



SChoA: a newly fusion of sine and cosine with chimp optimization algorithm for HLS of datapaths in digital filters and engineering applications

Mandeep Kaur¹ · Ranjit Kaur¹ · Narinder Singh² · Gaurav Dhiman³

Received: 22 September 2020 / Accepted: 2 December 2020 / Published online: 7 January 2021
© The Author(s), under exclusive licence to Springer-Verlag London Ltd. part of Springer Nature 2021

Abstract

The Chimp optimization algorithm (ChoA) inspired by the individual intelligence and sexual motivation of chimps in their group hunting, which is separate from the another social predators. Generally, it is developed for trapping in local optima on the complex functions and alleviate the slow convergence speed. This algorithm has been widely applied to find the best optima solutions of complex global optimization tasks due to its simplicity and inexpensive computational overhead. Nevertheless, premature convergence is easily trapped in the local optimum solution during search process and is ineffective in balancing exploitation and exploration. In this paper, we have developed a modified novel nature inspired optimizer algorithm based on the sine–cosine functions; it is called as sine–cosine chimp optimization algorithm (SChoA). During this research, the sine–cosine functions have been applied to update the equations of chimps during the search process for reducing the several drawbacks of the ChoA algorithm such as slow convergence rate, locating local minima rather than global minima, and low balance amid exploitation and exploration. Experimental solutions based on 23-standard benchmark and 06 engineering functions such as welded beam, tension/compression spring, pressure vessel, multiple disk clutch brake, planetary gear train and digital filters design, etc. demonstrate the robustness, effectiveness, efficiency, and convergence speed of the proposed algorithm in comparison with others.

Keywords Chimp Optimization Algorithm (ChoA) · Meta-heuristics · Optimization · Single objective engineering optimization problems · Sine–cosine functions · Swarm intelligence

1 Introduction

In the last few decades, optimization is a hot topic in each and every field of research. Researchers are developing the best-to-best optimizer algorithms for complex issues of engineering, statistics, mathematics, computer science, medical fields, and many more [1–7]. In general, the optimization is identified as the process of selecting the best possible optima solutions for a given function/or problem globally or almost globally. Most issues in the real world can be seen as problems of optimization [8–12]. Day by day, the demand of every field faces several complex issues related to real-world optimization problems [13–16]. Frankly, the high performing algorithms are required for fulfilling the future complex issues. Therefore, the researchers are trying to develop highly effective optimization algorithms for these kinds of issues. So that these algorithms fulfill the future complex issue requirements. In general, the optimization method has been applied to find the accurate best solution

✉ Narinder Singh
narindersinghgoria@gmail.com

Mandeep Kaur
ermandeep0@gmail.com

Ranjit Kaur
ranjit24_ucoe@pbi.ac.in

Gaurav Dhiman
gdhiman0001@gmail.com

¹ Department of Electronics and Communication Engineering, Punjabi University, Patiala, India

² Department of Mathematics, Punjabi University, Patiala, India

³ Government Bikram College of Commerce, Patiala, India

of the optimization problems which are mostly deterministic that suffer from various drawbacks or major issues like that unbalanced exploitation or exploration, local optima, slow convergence, premature convergence, and the need to derivate search space [17].

The interest of the researcher of the different fields is growing in the nature inspired techniques in the last few years due to these drawbacks [18, 19]. Often, mainly of the real applications in dissimilar fields/or domain like that tuning of machine learning parameters, engineering, pattern recognition, clustering, bio-medical, digital filtering, computer science, statistics, applied mathematics, image processing, etc., are severely non-linear, linear, not continuous, so it includes complex constraints and many design variables. Furthermore, there is no guarantee of searching a best and possible optima solution in the search space as well as often failing to solve such kinds of complex functions [20]. Being these reasons, the real-world problems motivate the scientists to originate an effective and alternative highly robust optimization algorithm to find the solutions of these. In this regard, the various robust optimizer algorithms suffer from some drawbacks in finding the solution for complex functions, and hence, nature inspired algorithms [21, 22] is considered as the most widely used among the algorithms for solving the optimization problems recently.

With the advantage of the different nature inspired techniques, merge/modified two or various techniques while enhancing their constraints. Several algorithms have failed to prove better convergence performance without modified or hybridization to apply on complex function and enhance its competence. With the strategy of hybridization [23] or modification [24] of the algorithms could be fully improved the exploitation and exploration parts of the algorithm.

A hybrid SSA–PSO algorithm has been developed by Ibrahim et al. [25] to enhance SSA's exploration and exploitation capacity. The algorithm implemented makes apply of the attributes of the particle swarm optimization algorithm to enhanced the feasibility of the Salp swarm optimization algorithm in searching the best or possible optima solutions. Therefore, the performance of the convergence is increased.

Luis and Arribas [26] have proposed a new approach for the design of digital frequency selective FIR filters using an flowers pollination algorithm (FPA), with a novel multiple fitness function, to get optimized filter coefficients that best approximate ideal specifications.

Ibrahim et al. [27] introduced a segmentation approach using SSA to select the Perfect Threshold for multi-level image segmentation thresholds.

Saha et al. [28] applied Cat Swarm optimization (CSO) approach to determine the best optimal impulse response coefficients of FIR band stop filters, band pass, low pass, high pass, and trying to meet the respective ideal frequency response characteristics.

Liu et al. [29] developed a novel approach merging the salp swarm optimizer algorithm with a local search strategy to classify the passive location of the non-linear Time-Difference-Of-Arrival (TDOA) problem. Numerical and statistical results have proven that salp swarm optimizer has significantly enhanced position precision, fewer control parameters, and more reliable performance verified to particle swarm optimization.

In [30], the effectiveness of employing the swarm intelligence (SI) based and nature inspired algorithm is investigated for determining the optimal global best solutions to the Finite Impulse Response filter design problem.

[31] have been designed a new modified approach and framework for retiming the DSP blocks based on evolutionary computation processes. During this study, the authors have used various algorithms for fair comparison.

In [32], SSA is utilized to train Feed-Forward Neural Networks (FNNs). The planned strategy applies to various conventional classification and regression datasets. Results also appreciate the better and balanced exploitation–exploration properties of the SSA optimization algorithm, making the algorithm favorable for neural network training.

Sahu et al. [33] have been presented a modified version of sine–cosine algorithm-based fuzzy-aided PID controller which is called M-SCA algorithm. In this methodology, the SCA version by improving and updating few mathematical equations which is able of making the balance between exploitation and exploration phase of this version and enhancing the updating quality of generation. On the basis of experiential results has been showed the superior ability in fast convergence rate performance outperform than others.

However, the merits of Chimp optimization algorithm (ChoA) have satisfied the convergence performance, and it has a best balance amid exploitation and exploration. However, this version is not able to solve each type of complex problem during the search process. This algorithm also faces several drawbacks such as premature convergence, slow convergence rate, locating local minima rather than global minima, and low balance amid exploitation and exploration etc.

Considering the above, it is necessary to improve the ChoA algorithm to hide its limitations. Therefore, during this research, the proposed method is introduced from its original Chimp Optimization Algorithm by updating and improving the various mathematical equations by sine–cosine functions, which is able of enhancing the quality of optima outputs, updating quality of iteration, and is also capable to making the balance amid exploitation and exploration phases of the original ChoA algorithm. This concept is helpful in optimization, because they help to generate accurate solutions. The inclusion of these functions in optimization methods helps to increase the diversity of the solutions avoiding the local solutions.

This research introduces a modified version of Chimp Optimizer that combines the features of sine–cosine functions with original ChoA called SChoA algorithm. The idea is to enhance the search process of the original version of ChoA to find the optimal solutions. Different comparisons and experiments are performed between the version to select the one that provide the most accurate solutions. It is noted that, in terms of numerical results, statistical measurements, and convergence curves, the SChoA algorithm trapped the best optima solutions verified with the competitor versions. To sum up, the major contributions of this work are:

1. Sine–cosine functions are applied to tackle with the drawbacks of original ChoA algorithm to enhance its convergence performance.
2. SChoA is developed for solving complex single objective optimization problems.
3. The experimental solutions and convergence graphs are demonstrate the superiority of the proposed algorithm in solving single objective complex optimization problems.

The rest of the paper is organized as follows: In Sect. 2, the literature review is presented. In Sect. 3, preliminaries of various methods including the basic of Chimp Optimization Algorithm Salp swarm algorithm (SSA) are introduced. Section 4 explains the proposed SChoA algorithm. Section 5 discusses and analyzes the obtained experimental results. In Sect. 6, the proposed algorithm is tested on engineering problems. Finally, Sect. 7 provides conclusion and future work for this paper.

2 Literature review

In the literature, the meta-heuristic techniques are classified in different phases such as (1) evolutionary Algorithms (EA), but not SI, (2) swarm intelligence (SI), includes nature inspired optimizer algorithms that mimic the social behavior of groups of birds, human, animals and plants, (3) natural phenomenon algorithms (NP), imitates the chemistry and physical principles also, includes methods inspired by human behavior but neither SI nor EA [34].

For fulfilling the demand of the complex issues of different fields, the researchers are developing the robust modified or hybrid nature nature techniques [35], such as genetic algorithm (GA) [36], and particle swarm optimization (PSO) [37], Salp swarm optimization [38], sine cosine algorithm [39], Harris Hawks Optimization (HHO) [40], suggested chaotic SSA (CSSA) [41], binary version of the SSA named BSSA [42], and many more [43]. Furthermore, the recently hybrid and modified version has been reported.

Alresheedi et al. [44] combined the SSA and sine cosine algorithm (SCA) with the means of improving MOPS

techniques (MOSSASCA) to find a suitable solution for virtual machine location problem. The main purpose of the proposed MOSSASCA is to mitigate infringements of service quality, reduce power usage, and increase the average time before an agent termination, as well as minimize tension between the three goals. In SCA, a local search technique is followed to increase SSA's performance to increase convergence speed and avoid being trapped in a local optimum solution. In a set of experiments, there were different digital and physical devices to test the performance of the combined algorithm. Well-known methods of MOP were compared with MOSSASCA's findings. Results show a compromise between achieving the three goals.

In [45], a least-squares support vector machine (LS-SVM) model was suggested by Zhao et al. to predict carbon dioxide emissions (CO₂) using the principles of SVM. Compared to the other selected models, the proposed model showed more predictive performance. The numerical results showed the great superiority and ability of the SSA–LSSVM model to improve the accuracy and reliability of predicting emissions of CO₂.

SSA can approximate the optimum solution with high convergence, but the search for the optimum solution that affects the efficiency of the algorithm is not yet advantageous to SSA. To that this impact and boost its capacity and efficacy, Sayed et al. [41] suggested chaotic SSA (CSSA) by combining SSA with chaos algorithm. Chaos has recently been used a novel numerical approach to improve the execution of meta-heuristic approaches. Chaos is described as a simulation for non-linear [46] self-motivated conduct. Meta-heuristic method populations have the same advantages as scalable approach, simplicity, and reduced computational time. However, there are two intrinsic weaknesses in these approaches; low convergence rate and local optima recession.

Features selection (FS) problems with binary parameters are considering binary problems. Al-Zoubi and Mirjalili [47] introduced two versions of binary SSA. The SSA is transformed from continuous to binary in the first iteration using eight separate transfer functions (TFs). SSA is built into a crossover operator in the second version. In addition, the best search agent (leader) of SSA is updated to facilitate exploration using the crossover operator while maintaining the main mechanism of this algorithm.

Qu et al. [48] have been proposed improved SCA algorithm by Greedy Levy mutation. This algorithm adopts the method of linear decreasing inertia weight to better balance the global searching and both exponential decreasing conversion parameter and local development ability of the algorithm. On the basis of simulation results prove that existing method can it has faster convergence speed and effectively avoid falling into the local optimum, and higher optimization accuracy.

While the original study of Salp algorithm was proposed to solve continuous problems, and it cannot be applied to binary problems directly, Rizk and Hassanien [42] proposed a new binary version of the SSA named BSSA to enhance the exploration and exploitation capabilities by modifying Arctan transformation. This modification has two features regarding the transfer function, namely multiplicity and mobility. The proposed BSSA is compared among four variants of transfer functions for solving global optimization problems. Furthermore, the non parametric statistical test based on Wilcoxon’s rank-sum is carried out to judge statistically the significant of the obtained results among the different algorithms. The results affirm the superior performance of the modified BSSA variant over the other variants as well as the existing approaches regarding solution quality.

A novel method based on the new moth flame optimizer technique has been developed by Mohammad Reza Esmaeili et al. [49]. After applying this method in synthesizing digital filters, it was observed that compared to the genetic algorithm-based and the Particle Swarm Optimization-based techniques, it exhibited a higher ability to reach an optimal solution under the same initial conditions (the initial population and the same number of iteration)

Gholizadeh and Sojoudizadeh [50] have developed a modified version of the SCA algorithm known as Modified SCA algorithm (MSCA). The proposed strategy combines with two computational strategies during its search process, such that, first, a kind of elitism is utilized by substituting a number of worst solutions of the current population with some variants of the global best optima solution, and second, a mutation operation is performed to increase the probability of searching the best global optimum solution. Experimental results showed that the proposed algorithm is able to give the highly accurate or fast convergence results than others.

3 Chimp optimization algorithm (ChoA)

A new nature inspired algorithm known as Chimp optimization algorithm (ChoA) has been developed by Khishe et al. [51]. This strategy is inspired by individual intelligence and sexual motivation of chimps in their group hunting. It is different from the other social predators. During this methodology, four different phases have been used for simulating diverse intelligence such as attacker, barrier, chaser, and driver, etc.

The mathematical model of this proposed algorithm has been described as following step by step:

The driving and chasing the target or prey is illustrated by the following mathematical equations (Eqs. ((1)–(2)):

$$D = \left| c \cdot a_{\text{prey}}(n) - m a_{\text{chimp}}(n) \right| \tag{1}$$

$$a_{\text{chimp}}(n + 1) = a_{\text{prey}} - a \cdot d, \tag{2}$$

where n is represented the total number of iterations, and c , m , and a are the coefficient vectors. The coefficients c , m , and a are calculated by Eqs. [(3)–(4)]:

$$a = 2 \cdot l \cdot r_1 - l \tag{3}$$

$$c = 2 \cdot r_2 \tag{4}$$

$$m = \text{chotic}_{\text{value}}, \tag{5}$$

where r_1 and r_2 are random values in range of [0, 1], m is represented the chotic vector, and l is reduced non-linearly from 2.5 to 0 through the iteration process.

During this step has been implemented the behavior of the chimps mathematically. Here, it assumes that the initial solution is available by the attacker, driver, barrier, and chaser which are better informed about the location of the target. In the next iteration, four of the other optima solutions yet obtained are stored and the rest of the chimps are forced to update their own location according to the best chimp locations. This process have been illustrated by the following mathematical Eqs. (6)–(9):

$$d_{\text{attacker}} = \left| c_1 a_{\text{attacker}} - m_1 \cdot x \right| \tag{6}$$

$$d_{\text{barrier}} = \left| c_2 a_{\text{barrier}} - m_2 \cdot x \right| \tag{7}$$

$$d_{\text{chaser}} = \left| c_3 a_{\text{chaser}} - m_3 \cdot x \right| \tag{8}$$

$$d_{\text{driver}} = \left| c_4 a_{\text{driver}} - m_4 \cdot x \right|. \tag{9}$$

When the random vectors are lie between the range of [−1, 1], then the next location of a chimp can be in any location amid its present location and the location of the target or prey:

$$x_1 = a_{\text{attacker}} - a_1 \cdot d_{\text{attacker}} \tag{10}$$

$$x_2 = a_{\text{barrier}} - a_2 \cdot d_{\text{barrier}} \tag{11}$$

$$x_3 = a_{\text{chaser}} - a_3 \cdot d_{\text{chaser}} \tag{12}$$

$$x_4 = a_{\text{driver}} - a_4 \cdot d_{\text{driver}}. \tag{13}$$

From the overall equations, the position of the chimps during the search process is updated by the following mathematical Eq. (14):

$$x_{n+1} = \frac{x_1 + x_2 + x_3 + x_4}{4}. \tag{14}$$

Finally, for the update of the location of the chimps during the search process in the search domain has been applied to the following mathematical equation 15:

$$a_{\text{chimp}}(n + 1) = \begin{cases} a_{\text{prey}}(n) - x.d, & \text{if } \phi < 0.5 \\ \text{chaotic}_{\text{value}} & \text{if } \phi > 0.5. \end{cases} \quad (15)$$

3.1 3.0.1 Pseudocode of ChoA

The pseudocode of CHoA algorithm is reported in Algorithm 1.

each optimization method could not be able to shows the best solution of all type of complex problems. All algorithms may be faced some drawbacks, so due to these weakness these could failure to find the solution of complex functions.

Therefore, due to the present competitive situations, we are needed most powerful optimization techniques, so that these could tackle complex problems easily. However, these techniques could be tackle complex functions easily if the exploration and exploitation phase of these techniques will be too strong.

After learning the initial phase of chimp optimization algorithm, here, we are trying to present a newly modified

Algorithm 1 Pseudo-code of ChoA algorithm

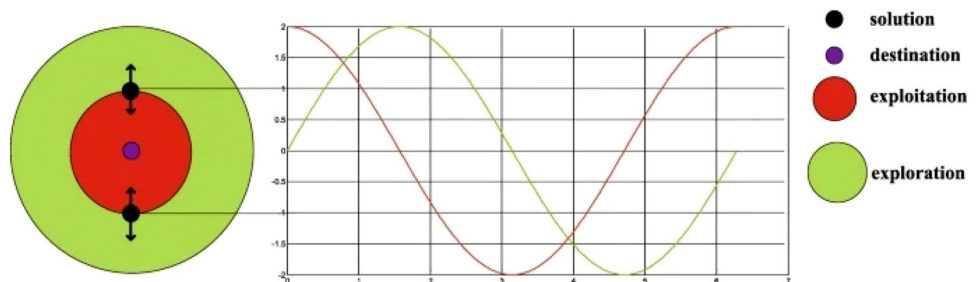
Inputs: The population size N and maximum number of iterations t
 Initialize the random population $X_i(i = 1, 2, \dots, N)$
while $t < \text{max.number of iterations}$ **do**
 for each chimps **do**
 Define the chimp’s group
 By using is group strategy to update
 end for
 for each search climb **do**
 if $x < 1$ **then**
 Update the position of the current search chimp
 else if $x > 1$ **then**
 Select a random search chimp
 end if
 Update the position of the current search chimp
 end for
 Update X Attacker, Barrier, Driver and Chaser
 $t + 1$
end while

4 The proposed sine–cosine Chimp optimization algorithm (SChoA)

The complex optimization application is challenges for the optimization meta-heuristics. According to the literature,

algorithm with the help of sine–sine function functions for the complex optimization functions. The sine and cosine function are helped by the algorithm in fluctuating outward or toward finding the optimal solutions. These functions help to ignore the local optima and force the algorithm for

Fig. 1 The performance graph of sine and cosine function



trapping the global optima fastly. The performance of the sine and cosine functions in the search space is illustrated by Fig. 1.

Figure 1 illustrates the present solution which will be away from the target and explore the search domain when the search member moves in $[-2, -1]$ or $(1, 2]$. Additionally, the present solution will be toward the target and exploit the search domain when the search member moves in $[-1, 1]$.

However, the ChoA algorithm is capable to solve various complex optimization functions; nevertheless, it face various drawbacks for trapping the local, global, and explore optima values in the search domain. It could not be fitted for highly complex optimization problems and cannot tackle their several drawbacks such as slow diversity, premature convergence, slow convergence speed, etc. For improving exploitation phase of the standard chimp optimization algorithm, we are trying to incorporate the sine–cosine functions in the position update equations of the ChoA algorithm, and to improve the weaknesses of standard ChoA version.

In the search domain for identify the accurate and best global optimum for complex optimization functions have been applied this modified algorithm. The chimp optimization algorithm (ChoA) phases operate in the directions of exploration of the optima solution and modified phases have been applied for exploitation of the optima in the search domain. The modification has helped for getting the best global outputs fastly and ignoring the local optima. Due to the impact of this, we have enhanced the convergence rate of the standard ChoA algorithm through this modification.

Furthermore, the mathematical formulation of the SChoA algorithm has been illustrated by the following steps:

- *Constants* During the runs of the Matlab code of the algorithms have applied the various parameter settings such as search agents is 30, total number of iterations are 500, and dimension as per Table 15.
- *Initialization* In this stage, first, we are initializing the population of the search agents and these are initialized randomly as per the given functions. Where the algorithms assigns a random vector of n dimensional for the i th chimp; $X_i(i = 1, 2, \dots, n)$.
- *Evaluation* In this stage, the search members of the population are evaluated according to the superiority of the own position in the search space during the search process. After that, the fitness value of each search member is tested through the given objective function and each search member of the crowd during the search process is used these fitness values for locating the new positions in the search space.

- *Exploration phase* In this state, the chimps are update their position for the purpose of searching next nearest position of his prey. These position are updated through Eqs. 1–5.
- *Exploitation phase* In this phase, the search agent position has been evaluated through modified mathematical equations (Eqs. 16–20). These equations have played an important role for enhancing the exploitation phase for chimps for trapping the best optima values in the search domain during the search process and ignoring the local optima solutions. With the help of this modification chimps could be easily trapped in the optima solutions in the complex space in at least a number of iterations. Additionally, it also helps in reducing the computational time in searching the next best position for the search agents:

$$r_2 = (2\pi) \times rand \tag{16}$$

$$x_1 = a_{\text{attacker}} - \cos(r_2) \times a_1 \cdot d_{\text{attacker}} \tag{17}$$

$$x_2 = a_{\text{barrier}} - \sin(r_2) \times a_2 \cdot d_{\text{barrier}} \tag{18}$$

$$x_3 = a_{\text{chaser}} - \cos(r_2) \times a_3 \cdot d_{\text{chaser}} \tag{19}$$

$$x_4 = a_{\text{driver}} - \sin(r_2) \times a_4 \cdot d_{\text{driver}}, \tag{20}$$

where r_2 represents the rand number lies between 0 and 1, x_1, x_2, x_3, x_4 are update positions of the chimps, and a_1, a_2, a_3, a_4 are coefficient vectors, respectively.

- *Leader Chimp position* The leader chimp position updated through Eq. 14.
- *Stopping conditions* Finally, the stopping conditions have been implemented for searching the best possible optima value in the search domain. These conditions have been applied for evaluating the assessment process of the all search agents of the group and replacing the position according to best search agent’s positions. It is repeated again and again until it satisfies the criteria of prevention for example it reaches the highest maximum iterations or the solution is earliest found.

4.1 4.0.1 Pseudocode of modified SChoA algorithm

The pseudocode of SChoA algorithm is reported in Algorithm 2.

Algorithm 2 Pseudo-code of SChoA algorithm

Inputs: The population size N and maximum number of iterations t
Initialize population $x_i (i = 1, 2, \dots, N)$
while $t < \text{max.number of iterations}$ **do**
 for each chimps **do**
 Define the chimp's group
 By using is group strategy to update m, c and l
 Use m, c, l for calculate a and then d
 end for
 for each search climb **do**
 if $\phi < 0.5$ **then**
 if $|x| < 1$ **then**
 Update the position of the current search chimp by Eq.(2)
 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 chimp by the Eq.(15)
 end if
 end for
 update m, c, l and a
 Update attacker, chaser, barrier and driver by Eq's. (16)-(20)
 $t + 1$
 end while
 Return $x_{attacker}$

5 Complexity

The complexity of the proposed method is illustrated in the two different following forms;

5.1 Time complexity of SChoA

The computational complexity of the proposed algorithm (SChoA) can be represented as:

$$R(v(M(sd + cs + ms^2))), \quad (21)$$

where v , s , d , c , m , and L illustrate the number of runs, number of solutions, dimension, cost of objective function, number of objective function, and number of iterations.

5.2 Search space complexity of SChoA

The search space complexity of SChoA algorithm in terms of the memory space depends on the constants or parameters of both the dimension of the given function and the search members. This identifies the amount of space that proposed method needs during the initialization process. Therefore, the search space complexity of proposed method can be illustrated by the mathematical equation as follows:

$$R(sd). \quad (22)$$

Here, the proposed algorithm also requires additional search space for other constants. However, this extra search space is not critical, and hence, the search space complexity of the proposed algorithm remains within the range of above mathematical equation (Eq. (22)).

6 Results and discussion

To evaluate the convergence and accuracy performance of the proposed SChoA algorithm, a experiential analysis has been performed in this section. For comparison, various robust population-based algorithms, such as MPSO, SCA, PSO, TACPSO, and Chimp, have been used.

6.1 Parameter setting and standard benchmark functions

The Matlab 2015a software has been applied for the coding of all algorithms for verifying the performance with each

Table 1 The optimal solutions obtained on 23-benchmark functions by the SChoA algorithm and the competitors

F	Algorithm	Best score	Best max. value	Average	S.D.
F1	MPSO	6.2999	6.6617e+04	1.1797e+04	1.5447e+04
	SCA	0.0087	3.8468e+04	1.4023e+04	1.7417e+04
	PSO	1.5989	6.3536e+04	1.5924e+04	5.8033e+03
	TACPSO	0.1459	7.2293e+04	1.8818e+03	6.0679e+03
	Chimp	3.3939e−05	7.1812e+04	3.6421e+04	3.1559e+04
	SChoA	5.6626e−33	7.7909e+04	1.5717e+03	5.6803e+03
F1	MPSO	49.4796	3.0056e+12	6.0384e+09	1.3442e+11
	SCA	0.0017266	3.1649e+12	6.9533e+09	1.4219e+11
	PSO	4.1197	2.3624e+12	4.7248e+09	1.0565e+11
	TACPSO	0.0968	4.7624e+11	1.0848e+09	2.1463e+10
	Chimp	2.1407e−06	3.3150e+13	7.3674e+12	1.3275e+13
	SChoA	1.7204e−20	3.6525e+13	1.0220e+09	1.9135e+10
F3	MPSO	2.6593e+04	1.3615e+05	4.5715e+04	2.2307e+04
	SCA	13531.6371	1.7753e+05	5.2755e+04	3.9729e+04
	PSO	281.3240	6.5932e+04	2.5335e+03	7.3345e+04
	TACPSO	814.3129	1.4726e+05	8.2458e+03	2.3106e+04
	Chimp	13.3838	1.6046e+05	9.0156e+04	7.3125e+04
	SChoA	6.1983e−08	1.9481e+05	1.0592e+03	2.2571e+04
F4	MPSO	10.4225	86.5907	41.5662	27.8297
	SCA	54.8864	89.0673	78.0030	17.7977
	PSO	2.4063	82.2175	6.3489	9.2631
	TACPSO	8.2275	86.5125	24.0546	14.2789
	Chimp	0.1265	45.0615	53.3266	37.1912
	SChoA	2.7577e−10	88.6998	12.1931	1.2680
F5	MPSO	138.8381	1.8253e+08	2.4462e+07	3.8299e+07
	SCA	2730.7454	3.1997e+08	6.9233e+07	1.1339e+08
	PSO	383.9547	1.9914e+08	8.6423e+05	1.0530e+07
	TACPSO	132.8287	2.4246e+08	2.20209e+06	1.3789e+07
	Chimp	28.7388	3.4148e+08	1.5709e+08	1.4146e+08
	SChoA	28.5806	3.6247e+08	5.9587e+05	1.007e+07
F6	MPSO	33.4766	7.9077e+04	1.6596e+04	2.1433e+04
	SCA	34.762	8.3331e+04	2.9954e+04	3.7143e+04
	PSO	3.1046	7.1844e+04	1.6113e+03	6.5984e+03
	TACPSO	130.7833	7.5853e+04	3.0128e+03	6.8110e+03
	Chimp	4.5517	6.8753e+04	3.8289e+04	3.2264e+04
	SChoA	3.0114	8.3694e+04	1.0111e+03	5.6249e+03
F7	MPSO	0.1142	61.2874	6.5500	9.4360
	SCA	0.073601	112.4585	41.3224	51.3600
	PSO	5.2057	75.4536	48.4544	21.9777
	TACPSO	0.0458	122.5885	1.5274	8.3476
	Chimp	0.076766	120.8919	60.3297	55.4527
	SChoA	0.0010	160.0359	1.4378	5.7739
F8	MPSO	− 8.9951e+03	− 3.0052e+03	− 8.2104e+03	1.4418e+03
	SCA	− 3.7518e+03	0	− 3.5110e+03	246.9999
	PSO	− 7.0215e+03	− 2.9755e+03	− 5.5697e+03	1.7085e+03
	TACPSO	− 9.2891e+03	− 1.5318e+03	− 7.4832e+03	2.4864e+03
	Chimp	− 5.7312e+03	− 2.8322e+03	− 5.7088e+03	188.8485
	SChoA	− 9.8671e+03	0	− 3.8115e+03	180.0194

Table 1 (continued)

F	Algorithm	Best score	Best max. value	Average	S.D.
F9	MPSO	166.3000	457.7561	238.6882	79.6640
	SCA	6.7832	446.6224	164.6431	174.8304
	PSO	142.1316	471.1388	264.5869	82.7104
	TACPSO	77.5040	460.2153	151.2479	80.9861
	Chimp	23.6326	441.4657	287.2308	179.1193
	SChoA	0	475.4462	43.9680	77.7203
F10	MPSO	2.6987	20.5399	10.3732	7.2253
	SCA	15.6986	20.6395	19.4357	1.8108
	PSO	2.5733	20.0753	5.7380	2.4067
	TACPSO	4.2307	20.6507	6.5348	3.6142
	Chimp	19.9597	20.5567	19.9626	0.0333
	SChoA	1.5099e−14	20.9545	1.8949	1.7160
F11	MPSO	0.6988	653.3792	102.4597	134.2159
	SCA	1.0643	552.4245	141.6881	225.4360
	PSO	0.0985	607.9306	44.4302	120.1320
	TACPSO	0.0634	624.8712	15.0644	48.4517
	Chimp	3.5813e−07	629.2019	329.8227	285.5492
	SChoA	0	660.3577	20.9825	83.0538
F12	MPSO	59.8943	6.5263e+08	4.9864e+07	8.4607e+07
	SCA	13032.133	7.8927e+08	3.5222e+08	3.5999e+08
	PSO	0.3456	6.0060e+08	2.9000e+06	3.6721e+07
	TACPSO	4.1876	5.3468e+08	5.3068e+06	3.4991e+07
	Chimp	2.0981	9.1085e+08	5.1711e+08	4.0341e+08
	SChoA	0.1626	9.4327e+08	2.5932e+06	3.3113e+07
F13	MPSO	7.7595	1.2000e+09	5.0545e+07	1.0305e+08
	SCA	12411.2108	1.0777e+09	4.2834e+08	4.9404e+08
	PSO	0.7689	1.1876e+09	5.0020e+06	6.4531e+07
	TACPSO	3.7814	1.4184e+09	6.4690e+07	8.0294e+07
	Chimp	2.7450	1.3221e+09	6.6209e+08	6.0195e+08
	SChoA	0.67679	1.6739e+09	1.6103e+07	5.1686e+07
F14	MPSO	0.9980	14.7165	1.1510	1.0258
	SCA	0.99804	137.6850	3.0215	10.7374
	PSO	1.9920	21.7810	2.3615	1.2611
	TACPSO	0.9980	284.7727	2.8733	17.9660
	Chimp	0.99801	247.5210	3.9429	20.2848
	SChoA	0.998	311.9313	1.3330	10.4292
F15	MPSO	0.0204	0.4371	0.0214	0.0189
	SCA	0.0013794	0.0593	0.0022	0.0032
	PSO	8.8218e−04	0.1171	0.0020	0.0069
	TACPSO	7.8266e−04	0.3714	0.0033	0.0266
	Chimp	0.0013	0.0764	0.0028	0.0095
	SChoA	6.8388e−04	0.4462	0.0020	0.0017
F16	MPSO	− 1.0316	10.3537	− 0.9861	0.5221
	SCA	− 1.0316	33.8446	− 0.9027	1.5922
	PSO	− 1.0316	25.6828	− 0.9121	1.2668
	TACPSO	− 1.0316	2.1536	− 1.0076	0.2272
	Chimp	− 1.0316	10.5347	− 0.1030	2.9730
	SChoA	− 1.0316	36.4936	− 1.0088	0.2212

Table 1 (continued)

F	Algorithm	Best score	Best max. value	Average	S.D.
F17	MPSO	0.3979	0.4363	0.4987	0.0050
	SCA	0.40003	1.2605	0.4393	0.1469
	PSO	0.3979	1.3905	0.4015	0.0456
	TACPSO	0.3979	2.0111	0.4007	0.0258
	Chimp	0.39794	0.7576	0.4181	0.0801
	SChoA	0.3979	2.1936	0.4007	0.0049
F18	MPSO	3.0000	135.7076	3.4676	6.1680
	SCA	3.0000	27.9057	3.1789	1.6783
	PSO	3.0000	26.8088	3.0669	1.0733
	TACPSO	3.0000	16.5713	3.0690	0.6871
	Chimp	3.0002	131.3970	4.6893	11.6675
	SChoA	3.0000	171.1468	3.0000	0.1925
F19	MPSO	− 3.8628	− 3.4964	− 3.8587	0.0256
	SCA	− 3.8527	0	− 3.8059	0.1828
	PSO	− 3.8628	− 2.8312	− 3.8575	0.0502
	TACPSO	− 3.8628	− 3.7018	− 3.8588	0.0213
	Chimp	− 3.8534	− 2.9339	− 3.8183	0.1347
	SChoA	− 3.8649	0	− 3.8410	0.0916
F20	MPSO	− 3.2031	− 2.5071	− 3.1871	0.1694
	SCA	− 2.9821	0	− 2.8162	0.2699
	PSO	− 3.2031	− 1.8479	− 3.1217	0.1257
	TACPSO	− 3.2031	− 1.3054	− 3.1663	0.1675
	Chimp	− 3.0135	− 2.5059	− 2.9688	0.0913
	SChoA	− 3.3219	0	− 3.2537	0.1243
F21	MPSO	− 5.1008	− 0.2814	− 4.8030	0.8142
	SCA	− 5.4441	0	− 1.9466	1.6559
	PSO	− 2.6829	− 0.3367	− 2.4692	0.4574
	TACPSO	− 2.6305	− 0.6105	− 2.5814	0.2104
	Chimp	− 0.8806	− 0.2929	− 0.8594	0.0648
	SChoA	− 10.0944	0	− 5.0904	0.3433
F22	MPSO	− 2.7659	− 0.4677	− 2.6932	0.2668
	SCA	− 4.3615	0	− 2.9022	1.2943
	PSO	− 5.0877	− 0.5200	− 4.5385	0.8716
	TACPSO	− 2.7519	− 0.3959	− 2.6837	0.3173
	Chimp	− 0.9070	− 0.4569	− 0.8668	0.0980
	SChoA	− 10.3082	− 0.5907	− 5.1439	0.0574
F23	MPSO	− 2.8711	− 0.3416	− 2.6117	0.5576
	SCA	− 2.4286	0	− 1.5989	0.6660
	PSO	− 3.8354	− 0.5958	− 3.3157	0.8183
	TACPSO	− 5.1756	− 0.4448	− 4.0242	0.6726
	Chimp	− 0.9475	− 0.3232	− 0.8982	0.1091
	SChoA	− 10.5139	0	− 4.1068	0.0495

others. In this implementation, have been applied different parameters settings, like total number of iterations 500, search chimps 30, dim, and upper and lower bounds as per as standard functions, respectively.

It is always beneficial to use a set of standard testbeds with different characteristics to conveniently and confidently

test the performance of any new optimization algorithm on different standard benchmark problems and compare it with other nature inspired algorithms. The diversity of test problems or functions allows observing and testing the ability of any new optimizer algorithm from different perspectives.

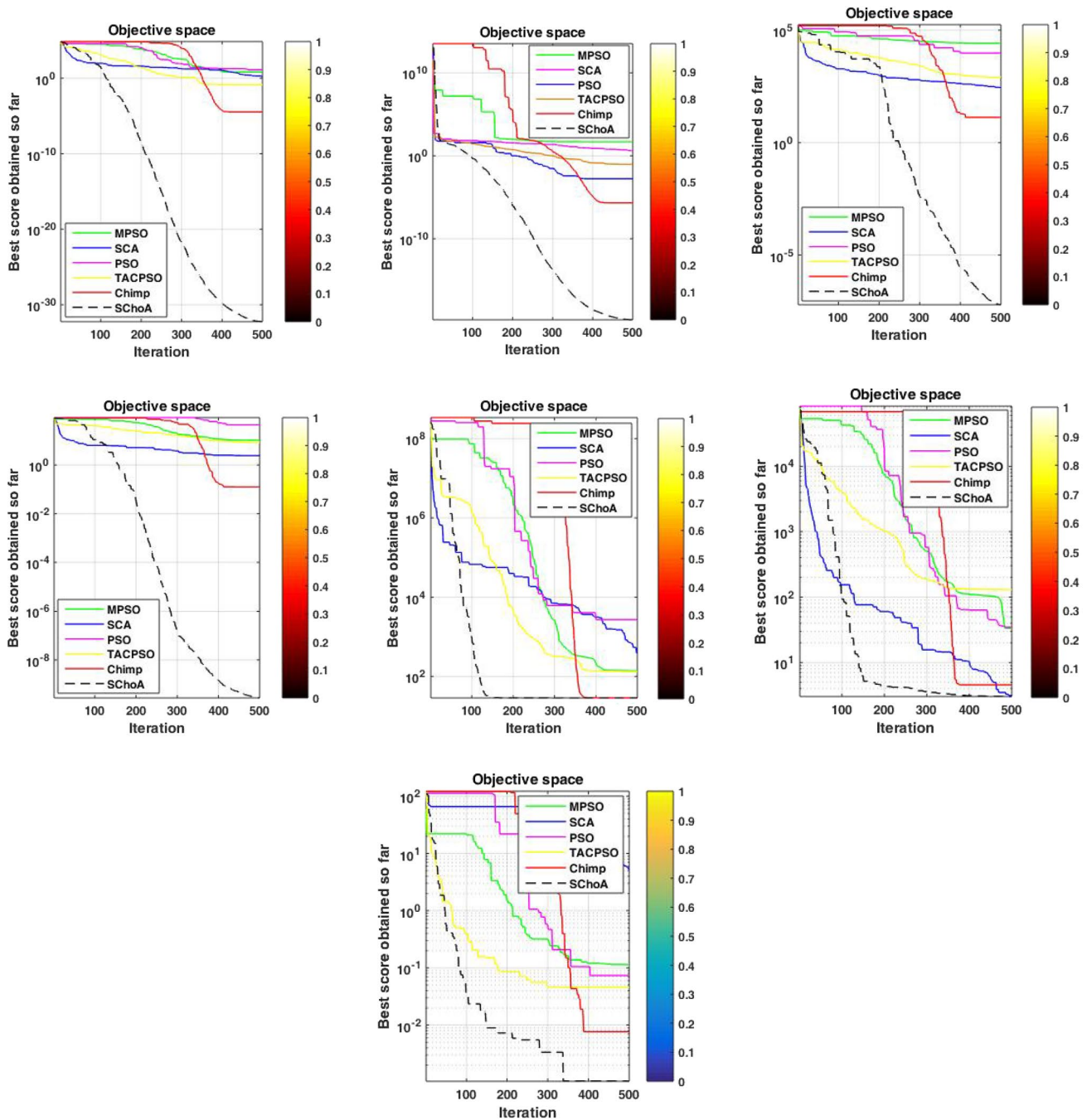


Fig. 2 Convergence graphs of algorithms on uni-modal functions

A set of 23-standard functions are applied to demonstrate the performance and efficiency of the SChOA algorithm compared to other existing optimization techniques in the literature. These functions can be divided into three different phases: uni-modal [52], multi-modal [53], and fixed dimension multi-modal [52] functions, respectively. The mathematical details of these functions have been illustrated in “Appendix”. A detailed description of the characteristics of the unimodal (F_1 – F_7), multi-modal (F_8 – F_{13}), and fixed

dimension multi-modal functions (F_{14} – F_{23}) are reported in Table 15 in “Appendix”.

The unimodal problems have only one global optimum with no local optima. These problems are highly suitable to explore the exploitative efficiency and convergence behavior of the new proposed optimization algorithm. Multimodal and fixed-dimension multimodal benchmark test problems face the existence of several local optimum solutions and more than one global optimum.

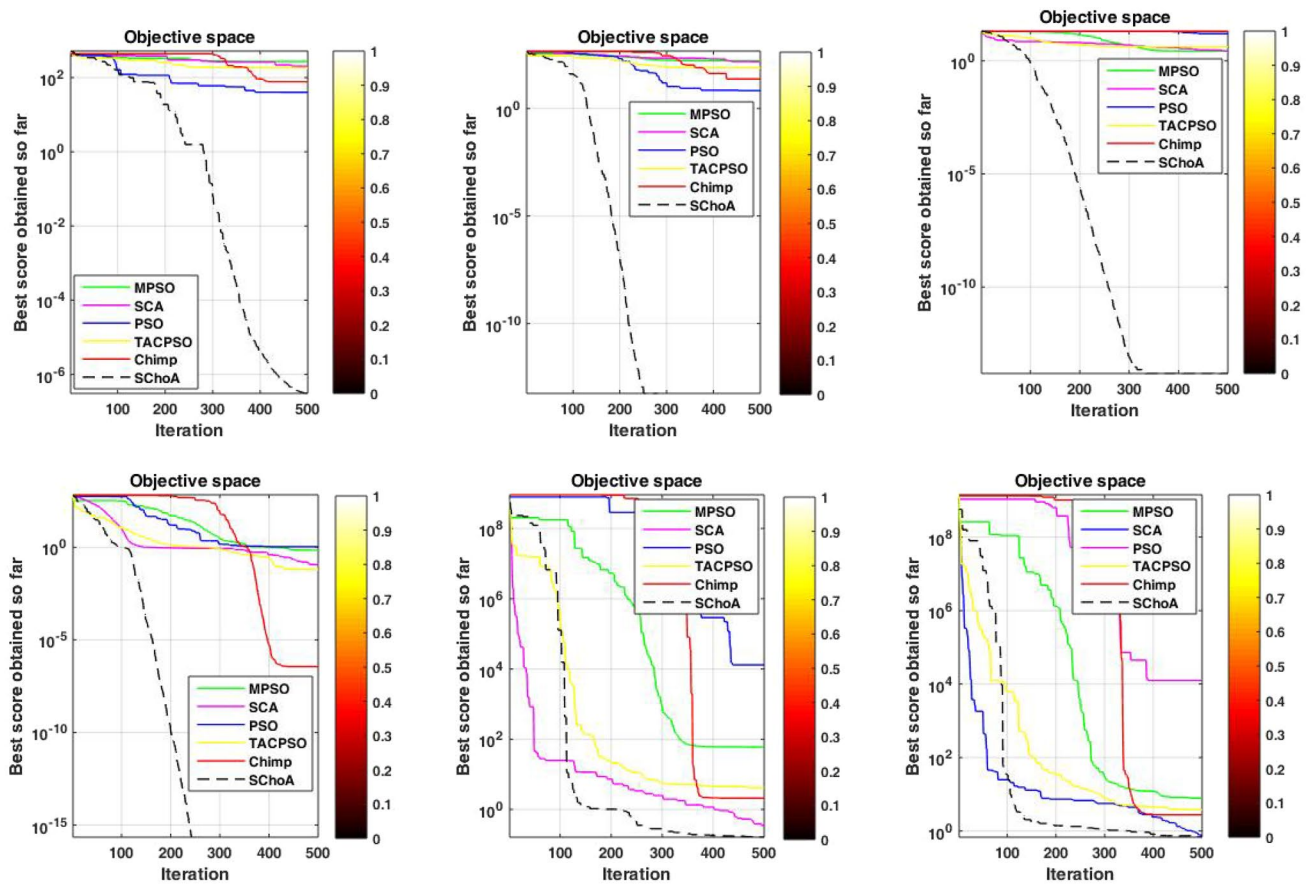


Fig. 3 Convergence graphs of algorithms on multi-modal functions

The landscapes of these benchmark test functions have been showed in 2-D versions through figures, respectively.

To verify the accuracy of the proposed SChOA algorithm have been tested on these 23-test functions as comparison with the other recent algorithms: MPSO, PSO, TACPSO, Chimp, etc. Furthermore, the more analysis of the results of algorithms on these functions has been illustrated through the following sections.

6.2 Experiment and comparison

The simulation must be required to done multiple *n*-times to evaluate stable numerical and statistical efforts for the performance of the optimizer methods. And for evaluating the stability of the algorithms, every run must be carried out to *m* number of generations or iterations [54]. In this work, has been applied the same experimental procedure for testing the performance of proposed method. And the accuracy of the results obtained by algorithms has been verified through different categories such as least best minima value, max best value, standard deviation, and mean. Experimental solutions of the algorithms are reported in Table 1, and convergence graphs are illustrated by Figs. 2, 3 and 4.

Here, the accuracy and performance of the SChOA algorithm has been compared with recent algorithms MPSO, PSO, TACPSO, Chimp, etc. in terms of best numerical and statistical outputs.

6.3 Discussion

The standard deviation (SD) and mean measures were used to verify the result output performance of SChOA to those of the Standard ChoA or Chimp and other nature inspired techniques on 23-standard test benchmark problems. All algorithms has run 30 independent times with at least 500 number of iterations on every test suite, to evaluate meaningful solutions. The statistical solutions are computed at the last iteration of each nature inspired approach on each test suite to get the best optimal result and give a highly fair comparison among all comparative nature inspired algorithms.

Furthermore, in the following sections have been illustrated the comparison of proposed algorithm with other algorithms in three different phases; evaluation of uni-modal, multi-modal, and fixed dimension multi-modal test suites.

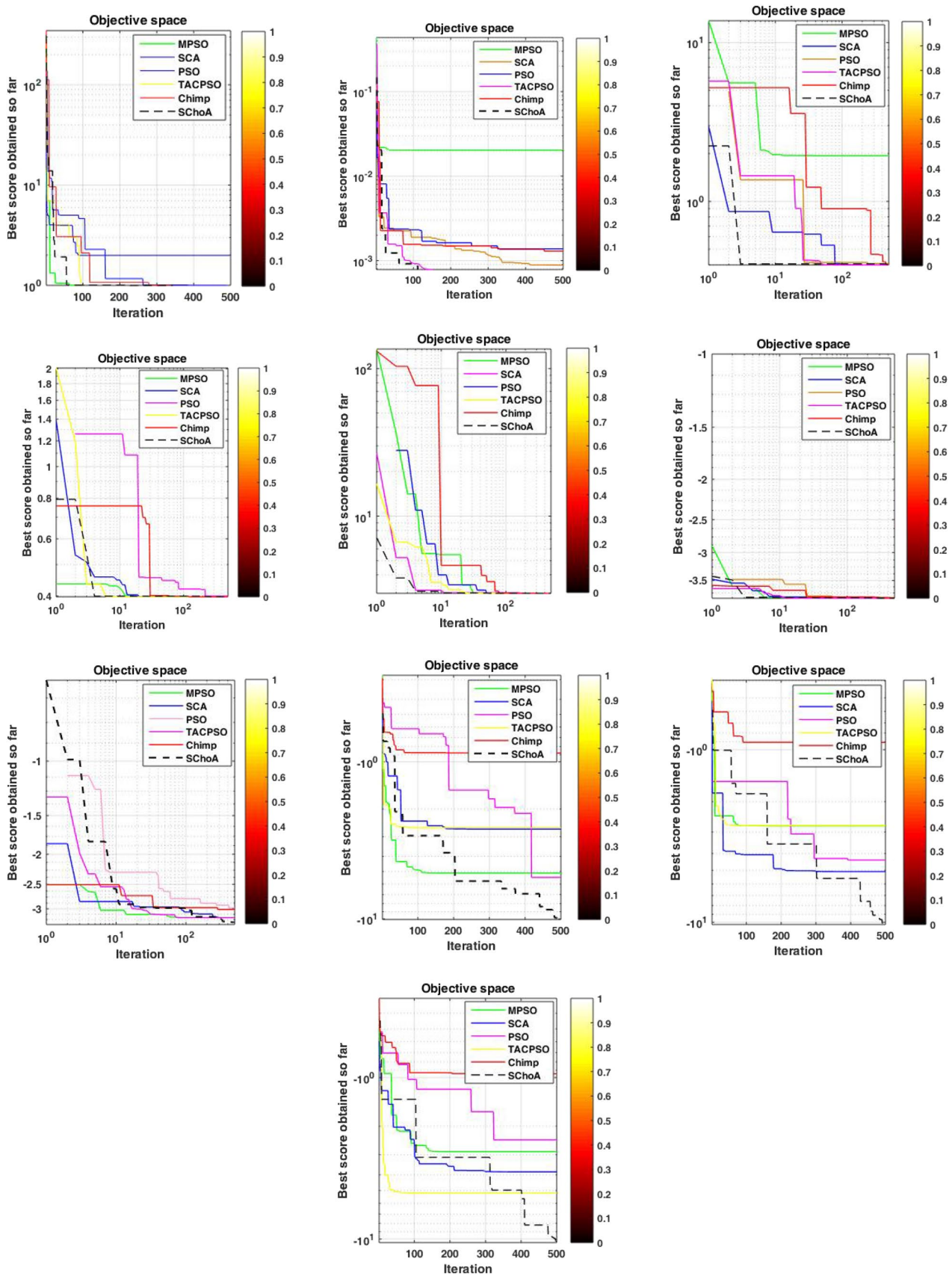
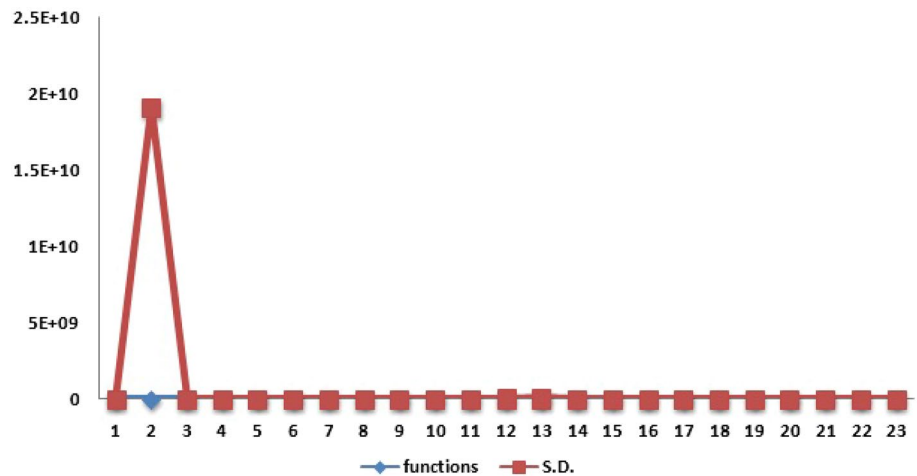


Fig. 4 Convergence graphs of algorithms on fixed dimension multi-modal functions

Table 2 Comparison of average values of algorithms on 23-benchmark functions

Functions	MPSO	SCA	PSO	TACPSO	Chimp	SChoA
F1	W	W	W	W	W	B
F2	W	W	W	W	W	B
F3	W	W	W	W	W	B
F4	W	W	B	W	W	W
F5	W	W	W	W	W	B
F6	W	W	W	W	W	B
F7	W	W	W	W	W	B
F8	W	W	W	B	W	W
F9	W	W	W	W	W	B
F10	W	W	W	W	W	B
F11	W	W	W	B	W	W
F12	W	W	W	W	W	B
F13	W	W	B	W	W	W
F14	B	W	W	W	W	W
F15	W	W	B	W	W	B
F16	W	W	W	W	B	W
F17	W	W	W	W	W	B
F18	W	W	W	W	W	B
F19	W	W	W	B	W	W
F20	W	W	W	W	W	B
F21	W	W	W	W	W	B
F22	W	W	W	W	W	B
F23	W	W	W	W	W	B

Fig. 5 The SD values of SChoA algorithm on 23-standard benchmark functions



6.3.1 Assessment of exploitation capability

The uni-modal (F_1 – F_7) functions of 23-test suites have only one global compatibility, so these functions have been used to evaluate the exploitation phase. In Table 1, it can be seen that the SChoA algorithm gives better optimal solutions on these test suites as comparison to others. Experimental solutions of these test suites reveals that the SChoA algorithm has demonstrated better exploitation capabilities than the

others. The main motive for these global optima solutions is that the modification by sine–cosine functions in standard chimp optimization algorithm allowed the test suites to reach the best optimum global point. Hence, all simulations have proven that the SChoA algorithm strategy has a robust exploitation behavior which is important for most issues of compatibility that need to be addressed. As mentioned earlier, these functions are most suitable for these standard

Table 3 The optimal solutions of the algorithms on welded beam design problem

Algorithm	<i>h</i>	<i>l</i>	<i>t</i>	<i>b</i>	Optimal sol.
RANDOM	0.4575	4.7313	5.0853	0.6600	4.1185
GA	0.2489	6.1730	8.1789	0.2533	2.4331
Chimp	0.1656	4.0829	10	0.2046	1.9036
GSA	0.1821	3.8569	10.0000	0.2023	1.8799
RO	0.20368	3.52846	9.00423	0.20724	1.73534
CPSO	0.202369	3.544214	9.048210	0.205723	1.73148
WOA	0.2053	3.4842	9.0374	0.2062	1.7304
MVO	0.2054	3.4731	9.0445	0.2056	1.7264
GWO	0.2056	3.4783	9.0368	0.2057	1.7262
HSSAHHO	0.205727	3.470523	9.03673	0.205729	1.724867
SChoA	0.1998	3.4407	9.1056	0.2056	1.7229

test suits. The solutions prove that the SChoA algorithm is highly functional.

Additionally, superior to the global results of ChoA algorithm, it is revealed that modification is most successful and effective in providing the best and accurate solutions for complex optimization functions with a single objective phase and reaching best global optima.

6.3.2 Capability assessment

In multimodal (F_8 – F_{13}) and fixed dimension multimodal (F_{14} – F_{23}) test suits are contains many local optima and the number of design variables increases exponentially with the size of the problem compared to the unimodal function. These functions have the ability to assess the exploration capability of algorithms.

In general, these functions are used to indicate the suitability of the suit meta-heuristics. Here, all outputs in Table 1 show that the SChoA method achieves a higher detection capability and better exploitation capabilities.

6.3.3 Accuracy of the algorithms

The average values obtained by the algorithms are discussed in Table 2. These values has been divided into two different phases, such as (*B*) is best value and (*W*) is indicate the worse average value of the algorithm. In general, the least mean values of the algorithm denoted accuracy of the algorithm for the best optimal score. On the basis of table, we notice that the proposed algorithm is providing the least mean values at the maximum number of standard test suits. Hence, SChoA algorithm could be capable to provide the best global optima outputs for complex optimization functions.

6.3.4 Stability of SChoA algorithm

In this section has been verified the performance of the proposed algorithm through standard deviation (SD) outputs, as shown in the table and illustrated through Fig. 5.

Through Fig. 5, we could be analyzed that the proposed algorithms on the 23-standard benchmark functions are near to zero, and it means that the SChoA algorithm is stable on the given functions that were performed. Furthermore, the least standard scores show the fast convergence performance of the algorithms. Here, it could easily be seen that the SChoA algorithm is able to give the least standard score at maximum test suites outperform than the others. Therefore, it can be concluded that SChoA algorithm convergence speed is faster than others for trapping the best global optimum value during the search process in the search domain.

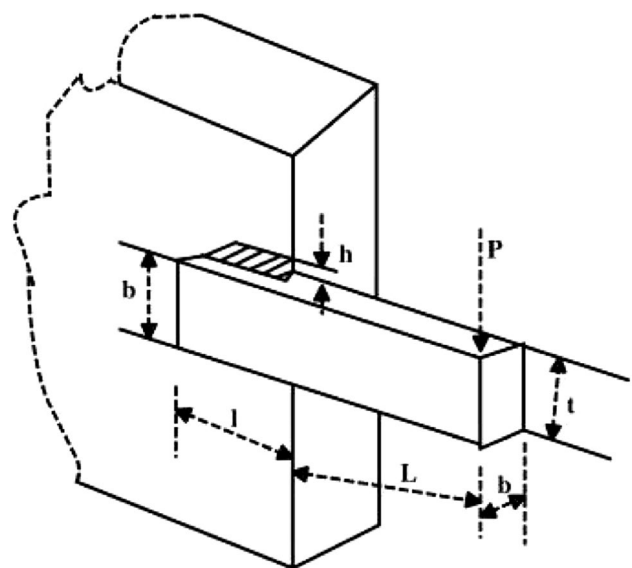


Fig. 6 Welded beam design

6.3.5 Convergence graphs analysis and discussion

The convergence performance graphs of uni-modal, multi-modal, and fixed dimension multi-modal test suites have been plotted through Figs. 2, 3 and 4. Please keep in mind that the average best solution for each iteration represents the average best solution found so far in each iteration.

In these graphs, Figs. 2, 3 and 4 could be seen easily that the convergence performance of the SChoA algorithm fall in two categories, such as if trapped the best optima value of the functions at least number of iterations and next, it tends to be accelerated as generation increases. Furthermore, as per Berg et al. [55], this behavior can assurance that a optimization algorithms ultimately converges to a point and finds locally. Therefore, SChoA algorithm is able to improve the fitness of all search chimps in the search domain and guarantee finding superior quality of global optima scores for complex optimization application as iteration increases.

Here, with this reasons, it can be discussed and as per reasoned to the modification of the chimp optimization algorithm. As the search agents move from higher values to global minimum values, so with the assumption of growth in ChoA, the overall chimps and their fitness are improved during the iterations.

Additionally, with the help of this strategy, we save the best value and this value helps to find the next best value to enhance the fitness of entire agents of the group during the search process. Here, we also observed that the SChoA algorithm is capable of getting the higher success rate in searching the optima solution in the complex space during the search space than sine–cosine functions and ChoA algorithm and it finds these goals fast with least number of iterations.

Table 4 The optimal solutions of the algorithms on tension/compression spring design problem

Algorithm	d	D	N	Optimum weight
RO	0.0413	0.3490	11.762	0.0126
DE	0.0516	0.3547	11.410	0.0126
ES	0.0519	0.3639	10.890	0.0126
GA	0.05148	0.35166	11.6322	0.01270
PSO	0.05172	0.35764	11.2445	0.01267
GWO	0.05169	0.35673	1.2888	0.01266
MFO	0.05100	0.36410	10.8684	0.01266
WOA	0.05120	0.34521	12.0040	0.01267
GSA	0.050276	0.323680	13.525410	0.0127022
Halton-PSO	0.0537	0.4058	8.9027	0.0126981
HSSAHHO	0.0514215	0.3535735	11.354662	0.012485407
Chimp	0.052457	0.34897	10.99876	0.012482
SChoA	0.052099	0.34567	10.65437	0.011873

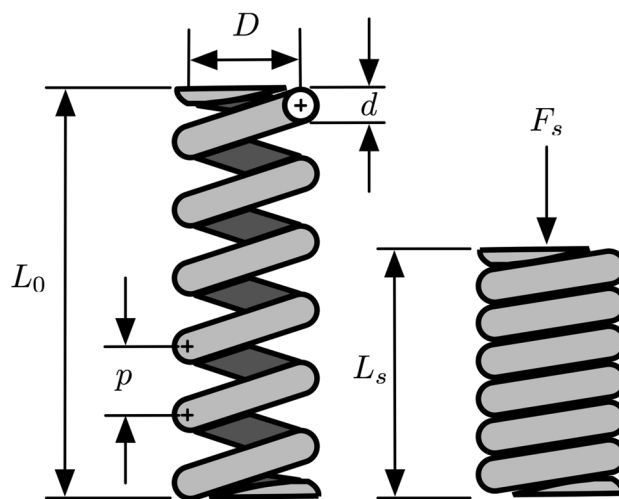


Fig. 7 Tension/compression spring design

With this modified strategy, which can help the search agents in the initial stages of the iteration to find new regions in the complex space during the search process and enhance the speed for searching the best optima goal with a least number of the iterations.

7 The SChoA algorithm for engineering designs

In this section has been tested the performance of the proposed SChoA algorithm on six-engineering real-world applications such as Welded beam, tension or compression spring, pressure vessel, multiple disk clutch brake, planetary gear train, and digital filters design problems, etc. To optimize the restricted issues of these functions have been compared the proposed method with various recent algorithms. These engineering designs are restricted in equal and in inequality measure. In the proposed methodology during the search process in the search domain the position updated through chimps.

However, between the fitness function of these designs and chimps, no any immediate relationship. On the basis of these designs have been verified the work performance of the proposed algorithm through recent optimizer.

7.1 Welded beam design problem

The objective of this design is decreasing the cost of manufacturing a welded beam, it is shown in Fig. 6. This graph represents a rigid component which is welded on a beam. At the end of the member, a load is applied. Here, we have been finding the values of thickness of weld (h), length of attached part of bar (l), the height of the bar (t), and thickness of the

bar (*b*) for evaluating the minimum price of this function (see Sect. 9.1 in “Appendix”).

The performance of the proposed method has been verified with recent optimization techniques on this design like that CPSO [56], Random [57], GA [58], RO [59], GWO [60], MVO [61], GSA [62] Chimp [51], HSSAHHO [63], and WOA [62]. The experimental solutions of this design are reported in Table 3. All these numerical solutions reveal that the proposed version is capable of giving the highly accurate and least optimal scores as compared to others.

7.2 Tension/compression spring design problem

The main objective of this design (see Sect. 9.4 in “Appendix”) is decreasing the pressure/voltage source weight, as shown in Fig. 7. The subject to constraints of this design has contain some restrictions such as shear stress, floating frequency, and minimum floating deflections. During this research have been applied various recent optimization techniques for the best or accurate possible solution of this design. By literature can be easily analysis that the

Table 5 The optimal solutions of the algorithms on pressure vessel design problem

Algorithm	X_1	X_2	X_3	X_4	Min $f(x)$
CGA	0.812500	0.437500	40.323900	200.0000	6288.744500
DGA	0.812500	0.437500	42.097400	176.654000	6059.946300
CPSO	0.812500	0.437500	42.091300	176.746500	6061.077700
HPSO	0.812500	0.437500	42.098400	176.636600	6059.714300
CDE	0.812500	0.437500	42.098411	176.637690	6059.734000
CSA	0.812500	0.437500	42.098445	176.636599	6059.714363
WOA	0.812500	0.437500	41.825682	180.046141	6093.211997
WOA-DE	0.812500	0.437500	42.098497	176.635957	6059.708057
WOA-BSA	0.812500	0.437500	42.098497	176.635954	6059.708025
Chimp	0	0	40.3301	200.0000	6058.6266
SChoA	1.0922	7.8270e-04	65.3254	10.0000	2318.0359

Fig. 8 Pressure vessel design

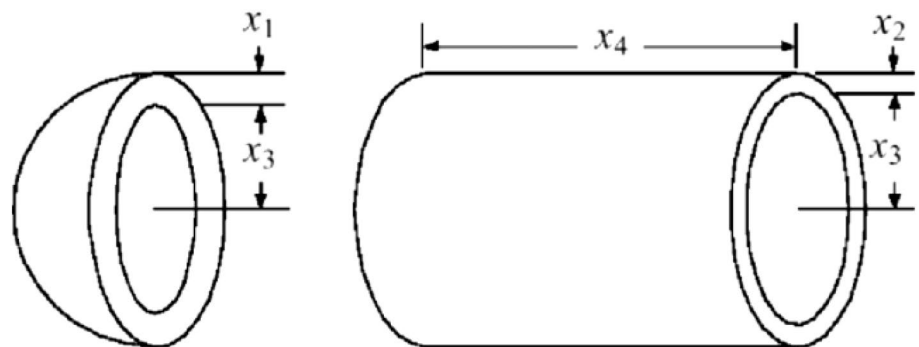


Table 6 The optimal solutions of the algorithms on multiple disk clutch brake design problem

Algorithm	X_1	X_2	X_3	X_4	X_5	Min $f(x)$
NSGA-II	70	90	1.5	1000	3	0.4704
SSA	77.1459	97.2218	1	628.1937	3.3809	0.3758
APSO	76	96	1	840	3	0.337181
TLBO	70	90	1	810	3	0.313657
WCA	70	90	1	910	3	0.313657
FSO	70	90	1	870	3	0.313656
SCA	68.8526	90	1	1000	2.6774	0.3027
Chimp	69.8782	90	1	1000	2.6537	0.2880
EGWO	69.4920	90	1.0099	1000	2.3690	0.2727
TACPSO	75.0044	95.0044	1	1000	2.1779	0.2648
MPSO	70	90	1	1000	2.3128	0.2598
SChoA	31.7505	51.7853	1	1000	4.4759	0.2246

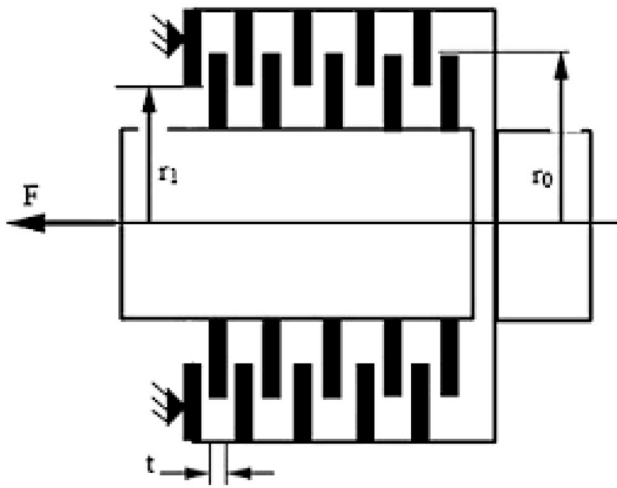


Fig. 9 Multiple disk clutch brake design

meta-heuristic most useful for giving the accurate solutions of these kinds of designs. All the simulation outputs obtained through algorithms are reported in Table 4.

During this study, the convergence performance of the proposed method has been verified with various recent techniques such as RO, DE, ES, GA, PSO, GWO, MFO, WOA, GSA, Halton-PSO [64], HSSAHHO [63], and Chimp [51] algorithms, which demonstrates that this algorithm can

highly be effective in solving real problems outperform than others.

7.3 Pressure vessel design

The main motive of this design (see Sect. 9.3 in “Appendix”) is reducing the construction cost of pressure vessel design, as seen in Fig. 8. This design cost include; associate with geometric parameters of the pressure vessel, shaping, materials, and welding. Therefore, for reaching the least cost for this function, calculations of the optima solutions of the geometric constants are necessary. The mathematical formulation of this design is provided in “Appendix”.

Table 5 provides the optimal solutions of SChoA algorithm and the other meta-heuristics CGA [65], DGA [66], CPSO [67], HPSO [68], CDE [69], CSA [70], WOA-DE [71], WOA-BSA [72], and Chimp [51] algorithm, respectively, for this design.

The solution accuracy of the proposed algorithm is clearly revealed in this design by giving the best or accurate result among the other recent meta-heuristics and is capable of evaluating the best design with the minimum cost.

7.4 Multiple disk clutch brake design

The main objective of this design is to minimize the weight of a multiple disc clutch brake (see Sect. 9.2 in “Appendix”) by considering various discrete variables such as inner

Table 7 The optimal solutions of the algorithms on planetary gear train design problem

Optimal value	SSA	EGWO	TACPSO	MPSO	SCA	Chimp	SChoA
X_1	35.4479	5.0200	65.7076	31.1533	56.9155	33.1809	36.1699
X_2	15.4274	2.7175	46.5288	54.4900	29.8118	19.6612	13.5100
X_3	14.0812	3.1710	20.5288	34.3362	13.5100	16.3498	17.9748
X_4	26.2348	2.6564	21.3166	26.5122	27.9305	18.5740	33.1821
X_5	15.9919	2.0467	20.0452	13.5100	24.4353	16.8072	35.7875
X_6	65.6218	12.6182	75.7920	117.4903	102.7000	68.8582	120.4121
X_7	3.1904	1.8098	2.3735	3.3266	1.5732	2.6944	3.1167
X_8	1.0999	1.0563	0.5100	0.5100	0.5100	0.6425	1.8475
X_9	0.5100	0.6231	2.2143	0.5100	0.8265	0.9238	0.6235
Min $f(x)$	1.1233	2.2233	0.5328	1.2583	0.9789	0.5332	0.5264

Table 8 The statistical solutions of the algorithms on planetary gear train design problem

Algorithm	X_{min}	X_{max}	X_{mean}	X_{sd}	X_{median}
EGWO	2.2233	4.2366e+07	8.4587e+05	5.1177e+06	708.7422
MPSO	1.2583	6.2092e+04	819.7833	6.9427e+03	1.6460
SSA	1.1233	1.0291e+04	363.5411	1.6264e+03	2.8148
SCA	0.9789	2.6436e+04	478.8328	3.0108e+03	2.0076
Chimp	0.5332	3.8973e+03	52.7726	435.3218	0.9433
TACPSO	0.5328	2.1118e+04	1.0605e+03	4.5581e+03	0.5946
SChoA	0.5264	27.9900	1.8058	4.2594	1.2615

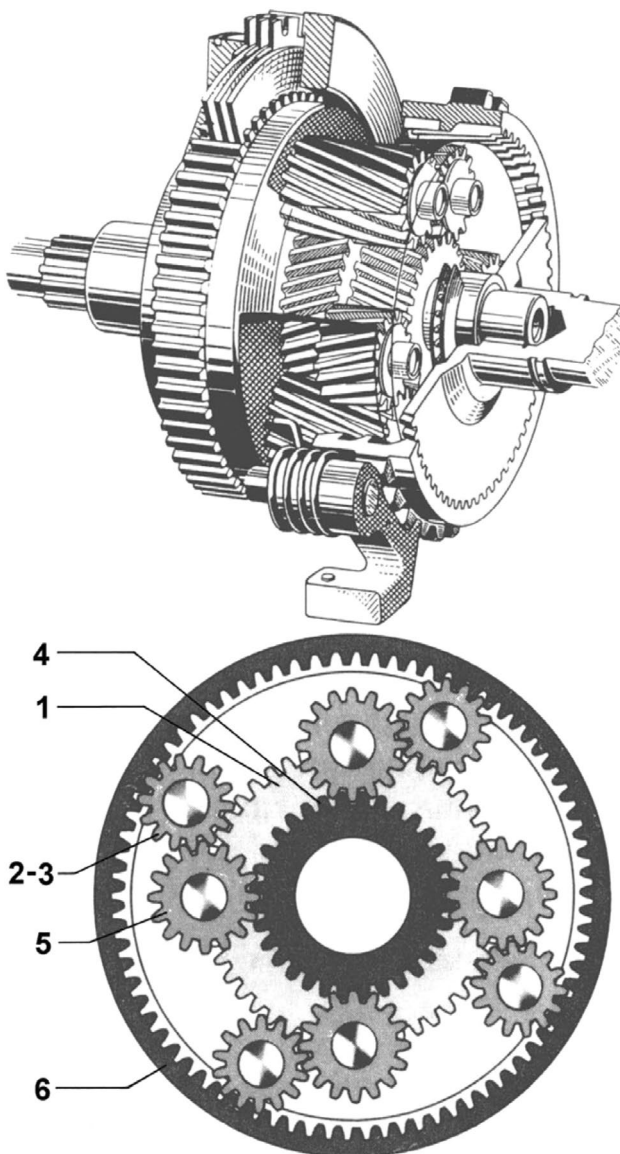


Fig. 10 Planetary gear train design

radius, number of friction surfaces outer radius, actuating force, and thickness of discs. This design includes eight different constraints based on operating conditions and the geometry. The global optima solution for this design is $f(x)$ at $X = (70, 90, 1, 810, 3)$ with one active constant. This design is represented by Fig. 9.

This design has been compared with various recent optimization algorithms like NSGA-II, TLBO, WCA, APSO [73], SCA [39], FSO [73], EGWO [74], TACPSO [75], MPSO [76], and Chimp [51] algorithms. The comparison to discover the best scores, given by such meta-heuristics, is reported in Table 6, respectively. Simulation results have

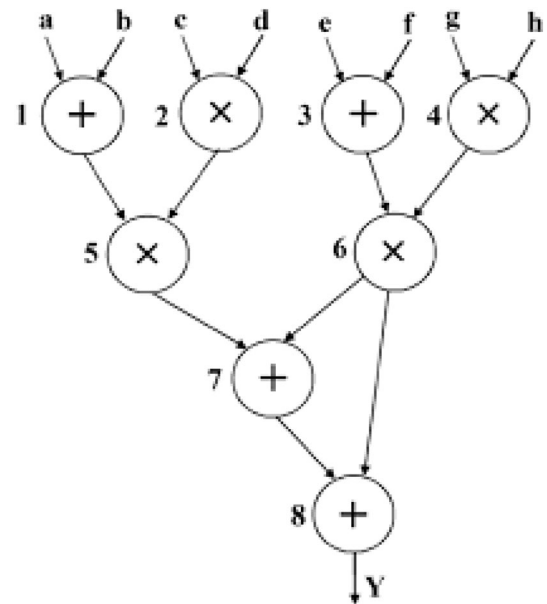


Fig. 11 Data flow graph for Eq. (23)

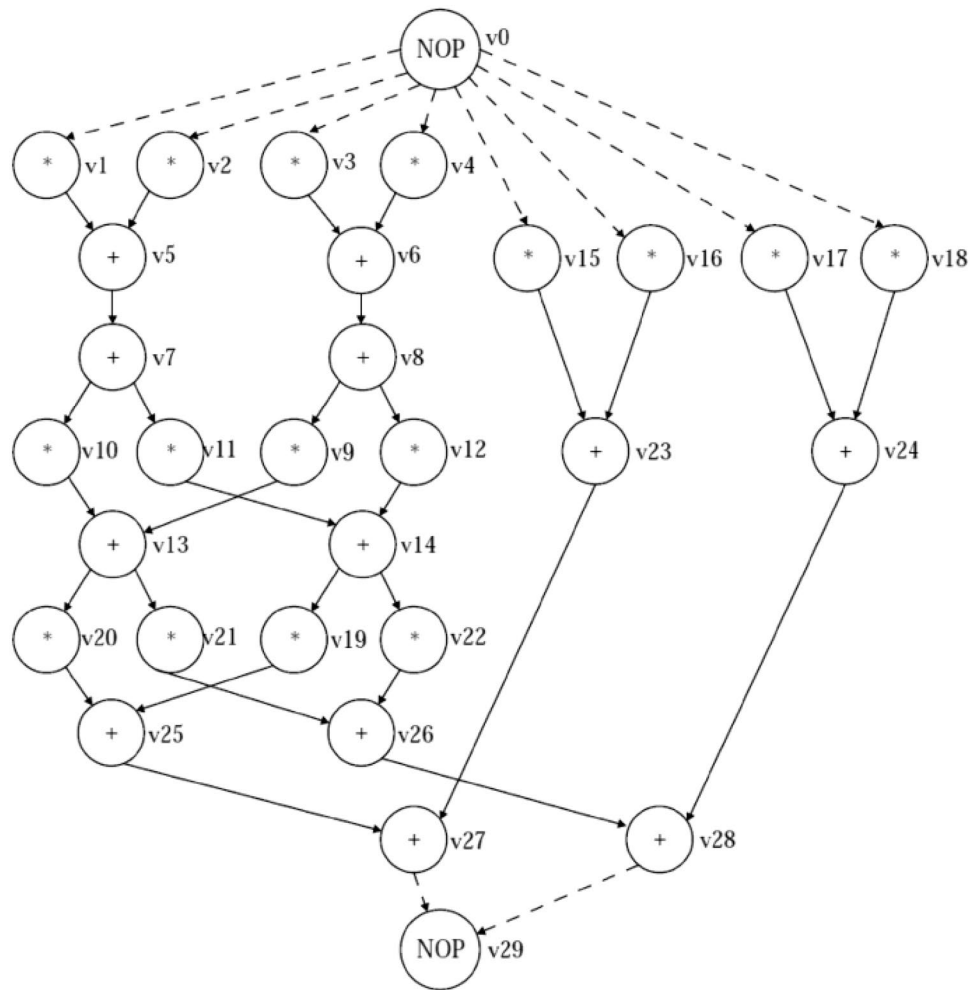
proven that the proposed algorithm is able to find the best minima solution for this design outperform than others.

7.5 Planetary gear train design

The main motive of this design (see Sect. 9.5 in “Appendix”) is to synthesize the gear-teeth number for an automatic planetary transmission system, as shown in Fig. 10: which is used in automobiles to reduce the errors in the gear ratio, Which involved 6 design variables based on the number of teeth in the gears ($N_1, N_2, N_3, N_4, N_5,$ and N_6), respectively, and which can only take specified integer constant values. There are three more design variables such as modules of the gears (m_1 and m_2) and number of planets gears (P), which can only take specific discrete constant values. In this design, 11 restrictions are considered in subject to constraints.

The numerical and statistical solutions of this design are reported in Tables 7 and 8. In these tables, has been compared the performance of the algorithms in different forms such as best optimal value, minima, maxima, mean, standard deviation and median, respectively. All these simulation results reveal that the proposed algorithm gives the best or accurate solutions for this design than others.

Fig. 12 The ARF data flow graph



7.6 HLS of digital filter design

In this section, we are implementing the proposed strategy for datapaths in digital filters. During designing, the digital filters HLS is highly paramount stage for this. In general, high-level optimization reduces design time at lower levels, leading to better circuit indices [77]. HLS is a stage of very-large-scale integration (VLSI) design where in behavioral description (here the filter’s behavioral characteristic) is converted into a structural characteristic [78, 79].

The high-level synthesis (HLS) is the first step in synthesizing a circuit and data flow graph (DFG) is used to display the behavioral description, which describes the operators’ type and the relationships between them. The hypothetical data flow graph is illustrated by the mathematical Eq. (23):

$$Y = (((a + b) \times (c \times d)) + ((e + f) \times (g \times h))) + ((e + f) \times (g \times h)). \tag{23}$$

In Fig. 11, 04 adder and 04 multiplier operators have been used for evaluating the given function (23). Here, 1, 2 and

3, 4 output operators are considered as the inputs operator 5 and 6. Similarly, 5 and 6 output operators are considered as the inputs operator 7 and 7, 8. At the end, 7 output operator is considered as the input operator 8, respectively.

In general, the digital filters are most used for images, videos, digital signal processing, process signals, communication applications, and so on. The finite impulse response (FIR), the infinite impulse response (IIR), the band-pass filter (BPF), the auto regressive filter (ARF), the wave digital filter (WDF), and the elliptic wave filter (EWF) are the digital filters used in this article. The data flow graph of the ARF used in this article is illustrated by Fig. 12 [80].

In this phase, the proposed strategy is applying for optimizing chip area, power, and delay in HLS of datapaths in digital filters. The obtained results of the proposed algorithm have been compared with the literature results of MFO [49] and PSO [49]. The working steps of the proposed strategy are illustrated by the following.

- The decision variables of the function are the same as the position of the chimps in the search domain. The position of the chimps can be defined by Eq. (24):

$$C = \begin{bmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,d} \\ c_{2,1} & c_{2,2} & \dots & c_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n,1} & c_{n,2} & \dots & c_{n,d} \end{bmatrix}, \tag{24}$$

Where n is the number of chimps or search agents and d is a dimension. The fitness values of the chimps or search agents in the search domain can be stored by Eq. (25):

$$FC = \begin{bmatrix} fc_1 \\ fc_2 \\ \vdots \\ fc_n \end{bmatrix}, \tag{25}$$

where n is number of search agents or chimps in search domain and fc_i represents the fitness values of the i th chimp. In the same way, the other two matrices can be defined for preys by the following equations:

Table 9 Results of algorithms on IIR DFG

Modes	Static	PSO			MFO			SChoA		
		Delay	Area	Power	Delay	Area	Power	Delay	Area	Power
$w_1 = 0.8$	Mean	5	6442.88	7007.66	5	5891.2	6349.97	5	5678.9	6187.76
$w_2 = 0.1$	SD	0	0.062	0.068	0	0.009	0.009	0	0.007	0.008
$w_3 = 0.1$										
$w_1 = 1$	Mean	7.42	3199.68	3217.85	7.08	3074.88	3142.2	6.12	2899.56	3023.1
$w_2 = 0.8$	SD	0.538	0.025	0.021	0.340	0.009	0.005	0.288	0.006	0.004
$w_3 = 0.1$										
$w_1 = 1$	Mean	7.3	3210.24	3187.59	7.14	3089.28	3142.2	6.55	2987.23	3078.4
$w_2 = 0.1$	SD	0.544	0.017	0.015	0.35	0.010	0.005	0.33	0.009	0.004
$w_3 = 0.8$										

Table 10 Results of algorithms on FIR DFG

Modes	Static	PSO			MFO			SChoA		
		Delay	Area	Power	Delay	Area	Power	Delay	Area	Power
$w_1 = 0.8$	mean	9.3	7855.04	8084.97	9.08	7143.04	7324.19	8.79	6876.23	7134.13
$w_2 = 0.1$	SD	0.462	0.077	0.088	0.274	0.063	0.073	0.219	0.045	0.054
$w_3 = 0.1$										
$w_1 = 1$	mean	15	3742.72	3222.89	14.84	3549.12	3157.33	13.45	3435.34	3043.87
$w_2 = 0.8$	SD	0.782	0.024	0.021	0.548	0.013	0.009	0.4734	0.011	0.007
$w_3 = 0.1$										
$w_1 = 1$	Mean	15.16	3723.2	3152.29	15.08	3694.08	3142.2	12.26	3478.23	3067.4
$w_2 = 0.1$	SD	0.618	0.001	0.008	0.274	0.01	0.005	0.213	0.001	0.004
$w_3 = 0.8$										

Table 11 Results of algorithms on ARF DFG

Modes	Static	PSO			MFO			SChoA		
		Delay	Area	Power	Delay	Area	Power	Delay	Area	Power
$w_1 = 0.8$	Mean	8.3	11,340.8	12,394.53	8.12	11,183.04	12,099.8	7.57	10,876.89	11,986.7
$w_2 = 0.1$	SD	0.505	0.023	0.027	0.328	0.004	0.004	0.298	0.003	0.004
$w_3 = 0.1$										
$w_1 = 1$	Mean	18.44	3627.2	3258.2	18.18	3543.04	3192.63	17.98	3467.12	3079.78
$w_2 = 0.8$	SD	0.787	0.016	0.017	0.388	0.011	0.014	0.298	0.009	0.010
$w_3 = 0.1$										
$w_1 = 1$	Mean	18.6	3688.32	3187.59	18.36	3575.04	3157.33	17.87	3498.57	3077.21
$w_2 = 0.1$	SD	0.670	0.014	0.014	0.485	0.12	0.009	0.408	0.011	0.008
$w_3 = 0.8$										

Table 12 Results of algorithms on EWF DFG

Modes	Static	PSO			MFO			SChoA		
		Delay	Area	Power	Delay	Area	Power	Delay	Area	Power
$w_1 = 0.8$	Mean	14	6924.8	6702.99	14	6902.4	6677.78	14	6865.6	6532.90
$w_2 = 0.1$	SD	0	0.006	0.007	0	0.007	0.008	0	0.005	0.006
$w_3 = 0.1$										
$w_1 = 1$	Mean	22.76	4039.04	3388.73	24.62	3873.6	3217.85	23.77	3765.4	3189.67
$w_2 = 0.8$	SD	5.057	0.043	0.057	4.347	0.016	0.021	4.123	0.015	0.019
$w_3 = 0.1$										
$w_1 = 1$	Mean	24.46	4070.08	3323.16	24.74	3954.56	3182.55	23.67	3894.16	3090.33
$w_2 = 0.1$	SD	3.95	0.043	0.057	3.433	0.15	0.013	3.103	0.012	0.015
$w_3 = 0.8$										

Table 13 Results of algorithms on BPF DFG

Modes	Static	PSO			MFO			SChoA		
		Delay	Area	Power	Delay	Area	Power	Delay	Area	Power
$w_1 = 0.8$	Mean	8.22	6996.48	7203.75	8.02	6458.88	6584.18	7.89	6345.98	6432.11
$w_2 = 0.1$	SD	0.545	0.062	0.069	0.141	0.021	0.022	0.105	0.019	0.020
$w_3 = 0.1$										
$w_1 = 1$	Mean	15.94	3884.48	3333.84	17.04	3628.8	3217.85	16.56	3534.22	3187.3
$w_2 = 0.8$	SD	3.449	0.033	0.027	4.035	0.02	0.016	3.546	0.01	0.011
$w_3 = 0.1$										
$w_1 = 1$	Mean	17.42	3870.72	3263.24	17.48	3728.96	3197.68	16.98	3675.11	3044.45
$w_2 = 0.1$	SD	3.038	0.025	0.025	2.894	0.28	0.014	2.789	0.024	0.012
$w_3 = 0.8$										

Table 14 Results of algorithms on WDF DFG

Modes	Static	PSO			MFO			SChoA		
		Delay	Area	Power	Delay	Area	Power	Delay	Area	Power
$w_1 = 0.8$	Mean	14.12	6904.64	6353.97	14.02	7040.96	6509.13	13.12	6987.43	6453.33
$w_2 = 0.1$	SD	0.385	0.063	0.082	0.141	0.028	0.037	0.127	0.026	0.034
$w_3 = 0.1$										
$w_1 = 1$	Mean	22.28	4310.08	3359.06	23.66	4184.32	3253.15	22.43	4098.44	3187.99
$w_2 = 0.8$	SD	5.789	0.019	0.032	4.926	0.016	0.024	4.109	0.014	0.022
$w_3 = 0.1$										
$w_1 = 1$	Mean	23.18	4342.72	3298.54	25.02	4257.28	3192.63	24.98	4201.22	3105.32
$w_2 = 0.1$	SD	5.283	0.016	0.029	4.321	0.012	0.014	4.002	0.010	0.012
$w_3 = 0.8$										

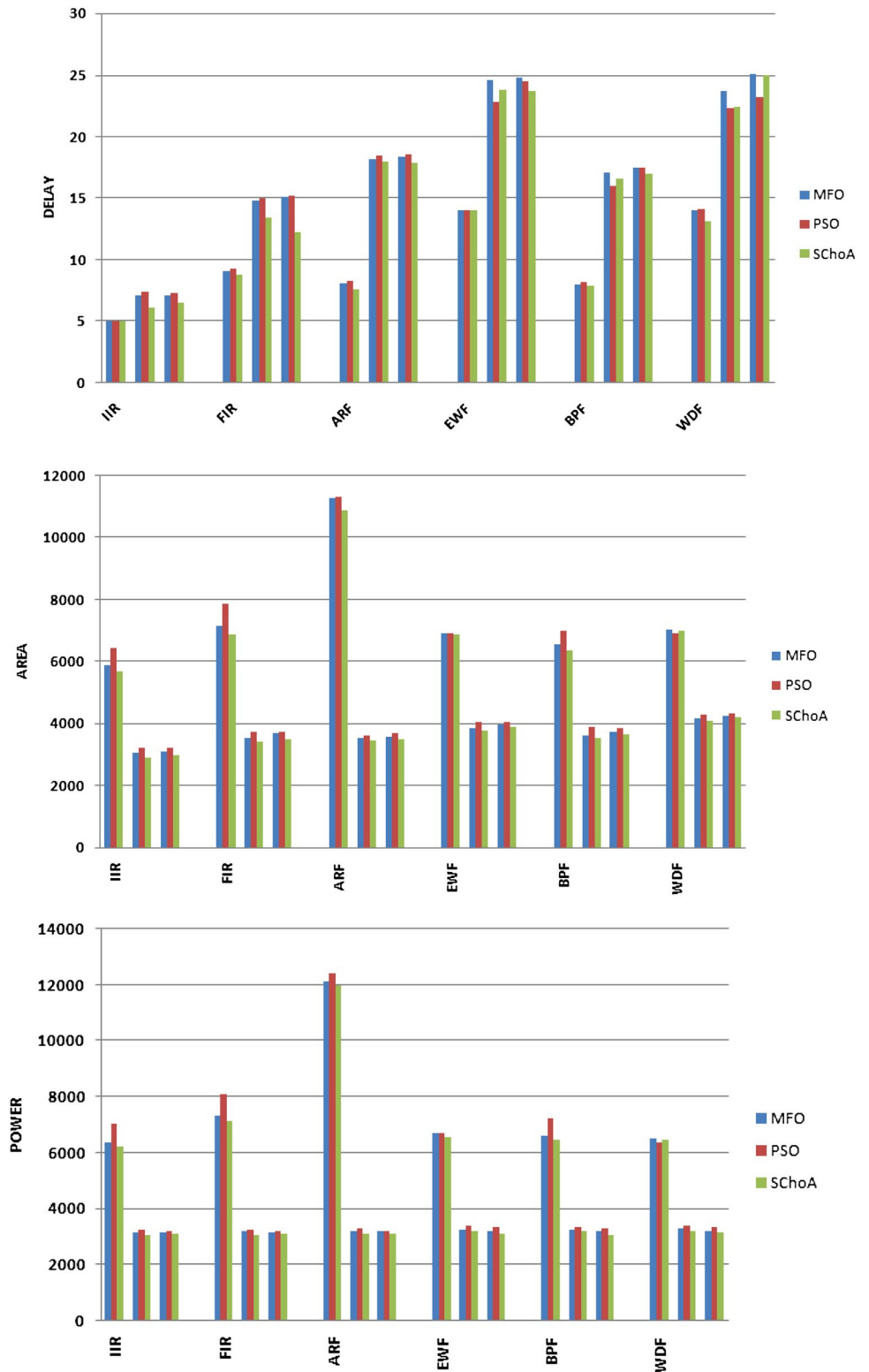
$$P = \begin{bmatrix} p_{1,1} & p_{1,2} & \dots & p_{1,d} \\ p_{2,1} & p_{2,2} & \dots & p_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,1} & p_{n,2} & \dots & p_{n,d} \end{bmatrix} \tag{26}$$

$$FP = \begin{bmatrix} fp_1 \\ fp_2 \\ \vdots \\ fp_n \end{bmatrix}, \tag{27}$$

where n is number of preys, d is the dimension of the function, and fp_i is the fitness value of i^{th} prey.

- In the next phase, the new location and distance of the given node is evaluated by Eqs. (16)–(20); these equations denote the new positions of variables of chimp and (1)–(5); these equations indicate the distance amid the position decision variables of the prey and the position of variables of chimp.
- For updating the number of search agents or chimps in each iterations with the objective of enhancing the extractability of the algorithm has used Eq. (28):

Fig. 13 The best solutions for delay, occupied area, and power in HLS of digital filter problem



$$C_N = (N - I) \times \frac{N}{M_I}, \tag{28}$$

where N is the number of search agents or chimps, I is the current iteration, and M_I is the maximum number of iterations. Hence, the search agents or chimps update

their positions in the search domain w.r.t. the best prey or goal in the last iterations.

- Fitness function for DFG: For evaluating the area, power, and delay in the proposed hybrid algorithm and verifying its accuracy performance against the results of MFO [49] and PSO [49] algorithms, the following fitness functions are applied:

$$F = w_1 \times \frac{l_t}{l_{\max}} + w_2 \times \frac{a_t}{a_{\max}} + w_3 \times \frac{p_t}{p_{\max}}, \quad (29)$$

where F denotes the fitness function, w_1, w_2, w_3 represent the weights of the power, delay, and area terms, l_t denotes the schedule length of sample evaluated, a_t denotes the total number of registers and transistors in the operators, p_t denotes the power consumption of operators, l_{\max} denotes the long scheduled length in the current crowd, a_{\max} denotes the largest area in the current crowd, and p_{\max} denotes the highest power in the current crowd, respectively.

Further meticulously elaborates on the solutions obtained by the proposed algorithm in the next phase of this section. These solutions have been verified through the literature results obtained by MFO [49] and PSO [49]. The program of the proposed algorithm has been runned on Matlab-R2015a under the system with Intel (R) Core (TM) i3-8130 U processor and 8GM of RAM. During this process, various constant settings have been applied for obtaining the solutions such as number search agents or chimps is 30, maximum number of iterations are 100, and maximum number of sources and operational units are 5, respectively.

The obtained experimental solutions of the HLS of the digital filters are illustrated in Tables 9, 10, 11, 12, 13 and 14. And the best solutions of the proposed algorithm for lowest delay, occupied area, and power are illustrated in Fig. 13. These results have been categorized into three different phases like delay, area, and power. Additionally, all these results evaluated on three different modes $w_1 = 0.8, w_2 = 1, w_3 = 1$, and $w_1 = 1, w_2 = 0.8, w_3 = 1$, and $w_1 = 1, w_2 = 1, w_3 = 0.8$, respectively. For each mode, the mean of the acquired response for a 50-times execution for the proposed method has been tabulated along with their relevant the standard deviation(sd). The point to keep in mind is that, here, the sd for the proposed method has been reported for a comparison and superior presentation of the result due to the power consumption and largeness of the occupied area.

In Tables 9, 10, 11, 12, 13 and 14, all results are represent the best response in terms of the area, power, and delay in the three algorithms. All the results of IIR, FIR, ARF, EWF, BPF, and WDF-DFG have been obtained by replacing the parameter values of the weight (w_1, w_2, w_3) and a significant improvement in the optimal responses of

the algorithms observed. For instance, the best delay will be achieved, compared to the other two modes, when assuming a weight of 0.8 for w_1 ; this factor is connected to the delay, and assuming a coefficient of 0.1 for w_2 and w_3 as the coefficients of the occupied area and power. The same is true for the other two modes.

The proposed hybrid algorithm is a superior than the MFO [49] and PSO [49] algorithms with average values 2899.56 and 3078.4 for the lowest power consumption and occupied area. Similarly, in Table 10, the lowest average values for the delay 8.70, 0.0219, 13.45, 0.4734, 12.26, and 0.213 have been obtained by proposed algorithm than others. These results prove that the proposed method is able to reduced the delay time as comparison to MFO and PSO algorithm for digital filter problem.

Hence, the proposed method is able to provide the best output response in terms of delay, area, and power consumption for high-level synthesis of VLSI circuits.

8 Conclusion

The Chimp optimization algorithm is a most recent population-based meta-heuristic. Chimp optimization (ChoA) could be widely applied to solve complex tasks of different domains due to its inexpensive computational and implementation simplicity overhead. Nevertheless, ChoA is also faced several drawbacks during search process in the search domain like ineffective in balancing exploitation and exploration, premature convergence, and complex multipeak search problems, especially in easily trapped in the local optima results. To overcome the shortcomings of ChoA, a modified chimp optimization algorithm with sine cosine functions is developed to solve single objective functions. In this work, we developed as new update position equations with sine–cosine functions to improving the convergence performance of the search agents during to search process in the search domain. So that, this algorithm could be effectively controlled the local search and convergence to the global optimum result during the search process.

During this extensive investigational research efforts, a huge number of simulations have been carried out for verifying the performance and results accuracy of the proposed SChoA method. Its performance has been verified through 23-benchmark and 06-engineering designs. Experiential solutions proved that SChoA algorithm possesses some superior performance in terms of numerical results, statistical measurements and convergence curves, stability, and robustness, as compared to state-of-the-art PSO and others, such as MPSO, SCA, PSO, TACPSO, and Chimp. Hence, the proposed algorithm can be a good alternative for tackling various complex applications.

Additionally, the proposed strategy is able to successfully solve the HLS of datapaths in digital filters problem in terms of lowest delay, area, and power, respectively, than the other two algorithms.

In future direction, the SChoA can be implemented on complex real-life application problems. The constrained and multi-objective version of SChoA or based feature selection approach for high dimensional with small instance to simultaneously maximize the classification performance and minimize the number of features can be developed also.

Compliances with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Appendix

See Table 15.

Table 15 23-standard benchmark test functions

Key	Function formulation	$f(x^*)$	Category	Dimension	Range
f_1	$\sum_i^n x_i^2$	0	U	10,30,50,100	$x_i \in [-100, 100]$
f_2	$\sum_{i=1}^n x_i + \prod_{i=1}^{x_i} x_i $	0	U	10,30,50,100	$x_i \in [-10, 10]$
f_3	$\sum_{i=1}^n \left(\sum_{j=1}^i x_j\right)^2$	0	U	10,30,50,100	$x_i \in [-100, 100]$
f_4	$\max_i \{ x_i , 1 \leq i \leq n\}$	0	U	10,30,50,100	$x_i \in [-100, 100]$
f_5	$\sum_{i=1}^{n-1} [100(x_{i+1} - x_i)^2 + (x_i - 1)^2]$	0	U	10,30,50,100	$x_i \in [-30, 30]$
f_6	$\sum_{i=1}^n (x_i + 0.5)^2$	0	U	10,30,50,100	$x_i \in [-100, 100]$
f_7	$\sum_{i=1}^n (ix_i^4 + rand[0, 1])$	0	U	10,30,50,100	$x_i \in [-128, 128]$
f_8	$\sum_{i=1}^n -x_i \sin\left(\sqrt{ x_i }\right)$	-418.9829×5	M	10,30,50,100	$x_i \in [-500, 500]$
f_9	$\sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	0	M	10,30,50,100	$x_i \in [-5.12, 5.12]$
f_{10}	$-20e^{(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2})} - e^{(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i) + 20 + e)}$	0	M	10,30,50,100	$x_i \in [-32, 32]$
f_{11}	$\frac{1}{400} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}} + 1\right)$	0	M	10,30,50,100	$x_i \in [-600, 600]$
f_{12}	$\frac{\pi}{n} \left\{ 10 \sin(\pi y_i) + \prod_{i=1}^{n-1} (y_i - 1)^2 \right\} \left 1 + 10 \sin^2(\pi y_i + 1 + (y_{n-1})^2) \right $	0	M	10,30,50,100	$x_i \in [-50, 50]$
f_{13}	$0.1 \left\{ \sin^2(3\pi x_i) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 \right.$ $\left. [1 + \sin^2(2\pi x_n)] \right\} + \sum_{i=1}^n \mu(x_i, 5, 100, 4)$	0	M	10,30,50,100	$x_i \in [-50, 50]$
f_{14}	$\left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{\sum_{i=1}^2 (x_i - a_j)^6}\right)^{-1}$	1	F	2	$x_i \in [-65, 65]$
f_{15}	$\sum_{i=1}^{11} \left[a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_i + x_4} \right]^2$	0.0003	F	4	$x_i \in [-5, 5]$
f_{16}	$4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	-1.0316	F	2	$x_i \in [-5, 5]$
f_{17}	$\left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	0.398	F	2	$x_i \in [-5, 5]$
f_{18}	$\left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right]$ $\times [(30 + (2x_1 - 3x_2)^2)(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	3	F	2	$x_i \in [-2, 2]$
f_{19}	$\sum_{i=1}^4 c_i e^{-\sum_{j=1}^3 (x_j - p_{ij})^2}$	-3.86	F	3	$x_i \in [1, 3]$
f_{20}	$\sum_{i=1}^4 c_i e^{-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2}$	-3.32	F	6	$x_i \in [0, 1]$
f_{21}	$-\sum_{i=1}^5 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	-10.1532	F	4	$x_i \in [0, 10]$
f_{22}	$-\sum_{i=1}^7 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	-10.4028	F	4	$x_i \in [0, 10]$
f_{23}	$-\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	-10.5363	F	4	$x_i \in [0, 10]$

Welded beam design problem

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

Minimize $f(\mathbf{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$

Subject to:

$g_1(\mathbf{x}) = \tau(\mathbf{x}) - 13600 \leq 0$

$g_2(\mathbf{x}) = \sigma(\mathbf{x}) - 30000 \leq 0$

$g_3(\mathbf{x}) = x_1 - x_4 \leq 0$

$g_4(\mathbf{x}) = 0.10471(x_1^2) + 0.04811x_3x_4(14 + x_2) - 5.0 \leq 0$

$g_6(\mathbf{x}) = \delta(\mathbf{x}) - 0.25 \leq 0$

$g_7(\mathbf{x}) = 6000 - p_c(\mathbf{x}) \leq 0$

where

$$\tau(\mathbf{x}) = \sqrt{(\tau') + (2\tau'\tau'')\frac{x_2}{2R} + (\tau'')^2}$$

$$\tau' = \frac{6000}{\sqrt{2}x_1x_2}$$

$$\tau'' = \frac{MR}{J}$$

$$M = 6000\left(14 + \frac{x_2}{2}\right)$$

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

$$j = 2 \left\{ x_1x_2\sqrt{2} \left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2 \right] \right\}$$

$$\sigma(\mathbf{x}) = \frac{504000}{x_4x_3^2}$$

$$\delta(\mathbf{x}) = \frac{65856000}{(30 \times 10^6)x_4x_3^3}$$

$$p_c(\mathbf{x}) = \frac{4.013(30 \times 10^6) \sqrt{\frac{x_3^2x_4^6}{36}} \left(1 - \frac{x_3 \sqrt{\frac{30 \times 10^6}{4(12 \times 10^6)}}}{28} \right)}{196}$$

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

Tension/compression spring design problem

Consider:

$\mathbf{x} = [x_1 x_2 x_3] = [d D N]$

Minimize $f(\mathbf{x}) = (x_3 + 2)x_2x_1^2$

subject to:

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

$g_2(\mathbf{x}) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0$

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

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

with $0.05 \leq x_1 \leq 2.0, 0.25 \leq x_2 \leq 1.3,$ and $2.0 \leq x_3 \leq 15.0$.

Pressure vessel design problem

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

Subject to:

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

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

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

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

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

$10 \leq x_i \leq 200, i = 3, 4.$

Multiple disk clutch brake design problem

Minimize $f(x) = \pi(x_2^2 - x_1^2)x_3(x_5 + 1)\rho$

Variable range

$x_1 \in \{60, 61, \dots, 79, 80\}$

$x_2 \in \{90, 91, \dots, 109, 110\}$

$x_3 \in \{1, 1.5, \dots, 3\}$

$x_4 \in \{600, 610, \dots, 990, 1000\}$

$x_5 \in \{2, 3, \dots, 8, 9\}$

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

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

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

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

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

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

$g_7(x) = T \leq 0$

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

$\Delta r = 20 \text{ (mm)}, I_z = 55 \text{ (kg, mm}^2\text{)}, p_{\max} = 1 \text{ (MPa)},$

$T_{\max} = 15 \text{ (s)}, \mu = 0.50$

$s = 1.50, M_s = 40 \text{ (Nm)}, M_f = 3 \text{ (Nm)}, n = 250 \text{ (rpm)},$

$v_{sr, \max} = 10 \left(\frac{m}{s}\right), L_{\max} = 30 \text{ (mm)}.$

Planetary gear train design problem

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

where

$i_1 = N_6/N_4$

$i_{01} = 3.11$

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

$i_{02} = 1.84$

$i_R = -\frac{N_2N_6}{N_1N_3}$

$i_{0R} = -3.11$

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

Subject to:

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

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

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

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

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

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

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

$g_8(X) = (N_6 - N_3)^2 + (N_4 + N_5)^2 - (N_6 - N_3)(N_4 + N_5)$

$\cos\left(\frac{2\pi}{p} - \beta\right) \leq (N + 3 + N_5 + 2 + \delta_{35})^2$

where

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

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

$g_{10}(X) = N_6 - N_4 - 2N_5 - 4 - 2\delta_{56} \geq 0$

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

where

$D_{\max} = 220, p = (3, 4, 5), m_1, m_3 = (1.75, 2.0, 2.25, 2.5, 2.75, 3.0),$

$\delta_{22}, \delta_{33}, \delta_{55}, \delta_{35}, \delta_{56} = 0.5$

$17 \leq N_1 \leq 96, 14 \leq N_2 \leq 54, 14 \leq N_3 \leq 51, 17 \leq N_4 \leq 46,$

$14 \leq N_5 \leq 51, 48 \leq N_6 \leq 124, N_i = \text{integer}.$

References

1. Dhiman G, Garg M (2020) MOSSE: a novel hybrid multi-objective meta-heuristic algorithm for engineering design problems. *Soft Comput*:1–20
2. Dhiman G (2019) Multi-objective metaheuristic approaches for data clustering in engineering application(s), Ph.D. thesis
3. Dhiman G, Kaur A (2019) HKN-RVEA: a novel many-objective evolutionary algorithm for car side impact bar crashworthiness problem. *Int J Vehicle Design* 80(2–4):257–284
4. Dhiman G, Singh KK, Slowik A, Chang V, Yildiz AR, Kaur A, Garg M (2020) Emosoa: a new evolutionary multi-objective seagull optimization algorithm for global optimization. *Int J Mach Learn Cybern*:1–26
5. Dhiman G, Oliva D, Kaur A, Singh KK, Vimal S, Sharma A, Cengiz K (2020) Bepo: A novel binary emperor penguin optimizer for automatic feature selection. *Knowl Based Syst*:106560
6. Dhiman G, Singh KK, Soni M, Nagar A, Dehghani M, Slowik A, Kaur A, Sharma A, Houssein EH, Cengiz K (2020) MOSOA: a new multi-objective seagull optimization algorithm. *Expert Syst Appl*:114150
7. Kaur H, Rai A, Bhatia SS, Dhiman G (2020) MOEPO: a novel multi-objective emperor penguin optimizer for global optimization: Special application in ranking of cloud service providers. *Eng Appl Artif Intell* 96:104008
8. Dhiman G, Kaur A (2019) STOA: a bio-inspired based optimization algorithm for industrial engineering problems. *Eng Appl Artif Intell* 82:148–174
9. Kaur S, Awasthi LK, Sangal A, Dhiman G (2020) Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. *Eng Appl Artif Intell* 90:103541
10. Dehghani M, Montazeri Z, Malik OP, Dhiman G, Kumar V (2019) Bosa: binary orientation search algorithm. *Int J Innov Technol Explor Eng* 9:5306–10
11. Dhiman G (2019) ESA: a hybrid bio-inspired metaheuristic optimization approach for engineering problems. *Eng Comput*:1–31
12. Dhiman G (2020) MOSHEPO: a hybrid multi-objective approach to solve economic load dispatch and micro grid problems. *Appl Intell* 50(1):119–137
13. Dhiman G, Kumar V (2018a) Emperor penguin optimizer: a bio-inspired algorithm for engineering problems. *Knowl Based Syst* 159:20–50
14. Dhiman G, Kumar V (2018) Multi-objective spotted hyena optimizer: a multi-objective optimization algorithm for engineering problems. *Knowl Based Syst* 150:175–197
15. Dhiman G, Kumar V (2019) Seagull optimization algorithm: theory and its applications for large-scale industrial engineering problems. *Knowl Based Syst* 165:169–196
16. Dhiman G, Kumar V (2019) KNRVEA: a hybrid evolutionary algorithm based on knee points and reference vector adaptation strategies for many-objective optimization. *Appl Intell* 49(7):2434–2460
17. Simpson AR, Dandy GC, Murphy LJ (1994) Genetic algorithms compared to other techniques for pipe optimization. *J Water Resour Plan Manag* 120(4):423–443
18. Boussaïf DI, Lepagnot J, Siarry P (2013) A survey on optimization metaheuristics. *Inf Sci* 237:82–117
19. Parejo JA, Ruiz-Cortés A, Lozano S, Fernandez P (2012) Metaheuristic optimization frameworks: a survey and benchmarking. *Soft Comput* 16(3):527–561
20. Zakeri E, Moezi SA, Bazargan-Lari Y, Zare A (2017) Multi-tracker optimization algorithm: a general algorithm for solving engineering optimization problems. *Iran J Sci Technol Trans Mech Eng* 41(4):315–341
21. Xue B, Zhang M, Browne WN, Yao X (2016) A survey on evolutionary computation approaches to feature selection. *IEEE Trans Evol Comput* 20(4):606–626
22. Bozorg-Haddad O (2018) Advanced optimization by nature-inspired algorithms. Springer, New York
23. Fister I, Strans D, Yang X-S (2015) Adaptation and hybridization in nature-inspired algorithms. In: *Adaptation and hybridization in computational intelligence*. Springer, pp 3–50
24. Zeng S, Dai J, Yi Z, He W (2018) A modified sine-cosine algorithm based on neighborhood search and greedy levy mutation. In: *Computational intelligence and neuroscience*. Hindawi, pp 1–20
25. Ibrahim RA, Ewees AA, Oliva D, Elaziz MA, Lu S (2018) Improved salp Swarm algorithm based on particle swarm optimization for feature selection. *J Ambient Intell Human Comput*:1–15
26. San-José-Revuelta LM, Arribas JI (2018) A new approach for the design of digital frequency selective FIR filters using an FPA-based algorithm. *Expert Syst Appl* 106:92–106
27. Ibrahim A, Ahmed A, Hussein S, Hassanien AE (2018) Fish image segmentation using salp swarm algorithm. *International Conference on advanced machine learning technologies and applications*. Springer, New York, pp 42–51
28. Saha SK, Ghoshal SP, Kar R, Mandal D (2013) Cat swarm optimization algorithm for optimal linear phase FIR filter design. *ISA Trans* 52(6):781–794
29. Liu X, Xu H, Application on target localization based on salp swarm algorithm. In: *37th Chinese Control Conference (CCC)*. IEEE, pp 4542–4545 (2018)
30. Aggarwal A, Rawat TK, Upadhyay DK (2016) Design of optimal digital fir filters using evolutionary and swarm optimization techniques. *AEU-International J Electron Commun* 70(4):373–385
31. Yagain D, Vijayakrishna A (2015) A novel framework for retiming using evolutionary computation for high level synthesis of digital filters. *Swarm Evol Comput* 20:37–47
32. Bairathi D, Gopalani D (2019) Salp swarm algorithm (SSA) for training feed-forward neural networks. *Soft computing for problem solving*. Springer, New York, pp 521–534
33. Sahu P, Prusty R, Sahoo B (2020) Modified sine cosine algorithm-based fuzzy-aided pid controller for automatic generation control of multiarea power systems. *Methodologies and application*. Springer, New York, pp 12919–12936
34. Hashim FA, Houssein EH, Mabrouk MS, Al-Atabany W, Mirjalili S (2019) Henry gas solubility optimization: A novel physics-based algorithm. *Future Gener Comput Syst* 101:646–667
35. Talbi E-G (2009) *Metaheuristics: from design to implementation*, vol 74. Wiley, New York
36. Bonabeau E, Dorigo M, Marco DRDF, Théraulaz G et al (1999) *Swarm intelligence: from natural to artificial systems*, no 1. Oxford University Press, Oxford
37. Eberhart R, Kennedy J (1995) Particle swarm optimization. In: *Proceedings of the IEEE international conference on neural networks*, vol. 4, pp 1942–1948 (**Citeseer**)
38. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Adv Eng Soft* 114:163–191
39. Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. *Knowledge-based systems*. Elsevier, Amsterdam, pp 120–133
40. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: Algorithm and applications. *Future Gener Comput Syst* 97:849–872
41. Sayed GI, Khoriba G, Haggag MH (2018) A novel chaotic salp swarm algorithm for global optimization and feature selection. *Appl Intell* 48(10):3462–3481
42. Rizk-Allah RM, Hassanien AE, Elhoseny M, Gunasekaran M (2018) A new binary salp swarm algorithm: development and application for optimization tasks. *Neural Comput Appl*:1–23

43. Ab Wahab MN, Nefti-Meziani S, Atyabi A (2015) A comprehensive review of swarm optimization algorithms. *PLoS One* 10(5):e0122827
44. Alreshdeedi SS, Lu S, Elaziz MA, Ewees AA (2019) Improved multiobjective salp swarm optimization for virtual machine placement in cloud computing. *Human Centric Comput Inf Sci* 9(1):15
45. Zhao H, Huang G, Yan N (2018) Forecasting energy-related CO2 emissions employing a novel ssa-lssvm model: considering structural factors in china. *Energies* 11(4):781
46. dos Santos Coelho L, Mariani VC (2008) Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization. *Expert Syst Appl* 34(3):1905–1913
47. Faris H, Mafarja MM, Heidari AA, Aljarah I, Ala'M A-Z, Mirjalili S, Fujita H (2018) An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. *Knowl Based Syst* 154:43–67
48. Qu C, Zeng Z, Dai J, Yi Z, He W (2018) A modified sine-cosine algorithm based on neighborhood search and Greedy Levy mutation
49. Esmaeili M, Zahiri S, Razavi S (2020) A novel method for high-level synthesis of datapaths in digital filters using a moth-flame optimization algorithm. *Evolutionary intelligence*, no 13. Springer, New York, pp 399–414
50. Gholizadeh S, Sojoudizadeh R (2019) Modified sine–cosine algorithm for sizing optimization of Truss structures with discrete design variables
51. Khishe M, Mosavi MR (2020) Chimp optimization algorithm. In: *Expert systems with applications*, vol 149. Elsevier, Amsterdam
52. Dhiman G, Kumar V (2017) Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. *Adv Eng Softw* 114:48–70
53. Mirjalili S (2015) Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl Based Syst* 89:228–249
54. Arora S, Singh S (2019) Butterfly optimization algorithm: a novel approach for global optimization. *Soft Comput* 23(3):715–734
55. Van Den Berg R, Pogosky AY, Leonov G, Rooda J (2006) Design of convergent switched systems. Group coordination and cooperative control. Springer, New York, pp 291–311
56. Krohling RA, dos Santos Coelho L (2006) Coevolutionary particle swarm optimization using gaussian distribution for solving constrained optimization problems. *IEEE Trans Syst Man Cybern Part B (Cybern)* 36(6):1407–1416
57. Ragsdell K, Phillips D (1976) Optimal design of a class of welded structures using geometric programming. *J Eng Ind* 98:1021–1025
58. Coello CAC (2000) Use of a self-adaptive penalty approach for engineering optimization problems. *Comput Ind* 41(2):113–127
59. Kaveh A, Khayatazad M (2012) A new meta-heuristic method: ray optimization. *Comput Struct* 112:283–294
60. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
61. Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Comput Appl* 27(2):495–513
62. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
63. Singh N, Houssein EH, Singh SB (2020) An efficient hybrid salp swarm harris hawks optimization for optimization problems. Communicated in engineering applications of artificial intelligence. Elsevier, Amsterdam, pp 1–50
64. Dighehara PA, Chegini SN, Bagheri A, Roknsaraei MP (2020) An improved particle swarm optimization based on the reinforcement of the population initialization phase by scrambled halton sequence. *Cogent Eng* 7(1):1737383
65. Alsa'deh A, Rafiee H, Meinel C (2012) Ipv6 stateless address autoconfiguration: balancing between security, privacy and usability. In: *5th International Symposium on Foundations & Practice of Security (FPS)*, pp 1–14
66. Woodbridge J, Anderson H, Ahuja A, Grant D (2016) Predicting domain generation algorithms with long short-term memory networks, pp 433–448. [arXiv:1611.00791](https://arxiv.org/abs/1611.00791)
67. Liu F, Jia Y, Ren L (2013) Anti-synchronizing different chaotic systems using active disturbance rejection controller based on the chaos particle swarm optimization algorithm. *Acta Phys Sin* 62(12):1–8
68. Yang J, Jin Y (2011) Chaotic based differential evolution algorithm for optimization of baker's yeast drying process. In: *2011 3rd International workshop on intelligent systems and applications (IISAP'11)*, pp 1–8
69. Yuzgec U, Eser M (2018) Hierarchy particle swarm optimization algorithm (hpsso) and its application in multi-objective operation of hydropower stations. *Egypt Inf J* 19(3):151–163
70. Weinmann R, Wirt K (2004) Analysis of the dvb common scrambling algorithm. In: *Proceeding of Conference on Communications and Multimedia. Security*, pp 1–8
71. Xiong G, Zhang J, Yuan X, Shi D, He Y, Yao G Parameter extraction of solar photovoltaic models by means of a hybrid differential evolution with whale optimization algorithm. *Solar Energy* 176
72. Tong W (2020) A hybrid algorithm framework with learning and complementary fusion features for whale optimization algorithm. *Scientific Programming (ID 5684939)*, pp 1–25
73. Azizyan G, Miarnaemi F, Rashki M, Shabakhty N (2019) Flying squirrel optimizer (fso): A novel si-based optimization algorithm for engineering problems. *Iran J Optim* 11(2):177–205
74. Joshi H, Arora (2017) S Enhanced grey wolf optimisation algorithm for constrained optimisation problems. In: *International journal of swarm intelligence*, vol. 3. Taylor & Francis, pp 126–151
75. Bao G, Mao K (2009) Particle swarm optimization algorithm with asymmetric time varying acceleration coefficients. In: *Proceedings of the 2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp 2134–2139
76. Geetha T, Sathya M (2012) Modified particle swarm optimization (mpso) algorithm for web service selection (wss) problem. In: *2012 International Conference on Data Science & Engineering (ICDSE)* (12964092), pp 1–8
77. Kim N, Xiong J, Hwu W (2017) heterogeneous computing meets near-memory acceleration and high-level synthesis in the post-moore era, in: *IEEE Micro*, vol. 37. IEEE, pp 10–18
78. Pilato C, Garg S, Wu K, Karri R, Regazzoni F (2018) Securing hardware accelerators: a new challenge for high-level synthesis. In: *IEEE Embed Syst Lett*, vol. 10. IEEE, pp 77–80
79. Sengupta BS A, Mohanty S (2017) Tl-hls: methodology for low cost hardware trojan security aware scheduling with optimal loop unrolling factor during high level synthesis. In: *IEEE Trans Comput Aided Des Integr Circuits Syst*, vol 36. IEEE, pp 655–668
80. Mohanty S, Ranganathan N, Kougiannos E, Patra P (2008) Low-power high-level synthesis for nanoscale cmos circuits. Springer, Berlin