \frac{504000}{x_4 x_3^2}$

$\delta\left(\vec{x}\right) = \frac{65856000}{\left(30\times10^6\right) x_4 x_3^3}$

$$p_c\left(\vec{x}\right) = \frac{4.013\left(30\times10^6\right)\sqrt{\frac{x_3^2 x_4^6}{36}}}{196}\left(1 - \frac{x_3\sqrt{\frac{30\times10^6}{4\left(12\times10^6\right)}}}{28}\right)$$

with $0.1 \leq x_1, x_4 \leq 2.0$ *and* $0.1 \leq x_2, x_3 \leq 10.0$.

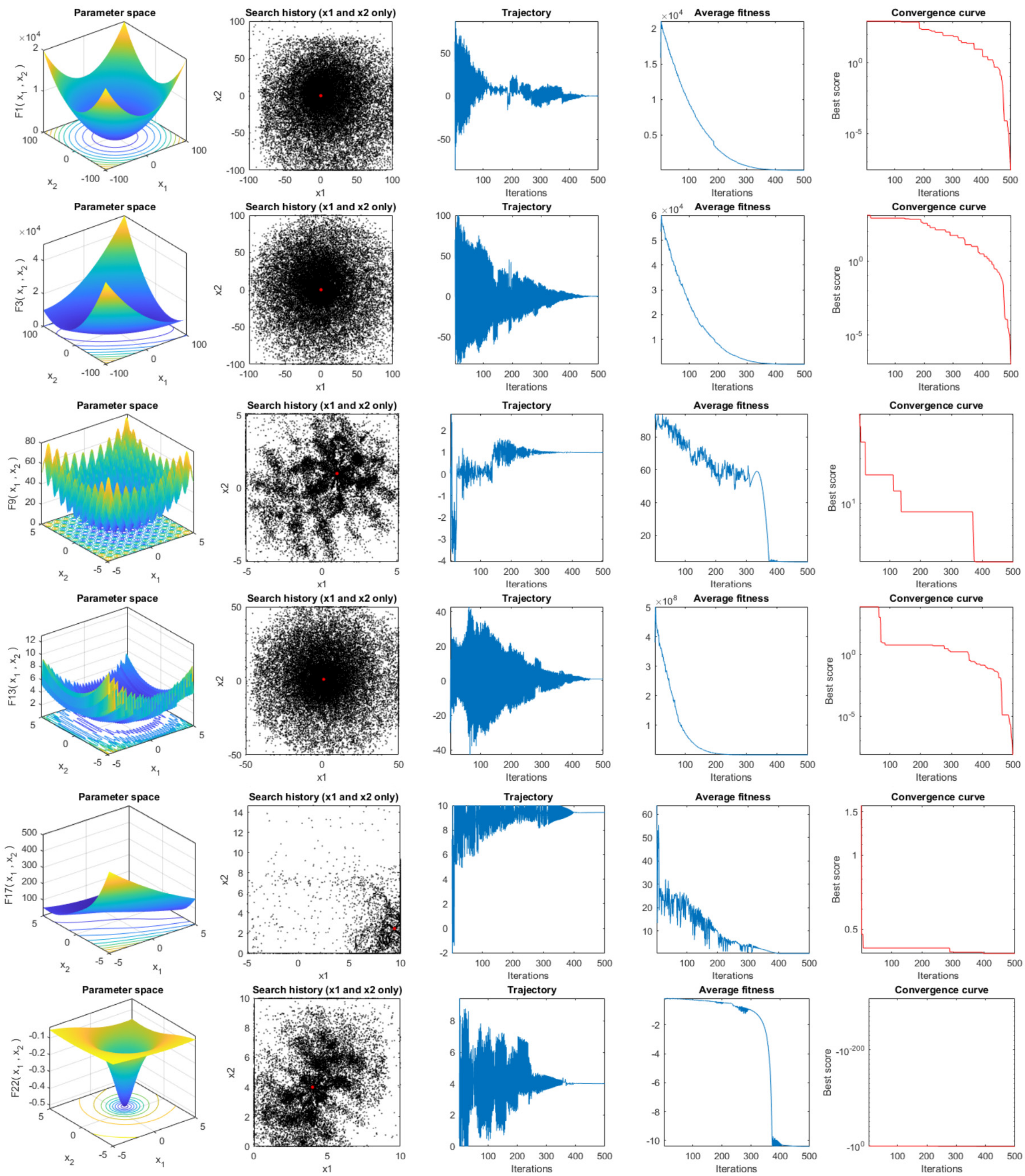

Fig. 2. Trajectory, average fitness, and convergence analysis of proposed algorithm.

The performance of the proposed algorithm has been compared with the latest optimizations such as *CPSO*, *Random*, *GA*, *RO*, *GWO*, *MVO*, and *GSA*. All these numerical solutions (see Table 4) show that the version proposed can provide the extremely precise and least optimal scores in contrast to other models.

4.2. Tension/compression spring design

The key goal of this design, as shown in Fig. 5, is to reduce the weight of the pressure/voltage source. The constraints of this design include shear stress, floating frequency and limited floating deflections. During this study different recent optimization techniques have been applied to the best or accurate solution of this design. The mathematical description of this problem is described as:

Consider: $\vec{x} = [x_1 x_2 x_3] = [d\, D\, N]$

Minimize $f(\vec{x}) = (x_3 + 2)\, x_2 x_1^2$

subject to:

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

$$g_2(\vec{x}) = \frac{4x_2^2 - x_1 x_2}{12566\left(x_2 x_1^3 - x_1^4\right)} + \frac{1}{5108 x_1^2} - 1 \leq 0$$

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

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

with $0.05 \leq x_1 \leq 2.0,\ 0.25 \leq x_2 \leq 1.3,\ and\ 2.0 \leq x_3 \leq 15.0$ It can easily be studied by literature that is the most useful meta-heuristic to provide accurate solutions of this form of design.

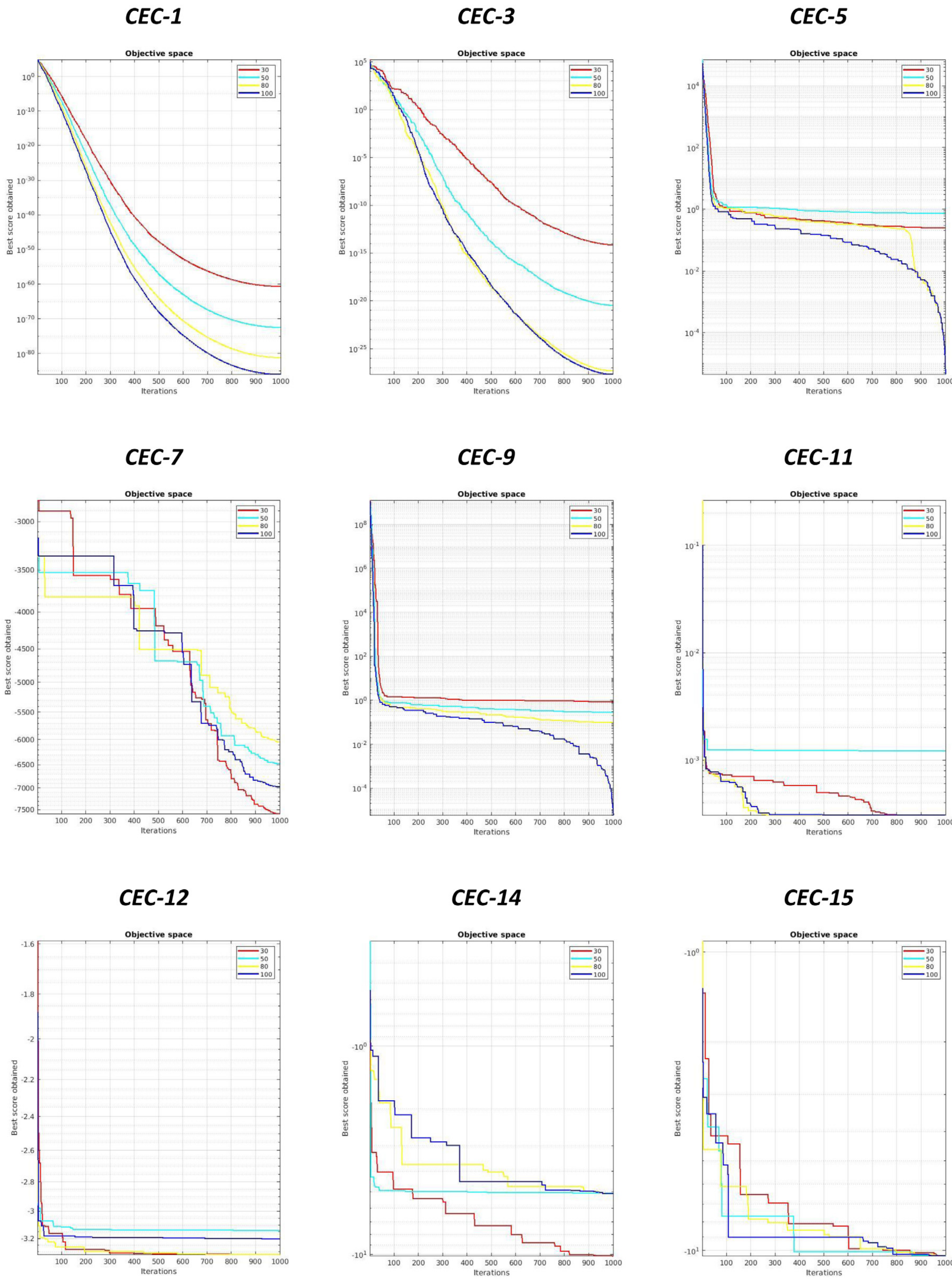

Fig. 3. Scalability analysis of the proposed SSC algorithm.

In Table 5, all simulation outputs obtained from algorithms are recorded.

During this analysis, a number of recent techniques such as *DE*, *GSA*, *Halton-PSO*, *RO*, *HSSAHHO ES*, *GA*, *GWO*, *MFO*, *PSO*, *WOA*, and *Chimp* have tested the convergence performance of the proposed process. These are the findings, which prove that this algorithm is capable of solving problems that are superior to others.

4.3. Pressure Vessel design

The key reason for this design is to minimize design pressure vessel construction costs, as can be seen in Fig. 6. This construction expense includes;associated with geometrical pressure vessel parameters, materials, welding, and shaping. Thus, calculation of the optimal solutions of geometric constants is important

Table 3
p-values obtained from the Wilcoxon ranksum test for IEEE CEC' 17 benchmark test functions.

F	SHO	TSA	EPO	SOA	RSO	ESA	DE	GA
CEC-1	0.0007	0.0004	0.0004	0.0003	0.0067	0.0001	0.0007	0.0060
CEC-2	0.0012	0.0003	0.0003	0.0057	0.0002	0.0083	0.0035	0.0050
CEC-3	0.0002	0.0250	0.0133	0.0053	0.0006	0.0006	0.0007	0.0004
CEC-4	0.0042	0.0226	0.0039	0.0076	0.0023	0.0040	0.0003	0.0001
CEC-5	0.0006	0.0369	0.0005	0.0003	0.0009	0.2237	0.0093	0.0552
CEC-6	0.0076	0.0007	0.0338	0.0097	0.0269	0.4118	0.0072	0.0423
CEC-7	0.0087	0.0092	0.0035	0.0043	0.0022	0.0010	0.0004	0.0003
CEC-8	0.0032	0.5671	0.0009	0.0842	0.0080	0.0001	0.0028	0.0077
CEC-9	0.0001	0.0001	0.0077	0.0095	0.0001	0.0001	0.0003	0.0001
CEC-10	0.0001	0.0001	0.0031	0.0523	0.0440	0.0083	0.0004	0.0009
CEC-11	0.0002	0.0086	0.0009	0.0004	0.0007	0.0023	0.0053	0.0095
CEC-12	0.0001	0.0005	0.0001	0.0001	0.0637	0.0001	0.0042	0.0277
CEC-13	0.0001	0.0251	0.0001	0.0046	0.0048	0.0001	0.0003	0.0006
CEC-14	0.0001	0.0080	0.0036	0.0033	0.0796	0.0001	0.0053	0.0006
CEC-15	0.0009	0.0063	0.0076	0.0193	0.0119	0.0001	0.0076	0.0004
CEC-16	0.0009	0.0063	0.0016	0.0195	0.0119	0.0011	0.0075	0.0004
CEC-17	0.0009	0.0063	0.0016	0.0153	0.0119	0.0001	0.0076	0.0004
CEC-18	0.5009	0.0063	0.0016	0.0101	0.0319	0.0101	0.0076	0.0004
CEC-19	0.0004	0.0063	0.0046	0.0193	0.0419	0.0007	0.0026	0.0034
CEC-20	0.0036	0.0032	0.0016	0.0145	0.0126	0.0012	0.0046	0.0011
CEC-21	0.0067	0.0060	0.0047	0.0112	0.0119	0.0001	0.0076	0.0004
CEC-22	0.0009	0.0063	0.0076	0.0193	0.0119	0.0001	0.0076	0.0004
CEC-23	0.0089	0.0055	0.0076	0.0089	0.0110	0.0069	0.0071	0.0001
CEC-24	0.0010	0.0063	0.0010	0.0003	0.0119	0.0010	0.0071	0.0008
CEC-25	0.0038	0.0009	0.0016	0.0008	0.0100	0.0001	0.0001	0.0002
CEC-26	0.0001	0.0013	0.0016	0.0111	0.0008	0.0001	0.0006	0.0006
CEC-27	0.0097	0.0036	0.0022	0.0056	0.0077	0.0786	0.0001	0.0001
CEC-28	0.0001	0.0001	0.0016	0.0001	0.0100	0.0001	0.0044	0.0003
CEC-29	0.0003	0.0003	0.0006	0.0003	0.0009	0.0001	0.0006	0.0004
CEC-30	0.0001	0.0014	0.0086	0.0111	0.0063	0.0963	0.0034	0.0893

to achieve the least costs for this function. The mathematical description of this problem is described as:

Minimize $f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$

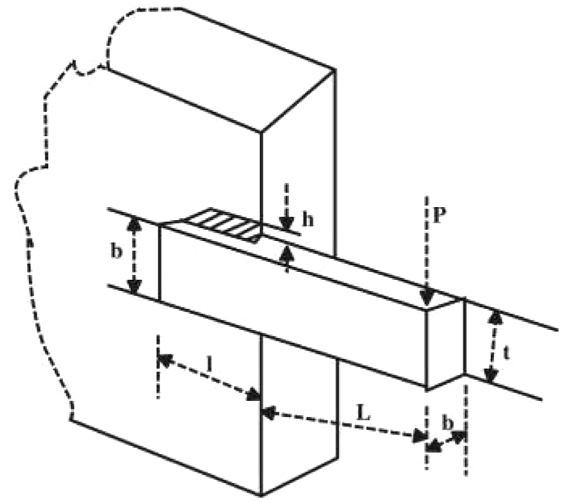

Fig. 4. Welded beam design.

Subject to:

$g_1(x) = -x_1 + 0.0193x$

$g_2(x) = -x_2 + 0.00954x_3 \ \leq 0$

$g_3(x) = -\pi x_3^2 x_4 - (4/3)\pi x_3^3 + 1,296,000 \leq 0$

$g_4(x) = x_4 - 240 \leq 0$

$0 \leq x_i \leq 100, \ \ i = 1, 2$

$10 \leq x_i \leq 200, \ \ i = 3, 4$ The optimal solutions of *SSC* algorithm and the other meta-heuristics *CGA*, *HPSO*, *CSA*, *CPSO*, *WOA-DE*, *WOA-BSA*, *DGA*, *CDE*, and *Chimp* algorithm respectively for this design are tabulated in Table 6.

The efficiency of the algorithm proposed is shown by the fact that it providing produces the optimal or correct outcome

Table 4
The optimal solutions of the algorithms on welded beam design problem.

Algorithm	h	l	t	b	Optimal solution
RANDOM [58]	0.4575	4.7313	5.0853	0.6600	4.1185
GA [58]	0.2489	6.1730	8.1789	0.2533	2.4331
Chimp [58]	0.1656	4.0829	10	0.2046	1.9036
GSA [58]	0.1821	3.8569	10.0000	0.2023	1.8799
RO [58]	0.20368	3.52846	9.00423	0.20724	1.73534
CPSO [58]	0.202369	3.544214	9.048210	0.205723	1.73148
WOA [58]	0.2053	3.4842	9.0374	0.2062	1.7304
MVO [58]	0.2054	3.4731	9.0445	0.2056	1.7264
GWO [58]	0.2056	3.4783	9.0368	0.2057	1.7262
HSSAHHO [58]	0.205727	3.470523	9.03673	0.205729	1.724867
SSC	0.1992	3.4307	9.1045	0.2051	**1.7222**

Table 5
The optimal solutions of the algorithms on Tension/compression spring design problem.

Algorithm	d	D	N	Optimum weight
RO [58]	0.0413	0.3490	11.762	0.0126
DE [58]	0.0516	0.3547	11.410	0.0126
ES [58]	0.0519	0.3639	10.890	0.0126
GA [58]	0.05148	0.35166	11.6322	0.01270
PSO [58]	0.05172	0.35764	11.2445	0.01267
GWO [58]	0.05169	0.35673	1.2888	0.01266
MFO [58]	0.05100	0.36410	10.8684	0.01266
WOA [58]	0.05120	0.34521	12.0040	0.01267
GSA [58]	0.050276	0.323680	13.525410	0.0127022
Halton-PSO [58]	0.0537	0.4058	8.9027	0.0126981
HSSAHHO [58]	0.0514215	0.3535735	11.354662	0.012485407
Chimp [58]	0.052457	0.34897	10.99876	0.012482
SSC	0.052076	0.34550	10.6537	**0.011856**

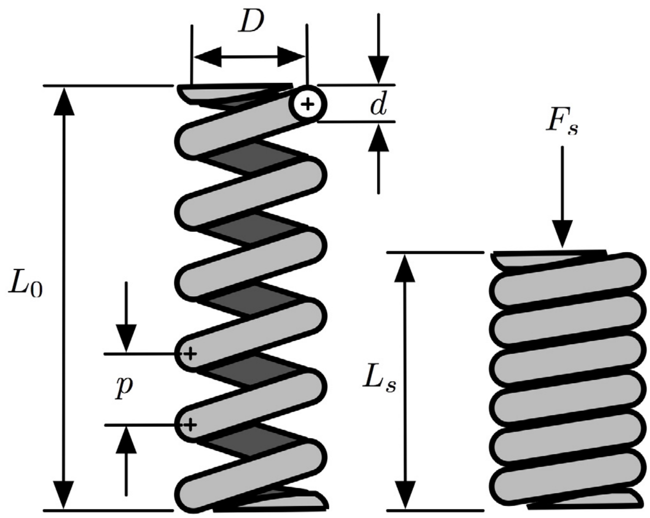

Fig. 5. Tension/compression spring design.

between other current meta-heuristics in this design and is able to determine the best design at minimal expense.

4.4. Multiple disk clutch brake design

The primary aim of this design is to reduce the weight of a multi-disk clutch brake by consideration of different discrete variables such as internal radius, number of outer radius friction surfaces, driving force, and disk thickness. Eight different constraints are posed in this design based on operating conditions and geometry. The Global Optima solution for this design is $f(x)$ for a single active constant at $X = (70, 90, 1, 810, 3)$. This pattern was shown in Fig. 7. This problem's mathematical explanation is as follows: Minimize$f(x) = \pi(x_2^2 - x_1^2)x_3(x_5 + 1)\rho$

Variable range

$x_1 \varepsilon \{60, 61, \ldots, 79, 80\}$

$x_2 \varepsilon \{90, 91, \ldots, 109, 110\}$

$x_3 \varepsilon \{1, 1.5, \ldots, 3\}$

$x_4 \varepsilon \{600, 610, \ldots, 990, 1000\}$

$x_5 \varepsilon \{2, 3, \ldots, 8, 9\}$

Subject to;$g_1(x) = x_2 - x_1 - \Delta r \leq 0$

$g_2(x) = L_{max} - (x_5 + 1)(x_3 + \delta) \leq 0$

$g_3(x) = p_{max} - p_{rz} \leq 0$

$g_4(x) = p_{max} v_{sr,max} - p_{rz} v_{sr} \leq 0$

$g_5(x) = v_{sr,max} - v_{sr} \leq 0$

$g_6(x) = M_h - sM_s \leq 0$

$g_7(x) = T \leq 0$

$g_8(x) = T_{max} - T \leq 0$

$\Delta r = 20$ (mm), $I_z = 55$ (kg, mm^2), $p_{max} = 1$ (MPa), $T_{max} = 15$ (s), $\mu = 0.50$

$s = 1.50.M_s = 40$ (N m), $M_f = 3$ (N m), $n = 250$ (rpm), $v_{sr,max} = 10$ $(\frac{m}{s})$, $L_{max} = 30$ (mm)

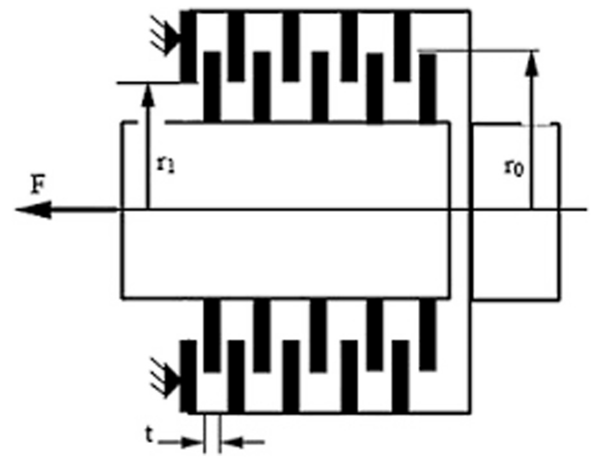

Fig. 7. Multiple disk clutch brake design.

This method has been compared with numerous recent optimization algorithms including *EPO*, *SOA*, *RSO*, *APSO*, *SCA*, *FSO*, *EGWO*, *TACPSO*, *MPSO*, and *Chimp* algorithms. Table 7, reports the results of the contrast to find the finest scores provided by such meta-heuristics. Simulation findings show that the proposed algorithm outperforms others in finding the best minima solution for this design.

4.5. Gear train design

The key motive for this design is to integrate the gear numbers of an automatic transmission system as shown in Fig. 8. It is used in cars to minimize equipment errors. This included six architecture variables based on the gear number (N_1, N_2, N_3, N_4, N_5, N_6) that only take the entered integer constant values. There are three additional architecture variables, including gear modules (m_1 and m_2) and planets gear numbers (P), which can only take those discrete constant values. Eleven restrictions are considered subject to restrictions in this design. The mathematical description of this problem is described as:

Maximize $f(x) = max\,|i_k - i_{ok}|\,;\, k = \{1, 2, R\}$

where

$i_1 = N_6/N_4$

$i_{01} = 3.11$

$i_2 = \frac{N_6(N_1N_3+N_2N_4)}{N_1N_3(N_6-N_4)}$

$i_{02} = 1.84$

$i_R = -\frac{N_2N_6}{N_1N3}$

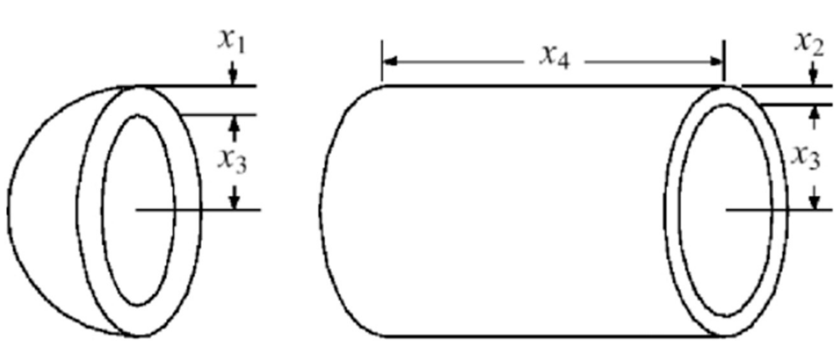

Fig. 6. Pressure vessel design.

Table 6
The optimal solutions of the algorithms on pressure vessel design problem.

Algorithm	X_1	X_2	X_3	X_4	Min $f(x)$
CGA [58]	0.812500	0.437500	40.323900	200.0000	6288.744500
DGA [58]	0.812500	0.437500	42.097400	176.654000	6059.946300
CPSO [58]	0.812500	0.437500	42.091300	176.746500	6061.077700
HPSO [58]	0.812500	0.437500	42.098400	176.636600	6059.714300
CDE [58]	0.812500	0.437500	42.098411	176.637690	6059.734000
CSA [58]	0.812500	0.437500	42.098445	176.636599	6059.714363
WOA [58]	0.812500	0.437500	41.825682	180.046141	6093.211997
WOA-DE [58]	0.812500	0.437500	42.098497	176.635957	6059.708057
WOA-BSA [58]	0.812500	0.437500	42.098497	176.635954	6059.708025
Chimp [58]	0	0	40.3301	200.0000	6058.6266
SSC	1.0783	5.36E−06	64.9278	10.0000	**2313.0923**

Table 7
The optimal solutions of the algorithms on Multiple disk clutch brake design problem.

Algorithm	X_1	X_2	X_3	X_4	X_5	Min $f(x)$
EPO [58]	70	90	1.5	1000	3	0.4704
SOA [58]	77.1459	97.2218	1	628.1937	3.3809	0.3758
APSO [58]	76	96	1	840	3	0.337181
RSO [58]	70	90	1	810	3	0.313657
WCA [58]	70	90	1	910	3	0.313657
FSO [58]	70	90	1	870	3	0.313656
SCA [58]	68.8526	90	1	1000	2.6774	0.3027
Chimp [58]	69.8782	90	1	1000	2.6537	0.2880
EGWO [58]	69.4920	90	1.0099	1000	2.3690	0.2727
TACPSO [58]	75.0044	95.0044	1	1000	2.1779	0.2648
MPSO [58]	70	90	1	1000	2.3128	0.2598
SSC	31.3954	51.5897	1	1000	4.3902	**0.2230**

Table 8
The optimal solutions of the algorithms on gear train design problem.

Optimal value	SSA [58]	EGWO [58]	TACPSO [58]	MPSO [58]	SCA [58]	Chimp [58]	SSC
X_1	35.4479	5.0200	65.7076	31.1533	56.9155	33.1809	36.1678
X_2	15.4274	2.7175	46.5288	54.4900	29.8118	19.6612	13.5101
X_3	14.0812	3.1710	20.5288	34.3362	13.5100	16.3498	17.9456
X_4	26.2348	2.6564	21.3166	26.5122	27.9305	18.5740	33.1672
X_5	15.9919	2.0467	20.0452	13.5100	24.4353	16.8072	35.7526
X_6	65.6218	12.6182	75.7920	117.4903	102.7000	68.8582	120.4009
X_7	3.1904	1.8098	2.3735	3.3266	1.5732	2.6944	3.1162
X_8	1.0999	1.0563	0.5100	0.5100	0.5100	0.6425	1.8390
X_9	0.5100	0.6231	2.2143	0.5100	0.8265	0.9238	0.6198
Min $f(x)$	1.1233	2.2233	0.5328	1.2583	0.9789	0.5332	**0.5189**

Table 9
The Statistical solutions of the algorithms on gear train design problem.

Algorithm	X_{min}	X_{max}	X_{mean}	X_{sd}	X_{median}
EGWO [58]	2.2233	4.2366e+07	8.4587e+05	5.1177e+06	708.7422
MPSO [58]	1.2583	6.2092e+04	819.7833	6.9427e+03	1.6460
SSA [58]	1.1233	1.0291e+04	363.5411	1.6264e+03	2.8148
SCA [58]	0.9789	2.6436e+04	478.8328	3.0108e+03	2.0076
Chimp [58]	0.5332	3.8973e+03	52.7726	435.3218	0.9433
TACPSO [58]	0.5328	2.1118e+04	1.0605e+03	4.5581e+03	0.5946
SSC	0.5250	27.9893	1.8089	4.2453	1.2532

$i_{0R} = -3.11$

$X = \{N_1, N_2, N_3, N_4, N_5, p, m_1, m_2\},$

Subject to:

$g_1(X) = m_3(N_6 + 2.5) \leq D_{max}$

$g_2(X) = m_1(N_1 + N_2) + m_1(N_2 + 2) \leq D_{max}$

$g_3(X) = m_3(N_4 + N_5) + m_3(N_5 + 2) \leq D_{max}$

$g_4(X) = |m_1(N_1 + N_2) - m_3(N_6 - N_3)| \leq m_1 + m_3$

$g_5(X) = (N_1 + N_2)sin(\frac{\pi}{p}) - N_2 - 2 - \delta_{22} \geq 0$

$g_6(X) = (N_6 - N_3)sin(\frac{\pi}{p}) - N_3 - 2 - \delta_{33} \geq 0$

$g_7(X) = (N_4 + N_5)sin(\frac{\pi}{p}) - N_5 - 2 - \delta_{55} \geq 0$

$g_8(X) = (N_6 - N_3)^2 + (N_4 + N_5)^2 - (N_6 - N_3)(N_4 + N_5)cos(\frac{2\pi}{p} - \beta) \leq (N + 3 + N_5 + 2 + \delta_{35})^2$

$where$

$\beta = \frac{cos^{-1}(N_6 - N_3)^2 + (N_4 + N - 5)^2 - (N_3 + N_5)^2}{2(N_6 - N_3)(N_4 + N_5)}$

$g_9(X) = N_6 - 2N_3 - N_4 - 4 - 2\delta_{34} \geq 0$

$g_10(X) = N_6 - N_4 - 2N_5 - 4 - 2\delta_{56} \geq 0$

$h(X) = \frac{N_6 - N_4}{p} = integer,$

$where$

$D_{max} = 220, p = (3, 4, 5), m_1, m_3 = (1.75, 2.0, 2.25, 2.5,$

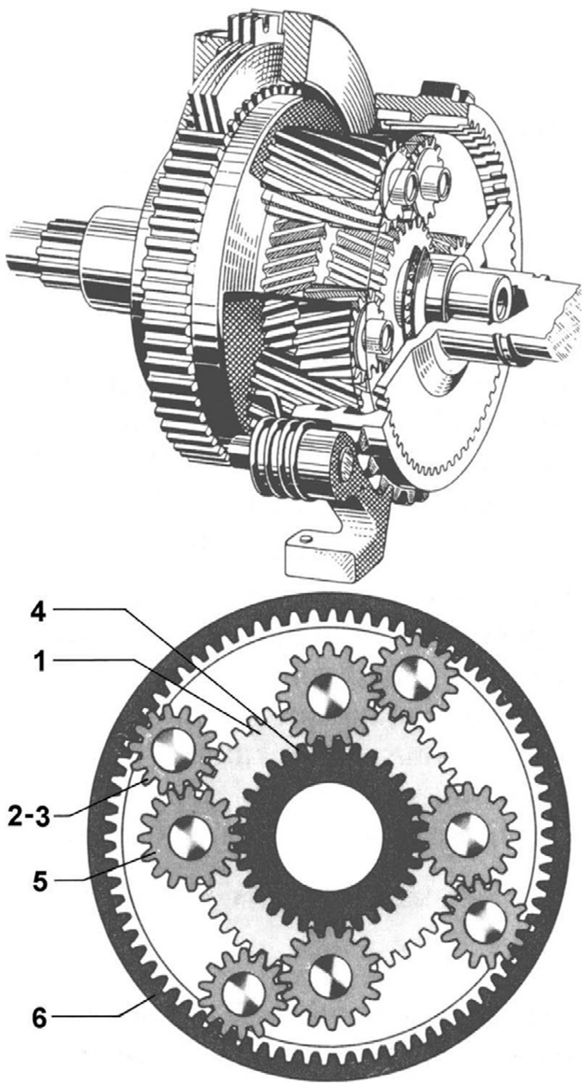

Fig. 8. Gear train design.

$$2.75, 3.0), \delta_{22}, \delta_{33}, \delta_{55}, \delta_{35}, \delta_{56}, = 0.5$$
$$17 \le N_1 \le 96,\ 14 \le N_2 \le 54,\ 14 \le N_3 \le 51,\ 17 \le N_4 \le 46,$$
$$14 \le N_5 \le 51,\ 48 \le N_6 \le 124,\ N_i = integer.$$

The mathematical and statistical solutions to this study are listed in Tables in Tables 8–9. In these tables, the efficiency of algorithms was compared in a variety of ways, including best optimal value, mean, median, minima, maximum, and standard deviation. All of these simulation findings indicate that the proposed technique provides the most effective or best solutions for this design.

4.6. Car side impact problem

Car side impact problem is a restricted problem of optimization to optimize crash resistance of the vehicle [59]. This problem consists of seven variables: internal b-pillar thickness, internal B-pillar strengthening, inner floor, cross seats, door beam, belt line armor and rail roof (see Fig. 9). This is a problem with seven variables. This problem is defined to minimize the car's weight.

In a car side-impact scenario, the following impulses will be exchanged:

- The primary momentum exchange takes place between the target vehicle and the *MDB*. In this case, the rigid body acceleration of the target vehicle will increase as the *MDB* speed will decrease until both the *MDB* and the target vehicle at some stage hit a common speed.
- The momentum between the *MDB* and the door is changed. The door can easily reach the high speed of the *MDB*.
- Finally, the energy exchange occurs when the intruding door collides with the permanent side-impact dummy (*SID*). The dummy pelvis is quickly pushed to the side and struck by a door.

The operating forces are described as follows:

- The raise strength of the *MDB*
- The body of the systemic resistance to door invasion of the affected vehicle.
- The touch power door-to-dummy, the reaction force of the dumb even.

The following is the mathematical terminology of the problem:

$$\begin{aligned}
&f_1(z) = 1.98 + 4.9z_1 + 6.67z_2 + 6.98z_3 + 4.01z_4\\
&\ + 1.78z_5 + 0.00001z_6 + 2.73z_7,\\
&\text{Subject to:}\\
&g_1(z) = 1.16 - 0.3717z_2z_4 - 0.0092928z_3 \le 1.0,\\
&g_2(z) = 0.261 - 0.0159z_1z_2 - 0.06486z_1 - 0.019z_2z_7\\
&\ + 0.0144z_3z_5 + 0.0154464z_6 \le 0.32,\\
&g_3(z) = 0.214 + 0.00817z_5 - 0.045195z_1 - 0.0135168z_1\\
&\ + 0.03099z_2z_6 - 0.018z_2z_7\\
&\ + 0.007176z_3 + 0.023232z_3 - 0.00364z_5z_6 - 0.018z_2z_2 \le 0.32,\\
&g_4(z) = 0.74 - 0.61z_2 - 0.031296z_3\\
&\ - 0.031872z_7 + 0.227z_2z_2 \le 0.32,\\
&g_5(z) = 28.98 + 3.818z_3 - 4.2z_1z_2\\
&\ + 1.27296z_6 - 2.68065z_7 \le 0.32,\\
&g_6(z) = 33.86 + 2.95z_3 - 5.057z_1z_2\\
&\ - 3.795z_2 - 3.4431z_7 + 1.45728 \le 0.32,\\
&g_7(z) = 46.36 - 9.9z_2 - 4.4505z_1 \le 0.32,\\
&g_8(z) = 4.72 - 0.5z_4 - 0.19z_2z_3 \le 4.0,\\
&g_9(z) = 10.58 - 0.674z_1z_2 - 0.67275z_2 \le 9.9,\\
&g_{10}(z) = 16.45 - 0.489z_3z_7 - 0.843z_5z_6 \le 15.7,\\
&\text{where,}\\
&0.5 \le z_1, z_3, z_4 \le 1.5,\ \ 0.4 \le z_6, z_7 \le 1.2,\\
&\ 0.45 \le z_2 \le 1.35,\ \ 0.875 \le z_5 \le 2.625.
\end{aligned} \tag{24}$$

The solution is a model that suddenly changes the curvature. It is known as the ideal front point in Pareto with the greatest distance from the line that in this study connects the person minimum. The ideal Pareto front solution typically means a higher quality compromise solution and is less far away from the utopia point (considering other points on the front) in most situations. In general, such a design point differentiates between different regions in objective space. In the following discussion, the utopia or ideal point constitutes an objective space in which both objectives achieve their individual minimum. (See Table 10.)

Numerical studies are conducted in two cases:

- All variables of nature are considered continuous.
- All variables in thickness design are continuous while variables in material design are known as discrete.

Table 10
Details of design variable categorization.

Description	X1	X2	X3	X4	X5	X6	X7
Common design variables			Yes		Yes		Yes
Local variables to structural weight	Yes	Yes		Yes			
Local variables to door velocity						Yes	

Table 11
The results obtained by the proposed algorithm for (Case 1).

Description	X1	X2	X3	X4	X5	X6	X7	X8	X9	Structural weight (kg)	Door velocity (ms^{-1})	HIC
Baseline values	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.190	0.191	29.01	15.107	229
Solution for Case 1	0.49	0.61	0.50	0.50	1.49	1.42	1.30	0.340	0.191	20.41	14.13	210

Description	G1	G2	G3	G4	G5	G6	G7	G8	G9
Constraint values for solution	0.991	4.392	9.91	27.93	28.17	37.7	0.21	0.22	0.33

Table 12
The results obtained by the proposed algorithm for (Case 2).

Description	X1	X2	X3	X4	X5	X6	X7	X8	X9	Structural weight (kg)	Door velocity (ms^{-1})	HIC
Baseline values	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.191	0.191	29.00	15.104	228
Solution for Case 2	0.48	0.621	0.51	0.53	1.5	1.43	1.30	0.341	0.190	20.42	14.20	212

Description	G1	G2	G3	G4	G5	G6	G7	G8	G9
Constraint values for solution	0.990	4.392	9.91	27.94	28.22	37.9	0.27	0.23	0.35

Table 13
SD values achieved by SSC and other competing algorithms.

Analysis	SSC	SHO	EPO	SOA	ESA	RSO
Mean	**0.0013500**	0.0561000	0.0042000	0.0023500	0.0934000	0.0248000
Median	**0.2550000**	0.7473000	6.3300000	0.3500000	0.7310000	0.4220000

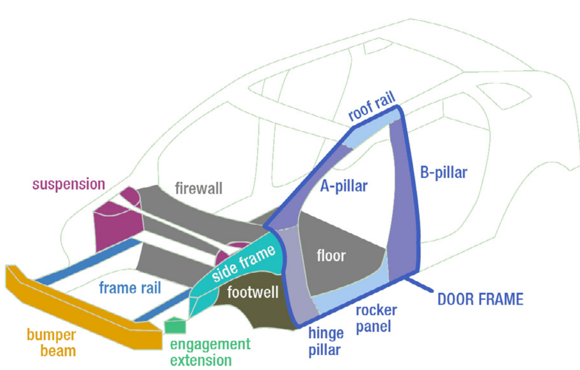

Fig. 9. Schematic view of car side impact problem.

The proximity of predicted to actual values as achieved. The predictive capacity of the generated response surfaces is established by simulation. The thickness design variables can be divided into the local and regional variables for two targets in . Results in Table 11 show that both goals increase marginally using the effective value of objective and restriction functions (equivalent to a solution obtained Point) in relation to the deterministic case. While the same objective functions are used when optimizing, in the second case (see Table 12), additional restrictions concerning the shift in objective functions are introduced within the specified neighborhood.

In addition, a well known efficiency metric for this problem is also validated for the proposed algorithm. Table 13 displays the standard deviation (*SD*) metric output values obtained by proposed and competition algorithms. The standard deviation (*SD*) value of *SSC* is lower than that of the others. The second and third best algorithms for optimization are the *SHO* and *EPO*. *SSC* can be seen as providing optimum results and very competitive solutions to the problem of crashing on the car side.

5. Conclusion

The Chimp Optimization Algorithm (*ChoA*) is a meta-heuristic approach which is broadly employed to address complex problems in various areas due to its low cost overhead computational simplicity and implementation. However, in the search domain, *ChoA* also faces various disadvantages such as ineffective exploitation and exploration balancing, premature convergence, complex multi-package search problems, particularly in the local optimal results that are easily trapped. A updated chimp optimization algorithm with sine–cosine functions and attacking strategy of *SHO* is proposed to solve the single-objective functions. In this paper, new location equations have been developed with sine–cosine functions to improve search agent convergence efficiency in the search domain. This algorithm can effectively control local search and convergence in the search process to the best global result.

During this comprehensive research effort, a large number of simulations were made to check the efficiency and correctness of the proposed *SSC*. Its efficiency has been checked by well-known algorithms and engineering problems. Experiential solutions have shown that *SSC* algorithm is superior to the state-of-the-art approaches. For average performance, the proposed *SSC* algorithm performance is 24%, 21%, 21%, 22%, 23.78%, 22%, 32.56%, 46% better than *SHO*, *TSA*, *EPO*, *SOA*, *RSO*, *ESA*, *DE*, AND *GA*, respectively. For standard deviation performance, the proposed *SSC* algorithm performance is 43%, 41%, 22.56%, 22%, 29%, 44%, 51%, 42% better than *SHO*, *TSA*, *EPO*, *SOA*, *RSO*, *ESA*, *DE*, AND *GA*, respectively. *SSC* can be applied in the future in the sense of dynamic real-life applications.

CRediT authorship contribution statement

Gaurav Dhiman: Wrote and simulated the whole paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The author would like to thanks to his parents for their divine blessings on him.

References

[1] P. Singh, G. Dhiman, A. Kaur, A quantum approach for time series data based on graph and Schrödinger equations methods, Modern Phys. Lett. A 33 (35) (2018).
[2] G. Dhiman, V. Kumar, KnRVEA: A hybrid evolutionary algorithm based on knee points and reference vector adaptation strategies for many-objective optimization, Appl. Intell. 49 (7) (2019) 2434–2460.
[3] G. Dhiman, P. Singh, H. Kaur, R. Maini, DHIMAN: A novel algorithm for economic dispatch problem based on optimization method using Monte Carlo simulation and astrophysics concepts, Modern Phys. Lett. A 34 (04) (2019).
[4] G. Dhiman, MOSHEPO: a hybrid multi-objective approach to solve economic load dispatch and micro grid problems, Appl. Intell. (2019) 1–19.
[5] P. Singh, G. Dhiman, S. Guo, R. Maini, H. Kaur, A. Kaur, H. Kaur, J. Singh, N. Singh, A hybrid fuzzy quantum time series and linear programming model: Special application on TAIEX index dataset, Modern Phys. Lett. A 34 (25) (2019).
[6] G. Dhiman, ESA: a hybrid bio-inspired metaheuristic optimization approach for engineering problems, Eng. Comput. (2019) 1–31.
[7] G. Dhiman, Multi-objective metaheuristic approaches for data clustering in engineering application (s), 2019.
[8] G. Dhiman, A. Kaur, Stoa: a bio-inspired based optimization algorithm for industrial engineering problems, Eng. Appl. Artif. Intell. 82 (2019) 148–174.
[9] S. Kaur, L.K. Awasthi, A. Sangal, G. Dhiman, Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization, Eng. Appl. Artif. Intell. 90 (2020) 103541.
[10] P. Singh, G. Dhiman, A fuzzy-LP approach in time series forecasting, in: International Conference on Pattern Recognition and Machine Intelligence, Springer, 2017, pp. 243–253.
[11] S. Verma, S. Kaur, G. Dhiman, A. Kaur, Design of a novel energy efficient routing framework for wireless nanosensor networks, in: 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), IEEE, 2018, pp. 532–536.
[12] A. Kaur, G. Dhiman, A review on search-based tools and techniques to identify bad code smells in object-oriented systems, in: Harmony Search and Nature Inspired Optimization Algorithms, Springer, 2019, pp. 909–921.
[13] G. Dhiman, A. Kaur, A hybrid algorithm based on particle swarm and spotted hyena optimizer for global optimization, in: Soft Computing for Problem Solving, Springer, 2019, pp. 599–615.
[14] G. Dhiman, V. Kumar, Spotted hyena optimizer for solving complex and non-linear constrained engineering problems, in: Harmony Search and Nature Inspired Optimization Algorithms, Springer, 2019, pp. 857–867.
[15] G. Dhiman, V. Kumar, Emperor penguin optimizer: A bio-inspired algorithm for engineering problems, Knowl.-Based Syst. 159 (2018) 20–50.
[16] M. Dehghani, Z. Montazeri, O.P. Malik, G. Dhiman, V. Kumar, BOSA: Binary orientation search algorithm, Int. J. Innov. Technol. Explor. Eng. 9 (2019) 5306–5310.
[17] R. Maini, G. Dhiman, Impacts of artificial intelligence on real-life problems, Int. J. Adv. Res. Innov. Ideas Educ. 4 (2018) 291–295.
[18] M. Garg, G. Dhiman, Deep convolution neural network approach for defect inspection of textured surfaces, J. Instit. Electron. Comput. 2 (2020) 28–38.
[19] G. Dhiman, M. Garg, A. Nagar, V. Kumar, M. Dehghani, A novel algorithm for global optimization: Rat swarm optimizer, J. Ambient Intell. Human. Comput. (2020) 1–26.
[20] M. Dehghani, Z. Montazeri, O. Malik, K. Al-Haddad, J.M. Guerrero, G. Dhiman, A new methodology called dice game optimizer for capacitor placement in distribution systems, Electr. Eng. Electromech. (1) (2020) 61–64.
[21] G. Dhiman, M. Garg, A. Nagar, V. Kumar, M. Dehghani, A novel algorithm for global optimization: Rat swarm optimizer, J. Ambient Intell. Human. Comput. (2020).
[22] M. Garg, G. Dhiman, A novel content based image retrieval approach for classification using GLCM features and texture fused LBP variants, Neural Comput. Appl. (2020).
[23] G. Dhiman, A. Kaur, HKN-RVEA: A novel many-objective evolutionary algorithm for car side impact bar crashworthiness problem, Int. J. Veh. Design (2020).
[24] G. Dhiman, M. Garg, MoSSE: A novel hybrid multi-objective meta-heuristic algorithm for engineering design problems, Soft Comput. (2020).
[25] M. Dehghani, Z. Montazeri, H. Givi, J.M. Guerrero, G. Dhiman, Darts game optimizer: A new optimization technique based on darts game, Int. J. Intell. Eng. Syst. (2020).
[26] R.K. Chandrawat, R. Kumar, B. Garg, G. Dhiman, S. Kumar, An analysis of modeling and optimization production cost through fuzzy linear programming problem with symmetric and right angle triangular fuzzy number, in: Proceedings of Sixth International Conference on Soft Computing for Problem Solving, Springer, 2017, pp. 197–211.
[27] P. Singh, G. Dhiman, A hybrid fuzzy time series forecasting model based on granular computing and bio-inspired optimization approaches, J. Comput. Sci. 27 (2018) 370–385, http://dx.doi.org/10.1016/j.jocs.2018.05.008, http://www.sciencedirect.com/science/article/pii/S1877750317300923.
[28] P. Singh, G. Dhiman, Uncertainty representation using fuzzy-entropy approach: Special application in remotely sensed high-resolution satellite images (RSHRSIs), Appl. Soft Comput. 72 (2018) 121–139, http://dx.doi.org/10.1016/j.asoc.2018.07.038, http://www.sciencedirect.com/science/article/pii/S1568494618304265.
[29] G. Dhiman, V. Kumar, Emperor penguin optimizer: A bio-inspired algorithm for engineering problems, Knowl.-Based Syst. 159 (2018) 20–50, http://dx.doi.org/10.1016/j.knosys.2018.06.001, http://www.sciencedirect.com/science/article/pii/S095070511830296X.
[30] G. Dhiman, V. Kumar, Multi-objective spotted hyena optimizer: a multi-objective optimization algorithm for engineering problems, Knowl.-Based Syst. 150 (2018) 175–197.
[31] P. Singh, K. Rabadiya, G. Dhiman, A four-way decision-making system for the Indian summer monsoon rainfall, Modern Phys. Lett. B 32 (25) (2018).
[32] E. Zakeri, S.A. Moezi, Y. Bazargan-Lari, A. Zare, Multi-tracker optimization algorithm: a general algorithm for solving engineering optimization problems, Iran. J. Sci. Technol. Trans. Mech. Eng. 41 (4) (2017) 315–341.
[33] C. Igel, M. Toussaint, A no-free-lunch theorem for non-uniform distributions of target functions, J. Math. Model. Algorithms 3 (4) (2005) 313–322.
[34] M. Khishe, M. Mosavi, Chimp optimization algorithm, in: Expert Systems with Applications, vol. 149, Elsevier, 2020, 113338.
[35] G. Dhiman, V. Kumar, Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications, Adv. Eng. Softw. 114 (2017) 48–70.
[36] E. Bonabeau, D.d.R.D.F. Marco, M. Dorigo, G. Théraulaz, G. Theraulaz, et al., Swarm Intelligence: from Natural to Artificial Systems, vol. 1, Oxford university press, 1999.
[37] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, vol. 4, IEEE, 1995, pp. 1942–1948.
[38] G. Dhiman, V. Kumar, Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications, Adv. Eng. Softw. 114 (2017) 48–70.
[39] Q. Zou, A. Li, X. He, X. Wang, Optimal operation of cascade hydropower stations based on chaos cultural sine cosine algorithm, in: IOP Conference Series: Materials Science and Engineering, vol. 366 (1), IOP Publishing, 2018, 012005.
[40] K.S. Reddy, L.K. Panwar, B. Panigrahi, R. Kumar, A new binary variant of sine–cosine algorithm: development and application to solve profit-based unit commitment problem, Arab. J. Sci. Eng. 43 (8) (2018) 4041–4056.
[41] S. Gholizadeh, R. Sojoudizadeh, Modified sine-cosine algorithm for sizing optimization of truss structures with discrete design variables, Iran Univ. Sci. Technol. 9 (2) (2019) 195–212.
[42] S.H.R. Pasandideh, S. Khalilpourazari, Sine Cosine crow search algorithm: a powerful hybrid meta heuristic for global optimization, 2018, arXiv preprint arXiv:1801.08485.
[43] M. El-Shorbagy, M. Farag, A. Mousa, I. El-Desoky, A hybridization of sine cosine algorithm with steady state genetic algorithm for engineering design problems, in: International Conference on Advanced Machine Learning Technologies and Applications, Springer, 2019, pp. 143–155.
[44] W. Guo, Y. Wang, F. Zhao, F. Dai, Riesz fractional derivative elite-guided sine cosine algorithm, Appl. Soft Comput. 81 (2019) 105481.

[45] S. Liu, Z.-K. Feng, W.-J. Niu, H.-R. Zhang, Z.-G. Song, Peak operation problem solving for hydropower reservoirs by elite-guide sine cosine algorithm with Gaussian local search and random mutation, Energies 12 (11) (2019) 2189.
[46] S.S. Alresheedi, S. Lu, M.A. Elaziz, A.A. Ewees, Improved multiobjective salp swarm optimization for virtual machine placement in cloud computing, Human-centric Comput. Inform. Sci. 9 (1) (2019) 15.
[47] C. Qu, Z. Zeng, J. Dai, Z. Yi, W. He, A modified Sine-cosine algorithm based on neighborhood search and greedy levy mutation, 2018,
[48] Q. Luo, J. Li, Y. Zhou, Spotted hyena optimizer with lateral inhibition for image matching, Multimedia Tools Appl. 78 (24) (2019) 34277–34296.
[49] H. Moayedi, D. Tien Bui, D. Anastasios, B. Kalantar, Spotted hyena optimizer and ant lion optimization in predicting the shear strength of soil, Appl. Sci. 9 (22) (2019) 4738.
[50] H. Jia, J. Li, W. Song, X. Peng, C. Lang, Y. Li, Spotted hyena optimization algorithm with simulated annealing for feature selection, IEEE Access 7 (2019) 71943–71962.
[51] S. Mirjalili, SCA: A sine cosine algorithm for solving optimization problems, Knowl.-Based Syst. 96 (2016) 120–133.
[52] G. Dhiman, V. Kumar, Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems, Knowl.-Based Syst. 165 (2019) 169–196.
[53] S. Kaur, L.K. Awasthi, A. Sangal, G. Dhiman, Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization, Eng. Appl. Artif. Intell. 90 (2020) 103541.
[54] V. Feoktistov, Differential Evolution, Springer, 2006.
[55] D. Whitley, A genetic algorithm tutorial, Statist. Comput. 4 (2) (1994) 65–85.
[56] N. Awad, M. Ali, J. Liang, B. Qu, P. Suganthan, Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization, Tech. Rep., 2016.
[57] A. Richardson, Nonparametric statistics for non-statisticians: A step-by-step approach by Gregory W. Corder, Dale I. Foreman, Internat. Statist. Rev. 78 (3) (2010) 451–452.
[58] M. Kaur, R. Kaur, N. Singh, G. Dhiman, Schoa: a newly fusion of sine and cosine with chimp optimization algorithm for HLS of datapaths in digital filters and engineering applications, Eng. Comput. (2021) 1–29.
[59] L. Gu, R. Yang, C.-H. Tho, M. Makowskit, O. Faruquet, Y.L. Y. Li, Optimisation and robustness for crashworthiness of side impact, Int. J. Veh. Design 26 (4) (2001) 348–360.