Engineering with Computers
https://doi.org/10.1007/s00366-020-00996-y

ORIGINAL ARTICLE

Sine cosine grey wolf optimizer to solve engineering design problems

Shubham Gupta[1] · Kusum Deep[2] · Hossein Moayedi[3,4] · Loke Kok Foong[5,6] · Assif Assad[7]

Received: 20 January 2020 / Accepted: 18 February 2020

Abstract

Balancing the exploration and exploitation in any nature-inspired optimization algorithm is an essential task, while solving the real-world global optimization problems. Therefore, the search agents of an algorithm always try to explore the unvisited domains of a search space in a balanced manner. The sine cosine algorithm (SCA) is a recent addition to the field of metaheuristics that finds the solution of an optimization problem using the behavior of sine and cosine functions. However, in some cases, the SCA skips the true solutions and trapped at sub-optimal solutions. These problems lead to the premature convergence, which is harmful in determining the global optima. Therefore, in order to alleviate the above-mentioned issues, the present study aims to establish a comparatively better synergy between exploration and exploitation in the SCA. In this direction, firstly, the exploration ability of the SCA is improved by integrating the social and cognitive component, and secondly, the balance between exploration and exploitation is maintained through the grey wolf optimizer (GWO). The proposed algorithm is named as SC-GWO. For the performance evaluation, a well-known set of benchmark problems and engineering test problems are taken. The dimension of benchmark test problems is varied from 30 to 100 to observe the robustness of the SC-GWO on scalability of problems. In the paper, the SC-GWO is also used to determine the optimal setting for overcurrent relays. The analysis of obtained numerical results and its comparison with other metaheuristic algorithms demonstrate the superior ability of the proposed SC-GWO.

Keywords Exploration and exploitation · Sine cosine algorithm · Grey wolf optimizer · Hybrid algorithms

1 Introduction

From some past decades, the nature-inspired algorithms have shown their superior ability against traditional optimization methods while solving the complex and nonlinear real-world optimization problems. Nature-inspired algorithms are widely used in engineering, science, management, business and various fields to solve their optimization problems. These algorithms have gained much attention because of their flexibility, derivative-free mechanism, easy implementation, better exploration and local optima avoidance potential. Some of the nature-inspired algorithms which are most common and widely used are the genetic algorithm (GA) [1], particle swarm optimization (PSO) [2], artificial bee colony algorithm (ABC) [3] and so on. Some of the recently developed, but efficient algorithms are sine cosine

✉ Hossein Moayedi
hossein.moayedi@tdtu.edu.vn

✉ Loke Kok Foong
lokekokfoong@duytan.edu.vn

Shubham Gupta
sgupta@ma.iitr.ac.in

Kusum Deep
kusumfma@iitr.ac.in

1 Institute for Mega Construction, Korea University, Seoul 02841, South Korea

2 Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand, India

3 Informetrics Research Group, Ton Duc Thang University, Ho Chi Minh City, Vietnam

4 Faculty of Civil Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam

5 Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam

6 Faculty of Civil Engineering, Duy Tan University, Da Nang 550000, Vietnam

7 Department of Computer Science and Engineering, Islamic University of Science and Technology Awantipora, Awantipora, J & K 192122, India

algorithm (SCA) [4], grey wolf optimizer (GWO) [5], moth-flame optimization [6] and so on. Some recently developed, but efficient algorithms include Harris Hawks optimization (HHO) [7, 8], salp swarm algorithm (SSA) [9–11] and grasshopper optimization algorithm (GOA) [9, 12]. Many other scholars also used similar artificial intelligence-based techniques [13–25] or mathematical modeling solutions [26–29].

In the field of nature-inspired algorithms, the "No Free Lunch (NFL)" theorem [30] was the crucial development. According to the NFL theorem, a unique optimization algorithm cannot be developed which is suitable for all the optimization problems. In a simple way, it can be stated that if an optimization algorithm "*A*" is well ideal for some particular set of problems, then there is always a set problem on which the algorithm "*A"* will perform poorly. Thus, the NFL theorem makes the field of nature-inspired algorithm dynamic and allows the researchers to either propose a new algorithm or improve the current existing algorithms. Toward the improvement in existence algorithms, the hybridization of two or more algorithms is a brilliant strategy in which the algorithms are hybridized by combining the best features of various algorithms. By inspiring the advantages of hybridizing two or more algorithms, in the present paper, the SCA and GWO are hybridized to combine the benefits of better exploration and the efficiency of maintaining a balance between exploration and exploitation.

The sine cosine algorithm is a comparatively new nature-inspired optimization algorithm that uses the properties of sine and cosine functions to explore and exploit the search space. Due to its exploration ability, it has been used by many researchers to solve their optimization problems. But, in some cases, the SCA suffers from the problem of low exploitation and skipping of correct solutions due to the presence of high diversity in search equations. Therefore, in the present work, the search strategy of SCA is improved so that the exploration of the search space can be done in more promising directions which are preserved in the memory of the population. After the research in promising areas, an appropriate balance between exploration and exploitation is established by hybridizing the SCA with GWO. The GWO is a recent but widely used and efficient algorithm to solve global optimization problems. In GWO, the search leads by three leading search agents, which maintain the balance between exploration and exploitation in the algorithm. In the paper, the proposed hybrid algorithm is named as SC-GWO. The performance of the SC-GWO is evaluated on 13 well-known benchmark test problems [31] with varying the dimension from 30 to 100 and on well-known engineering test problems. The numerical experimentation and results analysis of the proposed SC-GWO and the comparison with other metaheuristic algorithms through various metrics such as statistical test, convergence curves and scalability of problems show that the SC-GWO is far better than the classical version of SCA and GWO and very competitive to the other algorithms. In the present study, the SC-GWO is also employed on the problem of determining the optimal setting for the coordination of overcurrent relays. The 3-, 4-, 6- and 14-bus systems are used to test the performance of the proposed SC-GWO. The results of this problem also show that the SC-GWO has achieved a comparatively better optimal setting for overcurrent relays than other algorithms.

The rest of the paper is organized as follows: In Sect. 2, the overview of SCA and GWO is presented. Section 3 discusses the proposed SC-GWO algorithm in detail. Numerical experimentation and performance comparison of the SC-GWO with other algorithms are presented in Sect. 4. In Sect. 5, the SC-GWO is employed on five engineering test problems, and the results are compared with various different algorithms. In Sect. 6, the SC-GWO algorithm is used for the coordination of overcurrent relays. Finally, Sect. 6 concludes the paper and suggests some future directions.

2 Preliminaries

In this section, the conceptual description of the sine cosine algorithm (SCA) and grey wolf optimizer (GWO) is provided in detail with a literature review on these algorithms.

2.1 Sine cosine algorithm (SCA)

The SCA [4] is the latest development in the field of nature-inspired algorithms. In SCA, the characteristics of sine and cosine functions are utilized to search for an optimum. To update the search agents in SCA, firstly, a random number $r \sim U(0, 1)$ is generated; then, the following equations are performed

if $r < 0.5$

Table 1 Parameter details for algorithms

Algorithm	Parameter	Value of the parameter
SCA	r_1	$a_0\left(1 - \frac{t}{T}\right), a_0 = 2$
	r_2	$2 \cdot \pi \cdot r$ where $r = rand()$
	r_3	$2 \cdot r$ where $r = rand()$
PSO	c_1	2.0
	c_2	2.0
GWO	a	$a_0\left(1 - \frac{t}{T}\right), a_0 = 2$
	C	$2 \cdot r$ where $r = rand()$
SC-GWO	w	$0.7 - 0.5\left(\frac{t}{T}\right)$
	r_1	$rand()$
	r_2	$rand()$

$$x_{i,j}^{t+1} = x_{i,j}^{t} + r_1 * \sin(r_2) * \left| r_3 x_b^t - x_{i,j}^t \right| \tag{1}$$

else

$$x_{i,j}^{t+1} = x_{i,j}^{t} + r_1 * \cos(r_2) * \left| r_3 x_b^t - x_{i,j}^t \right| \tag{2}$$

end

where $x_{i,j}^{t+1}$ and $x_{i,j}^{t}$ represent the positions of *i*th solution vector (search agents) in *j*th dimension at iterations $t+1$ and t, respectively. x_b is the best available solution in the population. r_1 is a random number that decides the area for exploration or exploitation. The random number r_1 is taken as linearly decreasing variable from the value 2 to 0. The random number r_2 decides the random movement of search agent but either in the direction of the best solution x_b or outwards direction of the best solution. The parameter selection that has been fixed in the original SCA is presented in Table 1.

The exploration in the SCA is achieved by enlarging the range of sine and cosine functions to [−2, 2]. This range is obtained by multiplying sine and cosine functions with a factor r_1. The reposition of search agents either toward the best solution or outwards the best solution guarantees the exploitation and exploration within the algorithm.

Since a proper balance is required between exploration and exploitation in any optimization algorithm to perform a better search, in the SCA this balance is achieved by the random variable r_1, which allows the high exploration in the earlier iterations of algorithm and exploits the discovered promising search areas in the later iterations. The random number r_1 is decreased in the algorithm with the iterations as follows:

$$r_1 = 2 - 2 \cdot \left(\frac{t}{T}\right) \tag{3}$$

where t indicates the current iteration and T represents the maximum iterations. The practical steps of SCA are presented in Algorithm 1.

2.2 Grey wolf optimizer (GWO)

Grey wolf optimizer is a leadership hierarchy-based optimization algorithm and was developed by Mirjalili et al. [5] in 2014. Mirjalili et al. [5] have observed the social and hunting behavior of grey wolves and modeled them into mathematical form to develop an optimization algorithm. In a pack of grey wolves, wolves proceed their search process in three steps, namely tracking, encircling and attacking the prey.

In GWO, the search process is performed based on three leading search agents, which are known as alpha, beta and delta wolves. These leaders are selected by all the wolves of the pack based on dominance and intelligence behavior. The search equations which are used to update the search agents are as follows:

$$x_{i,j}^{t+1} = (x_1 + x_2 + x_3)/3 \tag{4}$$

where

$$x_1 = x_\alpha^t - A_\alpha \cdot D_\alpha \tag{5}$$

$$x_2 = x_\beta^t - A_\beta \cdot D_\beta \tag{6}$$

$$x_3 = x_\delta^t - A_\delta \cdot D_\delta \tag{7}$$

x_α^t, x_β^t and x_δ^t are the states of leaders alpha, beta and delta wolf at *i*th iteration, respectively, and A_α, A_β and A_δ are the random vectors and can be obtained with the help of equations:

$$A = 2ar_1 - a \tag{8}$$

$$a = 2 - 2 \cdot \left(\frac{t}{T}\right) \tag{9}$$

Algorithm 1: pseudo code of SCA

1. Initialize the search agents
2. Evaluate each search agent using the objective function
3. Select the best solution x_b from the set of search agents
4. Initialize the iteration count $t = 0$ and random parameter r_1,
5. while $t <$ T
6. For each search agent
7. Update the position of search agent using solution using equations (1)-(3)
8. Evaluate the updated search agent
9. Update the best solution x_b
10. Update the parameters r_1, r_2 and r_3
11. $t = t + 1$
12. end
13. Return the best solution x_b

where T indicates the maximum iterations. The difference vectors D_α, D_β and D_δ can be obtained by the following equations:

$$D_\alpha = \left| C_\alpha \cdot x_\alpha^t - x_{i,j}^t \right| \tag{10}$$

$$D_\beta = \left| C_\beta \cdot x_\beta^t - x_{i,j}^t \right| \tag{11}$$

$$D_\delta = \left| C_\delta \cdot x_\delta^t - x_{i,j}^t \right| \tag{12}$$

The random number C_α, C_β and C_δ can be calculated by the following equations:

$$C = 2r_2 \tag{13}$$

In this way, the process is performed in GWO. The random numbers a and C help in exploration and exploitation within the algorithm. The random number a maintains suitable synergy between the operators' exploitation and exploration. The random numbers r_1 and r_2 are uniformly distributed in the interval (0, 1). The parameter selection that has been fixed in the original GWO is presented in Table 1.

The determination of a new position based on the leading wolves helps in exploiting the neighborhood regions of the best fitted locations of the search space. The steps of GWO are presented in Algorithm 2.

2.3 Previous work on SCA and GWO

Since the NFL theorem contradicts the existence of an algorithm that is best suited for all optimization problems, there is always a scope to improve an algorithm by enhancing and balancing the operators' exploitation and exploration. In the literature, to increase the search efficiency in SCA, various attempts have been made. In [32], a modified version of the SCA is developed using the orthogonal parallel information. In [33], a modified variant of SCA is developed by integrating opposition-based learning to speed up the convergence rate. In [34], the search mechanism of SCA is improved by addressing the problem of high exploration. The pathological brain detection has been done with modified SCA in [35] which combines the extreme learning machine. In [36], the water wave optimization algorithm and SCA are hybridized to present a better global optimizer. In [37], the SCA is combined with DE to solve the visual tracking problem. In [38], the solution of combinatorial is achieved through a hybrid Q-learning sine–cosine-based strategy. In [39], an orthogonal learning-driven multi-swarm sine cosine optimization algorithm is proposed to improve the global exploration and local exploitation in the SCA. The balance between the exploitation and exploration is tried maintained in the SCA by hybridizing it with ABC [40] and simulated quenching [40].

Like the other nature-inspired algorithms, the GWO also faces the problem of stagnation at sub-optimal solutions and

Algorithm 2: pseudo code of GWO

1. **Initialize** population of grey wolves
2. **Initialize** the parameters a and maximum number of iteration T
3. **Evaluate** the fitness of each grey wolf
4. **Select** the leaders as –
5. x_α – the fittest solution
6. x_β – second-best solution
7. x_δ – third-best solution
8. initialize the iteration count $t = 0$
9. ***while*** $t < T$
10. update each wolf position with the help of equations (4) - (13)
11. **Evaluate** the fitness of each grey wolf
12. **update** the leading wolves x_α, x_β and x_δ
13. **update** the parameter a
14. $t = t + 1$
15. **end while**
16. **Return** the best solution x_α

premature convergence. In the literature, various improvements have been made to enhance its searchability, so that the above-mentioned issues can be avoided from the GWO. In [41], improved GWO (named as IGWO) has been proposed to train q-Gaussian radial basis functional link net neural network. In [42], GWO is hybridized with crossover and mutation to solve economic dispatch problems. In [43], fuzzy hierarchical operators are introduced in GWO to present different modified versions of GWO. The concept of fuzzy logic is also used to improve the search quality and for parameter adaptation in various algorithms [44–47]. In [48], GWO is used for modular granular neural networks for human recognition. In [49], GWO is hybridized with GA to minimize the potential energy of molecules. In [50, 51], the leadership of the grey wolf pack is enhanced by introducing the random walk concept. In [51], Cauchy operator is presented as a crossover operator to enhance the searchability of wolves. In [52], the exploration in GWO is improved by proposing a new position update equation. Opposition-based learning and chaotic maps are also used to enhance the convergence speed and local exploitation in the GWO [53].

In [54], the SCA and GWO are hybridized to extract the exploitation strength of GWO and the exploration strength of SCA. In this hybrid algorithm, called HGWOSCA, the search equation of SCA is employed to update the best solution (alpha wolf) only. Since the NFL theorem allows to improve the search efficiency of the algorithm, they can be applied to a large variety of optimization problems; therefore, in the present paper, the hybrid algorithm called SC-GWO is proposed. This SC-GWO first enhances the exploration in the SCA and then tries to balance exploitation and research by the hybridization of GWO. This hybrid version is different in structure from the HGWOSCA [54]. A detailed description of the proposed SC-GWO algorithm is presented as follows.

3 Proposed SC-GWO algorithm

In this section, firstly, the motivation behind the hybridization of SCA and GWO is discussed in detail. Secondly, the framework of the proposed algorithm is presented along with the pseudo-code.

3.1 Motivation

Although the SCA has been used by many researchers for their optimization purposes due to its excellent exploration ability in some cases, it shows low exploitation and inappropriate balance between exploration and exploitation. The skipping of correct solutions during the search procedure of the SCA may cause the problem of little exploitation and weak local searchability. On the other hand, GWO is useful in balancing exploration and exploitation during the search procedure. However, similar to other metaheuristics, GWO also stagnates at local optima in some situations, where a large number of local optima are present. Therefore, in the present study, a new hybrid method called SC-GWO is introduced, which tries to alleviate all these issues. In the SC-GWO, first, the search mechanism of the SCA is modified based on the social and cognitive components. Then, the hybridization of this new search mechanism is performed with the GWO to produce a comparatively better balance of exploitation and exploration. A detailed description of the proposed SC-GWO algorithm is as follows.

3.2 SC-GWO algorithm

The proposed SC-GWO algorithm extends the original SCA by adding three features: (1) addition of social and cognitive direction, (2) weighted factor for the position component achieved from unique search mechanism of the SCA to balance the diversity and (3) integration of GWO phase to maintain the balance between exploration and exploitation. The following subsections will elaborate on all these modifications.

3.2.1 Social and cognitive components

In the proposed SC-GWO algorithm, the social and cognitive components are added in the search equation, so that each search agent can contribute to the search process and can explore the regions which are more promising and present in a neighborhood of the best memory of search agents. These components limit the search agents around the personal best and global best memory so that the promising regions near the best available memory of search agents can be exploited.

3.2.2 The weighted factor for the position component of SCA

In the proposed algorithm SC-GWO, the weighted factor is introduced for the position component obtained by the original search mechanism of the SCA. The weighted factor is added to control the high diversity during the search so that the skipping of true solutions can be avoided.

The modified search equations based on the above two features are:

$$x_{i,j}^{t+1} = w * x_{i,j}^{t+1} + r_1 * \left(x_{ip,j}^{t} - x_{i,j}\right) + r_2 * \left(x_{b,j}^{t} - x_{i,j}^{t}\right) \quad (14)$$

where w is a weight factor that controls diversity during the search. In the algorithm, it has been fixed in the interval [0.7, 0.2] and decreased linearly over the iterations. r_1 and r_2 are the uniformly distributed random numbers between 0 and 1. x_{ip}^{t} and x_{b}^{t} represented the personal best position of ith candidate solution and global best solutions at tth iteration, respectively, and x_i^{t+1} is the updated position of the candidate solution x_i at iteration, $t + 1$ using original SCA.

3.2.3 Integration of GWO phase

In the proposed SC-GWO algorithm, when the exploration of new promising regions is performed using original SCA and above proposed factors, the GWO phase is completed for each search agent. This phase maintains a sufficient balance between exploration and exploitation during the search. The GWO phase also allows the search to be performed around the best solutions so that the best available memory of the population can be exploited. In GWO phases, the search agents, which are scattered to explore new promising areas of the search space, update their positions and attain new positions with the help of the best solutions of the population. The search procedure of the proposed SC-GWO algorithm is presented in Algorithm 3.

4 Results and analysis

In this section, the proposed SC-GWO algorithm has been tested and analyzed on 13 well-known benchmark problems. In this set, the first 7 problems are unimodal, while the remaining are multimodal in nature. The population size of search agents is taken to 30, and the termination criteria are considered to 1000 maximum iterations. The other parameters involved in the SC-GWO are shown in Table 1. Since the SC-GWO evaluates the search agents twice in each iteration, for original SCA, PSO, GWO and HGWOSCA [54], the maximum number of iterations is fixed to 2000 for a fair comparison in an aspect of function evaluation cost. The set of test problems are presented in Table 2. The results on these problems, by conducting 30 trials of each algorithm, are presented in Tables 3, 4 and 5. In these tables, results are presented for 30-, 50- and 100-dimensional problems, respectively,

Algorithm 3: pseudo-code of the proposed SC-GWO algorithm

Set the parameters: *population size* (NP), *maximum number of iterations* (T)
Initialize the population $P = [x_{i,j}]$ *of search agents*
calculate the fitness at each search agent
Initialize the inertia weight w
Select the best solution x_b *from the population of search agents*
Set the loop counter $it = 0$
while $it < T$
  Update the inertia weight w
    for $i = 1{:}NP$
      update the position of each search agent using equation (1), (2) *and* (14)
      calculate the fitness of updated search agent x'_i
      update the personal best sultion x_{ip}
    end of for
    select the three best solutions as alpha, beta and delta from the set
    $\{x_b\} \cup P'$ *for GWO phase, where* $P' = [x'_{i,j}]$
    Set $P = P'$
%%%%%%%%   GWO phase   %%%%%%%%%
    for $i = 1{:}NP$
      update the position of search agent x_i *using equation* (4)
      calculate the fitness at updated search agent x'_i
      update the personal best sultion x_{ip}
    end of for
  update the best solution x_b
  $P = [x'_{i,j}]$
  $it = it + 1$
end of while
return the best solution x_b

to observe the robustness of the proposed SC-GWO algorithm on the scalability of test problems. In the table, the best, median, average, standard deviation (STD) and worst value of objective functions recorded over 30 runs are reported.

The performance of the SC-GWO algorithm on unimodal test functions shows the better exploitation ability as compared to the other algorithms. In all the unimodal test problems, the SC-GWO beats the classical version of SCA in all the metrics such as best, median, average, standard deviation and worst value of the objective function. In most of the test problems, the SC-GWO is either competes or performs better than GWO, PSO and HGWOSCA. In multimodal problems ($F8$–$F13$), the proposed algorithm also provides very competitive results with the PSO, GWO and HGWOSCA and outperforms original SCA in all the problems. Overall, in terms of exploitation and exploration, the SC-GWO shows their better efficacy as compared to the other algorithm such as SCA, PSO, GWO and HGWOSCA.

In order to show that the acquired results are not just by chance, a nonparametric Wilcoxon signed-rank test is applied at a 5% level of significance. In Tables 3, 4 and 5, the obtained statistical decision is presented by the signs "$+/=/-$" that indicates that the proposed SC-GWO is significantly better, similar or worse than its competitive algorithm. The overall summary of statistical decisions is presented in Table 6. From this table, it can be concluded that the proposed SC-GWO algorithm beats original SCA in terms of accuracy in obtaining the results in all the problems. The comparison against PSO, GWO and HGWOSCA also shows that the proposed SC-GWO algorithm is significantly better than these algorithms in most of the test problems.

4.1 Convergence behavior analysis

In order to observe the convergence behavior of algorithms, the convergence curves are plotted in Figs. 1 and 2 corresponding to the unimodal and multimodal test problems. In these figures, the iterations are depicted on the x-axis and the objective function values have been shown on the y-axis. The convergence curves are plotted by considering the median value of the objective function obtained over 30 trials. The curves are plotted by fixing the population of the search agent to 30 and a maximum number of iterations to 500. From the convergence curves, it can be observed that in all the unimodal problems, the SC-GWO algorithm shows a better convergence rate than PSO, SCA and GWO. For the multimodal problems, the proposed algorithm has shown their better ability in terms of convergence rate except for the function $F8$. In function $F8$, the convergence rate in PSO and GWO is better than the proposed algorithm. Overall, the proposed algorithm can be considered as a better optimizer in terms of convergence rate.

4.2 Comparison with recent optimization algorithms

In the present section, the proposed SC-GWO algorithm is compared with some recent optimization algorithms. The comparison with PSO, classical SCA, classical GWO and HGWOSCA has been performed earlier that shows that the proposed SC-GWO algorithm has comparatively better search efficiency to locate the optima of the problems.

In the present section, the performance of the proposed SC-GWO is compared with some of the recently developed algorithms such as modified GWO (mGWO) [55], weighted GWO (wGWO) [43], SSA [11], opposition-based sine cosine algorithm (OBSCA) [33], modified sine cosine algorithm (m-SCA) [34], improved sine cosine algorithm (ISCA) [34], whale optimization algorithm (WOA) [56], moth-flame optimization (MFO) [6] and chaotic salp swarm algorithm (chaotic SSA) [57]. For a fair comparison, the number of iterations is fixed to 1000 for each algorithm and 30 population size is adopted to implement the algorithm on test problems. From the results recorded in Table 7, it can be observed that in the problems $F1$, $F2$, $F3$, $F4$, $F9$ and $F11$, the proposed algorithm is able to locate the optima. In problems $F1$, $F2$, $F3$ and $F4$, only the proposed algorithm is able to find the optima. In problems $F9$ and $F11$, along with the proposed SC-GWO algorithm, ISCA and WOA also provide the optima. In the problems $F7$, the SC-GWO is superior than all other algorithms. In $F9$ and $F10$, the SC-GWO and ISCA perform equally and outperform different algorithms. In problems $F5$, $F6$, $F12$, $F13$, the proposed SC-GWO is very competitive to the other algorithms. Thus, from the comparison of results recorded in Table 7, the proposed SC-GWO algorithm can be considered as a better optimizer than different algorithms.

5 Applications of SC-GWO on real engineering problems

This section evaluates the search efficiency of the proposed SC-GWO in solving five real engineering design problems, namely gear train design, three-bar truss design, speed reducer design, tension/compression spring design and pressure vessel design problems. The obtained results by the SC-GWO are also compared with several other metaheuristic algorithms. Since most of these engineering design problems are constrained in nature, to deal with the constraints, the concept of constraint violation [58] is used. In this approach, the best solution from the population is elected

Table 2 Details of benchmark problems

Test problem	Range of search space	$F_{\min}$
$F1(x) = \sum_{i=1}^{d} x_i^2$	[− 100,100]	0
$F2(x) = \sum_{i=1}^{d} \lvert x_i \rvert + \prod_{i=1}^{d} \lvert x_i \rvert$	[− 10,10]	0
$F3(x) = \sum_{i=1}^{d} \left(\sum_{j-1}^{i} x_i \right)^2$	[− 100,100]	0
$F4(x) = \max_i \left\{ \lvert x_i \rvert, 1 \leq i \leq d \right\}$	[− 100,100]	0
$F5(x) = \sum_{i=1}^{d-1} \left[100 \cdot \left(x_{i+1} - x_i^2 \right)^2 + (x_i - 1)^2 \right]$	[− 30,30]	0
$F6(x) = \sum_{i=1}^{d} \left(\left[x_i + (1/2) \right] \right)^2$	[− 100,100]	0
$F7(x) = \sum_{i=1}^{d} i \cdot x_i^4 + rand[0, 1)$	[− 1.280, 1.280]	0
$F8(x) = \sum_{i=1}^{d} -x_i \sin \left(\sqrt{\lvert x_i \rvert} \right)$	[− 500,500]	$-418.9829 \times d$
$F9(x) = \sum_{i=1}^{d} \left[x_i^2 - 10cos\left(2\pi x_i\right) + 10 \right]$	[− 5.12, 5.12]	0
$F10(x) = \sum_{i=1}^{d} -20 \exp \left(-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^{d} x_i^2} \right) - \exp \left(\frac{1}{d} \sum_{i=1}^{d} \cos \left(2\pi x_i \right) \right) + 20 + e$	[− 32, 32]	0
$F11(x) = \frac{1}{4} \times 10^{-3} \sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} cos\left(x_i / \sqrt{i} \right) + 1)$	[− 600, 600]	0
$F12(x) = \frac{\pi}{d} \left\{ 10 \sin \left(\pi y_1 \right) + \sum_{i=1}^{d-1} \left(y_i - 1 \right)^2 \left[1 + 10 \sin^2 \left(\pi y_{i+1} \right) \right] + \left(y_n - 1 \right)^2 \right\} + \sum_{i=1}^{d} u\left(x_i, 10, 100, 4 \right)$ $y_i = \frac{x_i + 5}{4}$ $u\left(x_i, b, c, m \right) = \begin{cases} c\left(x_i - b \right)^m & \text{if } x_i > b \\ c\left(-x_i - b \right)^m & \text{if } x_i < -b \\ 0 & \text{otherwise} \end{cases}$	[− 50, 50]	0
$F13(x) = 0.1 \left\{ \sin^2 \left(3\pi x_1 \right) + \sum_{i=1}^{d} \left(x_i - 1 \right)^2 \left[1 + \sin^2 \left(1 + 3\pi x_i \right) \right] + \left(x_d - 1 \right)^2 \left[1 + \sin^2 \left(2\pi x_n \right) \right] \right\} + \sum_{i=1}^{d} u\left(x_i, 5, 100, 4 \right)$	[− 50, 50]	0

using the constraint violation value of each solution. For an optimization problem:

$$\begin{array}{lll} \boldsymbol{Min} & f(x), & x = \left(x_1, x_2, \ldots, x_d \right) \in \boldsymbol{R}^d \\ \boldsymbol{s.t.} & g_j(x) \leq 0 & j = 1, 2, \ldots, m \\ & h_k(x) = 0 & k = 1, 2, \ldots, n \\ & l_i \leq x_i \leq u_i & i = 1, 2, \ldots, d \end{array} \tag{15}$$

where f, g_j and h_k represent the objective function, inequality constraints and equality constraints, respectively, l_i and u_i are the lower and upper limits for a i^{th} component of a decision variable x, and the constraint violation of a solution $\hat{x}$ can be calculated as follows:

$$\text{viol}_{\hat{x}} = \sum_{j=1}^{m} G_j(x) + \sum_{k=1}^{n} H_k(x) \tag{16}$$

where

Table 3 Comparison of best, median, average, worst and standard deviation of objective function values corresponding to the dimension 30

Test problem	Algorithm	Best	Median	Average	Worst	STD	Wilcoxon test
*F*1	SCA	1.13E−15	1.17E−09	3.03E−08	4.91E−07	9.45E−08	+
	PSO	4.56E−23	1.79E−21	1.24E−20	1.01E−19	2.39E−20	+
	GWO	8.09E−126	6.98E−123	2.94E−122	2.04E−121	5.51E−122	+
	HGWOSCA	1.42E−137	3.81E−135	4.58E−133	1.23E−131	2.23E−132	+
	SC-GWO (present study)	**0**	**0**	**0**	**0**	**0**	
*F*2	SCA	1.31E−15	9.17E−13	1.31E−10	3.76E−09	6.86E−10	+
	PSO	1.25E−14	3.20E−13	3.88E−12	7.36E−11	1.34E−11	+
	GWO	6.64E−72	3.78E−71	6.33E−71	4.15E−70	7.93E−71	+
	HGWOSCA	8.08E−82	8.39E−81	7.23E−80	1.69E−78	3.06E−79	+
	SC-GWO (present study)	**0**	**0**	**0**	**0**	**0**	
*F*3	SCA	8.76E+00	3.42E+02	1.35E+03	6.66E+03	2.03E+03	+
	PSO	3.66E+00	3.86E+01	7.06E+02	1.00E+04	2.17E+03	+
	GWO	1.53E−42	3.02E−36	5.09E−33	1.34E−31	2.45E−32	+
	HGWOSCA	3.26E−35	1.99E−30	9.82E−26	2.88E−24	5.25E−25	+
	SC-GWO (present study)	**0**	**0**	**0**	**0**	**0**	
*F*4	SCA	9.67E−01	4.33E+00	6.81E+00	2.44E+01	5.90E+00	+
	PSO	8.30E−01	1.76E+00	1.83E+00	2.96E+00	6.63E−01	+
	GWO	1.30E−31	2.04E−30	9.90E−30	8.89E−29	2.00E−29	+
	HGWOSCA	2.14E−29	1.82E−26	8.47E−26	1.32E−24	2.41E−25	+
	SC-GWO (present study)	**0**	**0**	**0**	**0**	**0**	
*F*5	SCA	2.69E+01	2.81E+01	3.08E+01	1.11E+02	1.51E+01	+
	PSO	**5.14E−01**	4.94E+01	2.47E+02	3.04E+03	7.58E+02	+
	GWO	2.52E+01	**2.62E+01**	**2.65E+01**	2.87E+01	9.07E−01	=
	HGWOSCA	2.52E+01	2.67E+01	2.67E+01	2.87E+01	8.78E−01	–
	SC-GWO (present study)	2.59E+01	2.71E+01	2.68E+01	**2.80E+01**	**5.95E−01**	
*F*6	SCA	3.43E+00	4.13E+00	4.09E+00	4.63E+00	3.06E−01	+
	PSO	**1.43E−22**	**2.63E−21**	**1.84E−20**	**1.48E−19**	**3.50E−20**	–
	GWO	2.69E−06	6.22E−01	5.47E−01	1.00E+00	3.18E−01	=
	HGWOSCA	3.34E−06	5.08E−01	6.43E−01	1.75E+00	3.98E−01	–
	SC-GWO (present study)	2.50E−01	7.47E−01	7.09E−01	1.75E+00	3.22E−01	
*F*7	SCA	1.97E−03	8.28E−03	1.53E−02	1.53E−01	2.69E−02	+
	PSO	6.25E−03	1.43E−02	1.45E−02	2.82E−02	4.70E−03	+
	GWO	6.68E−05	4.31E−04	4.19E−04	9.24E−04	2.39E−04	+
	HGWOSCA	8.66E−05	4.54E−04	5.03E−04	1.36E−03	2.52E−04	+
	SC-GWO (present study)	**1.54E−05**	**4.84E−05**	**4.94E−05**	**1.29E−04**	**2.30E−05**	
*F*8	SCA	−4.49E+03	−3.94E+03	−4.00E+03	−3.59E+03	**2.73E+02**	+
	PSO	**−1.02E+04**	**−9.03E+03**	**−9.00E+03**	**−7.42E+03**	5.74E+02	–
	GWO	−7.50E+03	−6.15E+03	−6.12E+03	−4.81E+03	6.30E+02	=
	HGWOSCA	−7.23E+03	−6.13E+03	−5.98E+03	−4.68E+03	6.88E+02	+
	SC-GWO (present study)	−7.96E+03	−6.26E+03	−5.66E+03	−3.21E+03	1.57E+03	
*F*9	SCA	5.68E−14	6.45E−04	5.24E+00	5.47E+01	1.41E+01	+
	PSO	2.19E+01	3.98E+01	4.06E+01	7.37E+01	1.14E+01	+
	GWO	**0**	**0**	**0**	**0**	**0**	=
	HGWOSCA	**0**	**0**	**0**	1.72E+01	4.29E+00	+
	SC-GWO (present study)	**0**	**0**	**0**	**0**	**0**	
*F*10	SCA	2.64E−07	2.01E+01	1.51E+01	2.02E+01	7.83E+00	+
	PSO	7.61E−13	2.09E−11	4.53E−11	2.39E−10	6.35E−11	+
	GWO	7.99E−15	7.99E−15	8.82E−15	1.51E−14	2.02E−15	+
	HGWOSCA	8.88E−16	8.88E−16	8.88E−16	8.88E−16	0	=
	SC-GWO (present study)	**8.88E−16**	**8.88E−16**	**8.88E−16**	**8.88E−16**	**0**	

Table 3 (continued)

Test problem	Algorithm	Best	Median	Average	Worst	STD	Wilcoxon test
*F*11	SCA	5.80E−14	8.32E−05	6.53E−02	4.06E−01	1.26E−01	+
	PSO	**0**	9.86E−03	1.43E−02	7.37E−02	1.86E−02	+
	GWO	**0**	**0**	1.01E−03	1.29E−02	3.14E−03	=
	HGWOSCA	**0**	**0**	2.00E−03	2.33E−02	5.49E−03	+
	SC-GWO (present study)	**0**	**0**	**0**	**0**	**0**	
*F*12	SCA	3.08E−01	4.67E−01	5.80E−01	1.63E+00	2.92E−01	+
	PSO	**5.30E−25**	**5.03E−22**	**3.80E−02**	4.15E−01	8.81E−02	=
	GWO	6.55E−03	2.67E−02	3.87E−02	1.27E−01	2.99E−02	=
	HGWOSCA	1.87E−02	4.65E−02	5.13E−02	1.08E−01	2.28E−02	+
	SC-GWO (present study)	1.32E−02	3.79E−02	4.07E−02	**1.01E−01**	**2.29E−02**	
*F*13	SCA	1.78E+00	2.29E+00	2.30E+00	2.96E+00	2.22E−01	+
	PSO	**1.16E−23**	**2.34E−21**	**3.30E−03**	**4.39E−02**	**8.73E−03**	−
	GWO	2.00E−01	5.03E−01	5.11E−01	7.97E−01	1.78E−01	=
	HGWOSCA	3.24E−01	6.73E−01	6.85E−01	1.08E+00	1.86E−01	+
	SC-GWO (present study)	7.58E−02	5.72E−01	5.49E−01	1.02E+00	2.47E−01	

$$G_j(x) = \begin{cases} g_j(x) & \text{if } g_j(x) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

and

$$H_k(x) = \begin{cases} |h_k(x)| & |\text{if } |h_k(x)| - \in > 0 \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

here $\in$ is a predefined tolerance parameter. The description of engineering design problems and the discussion of the results are as follows:

6 Design of gear train problem

This problem is introduced initially by Sandgren [59] with the objective of determining the optimal number of teeth for gear so that the gear ratio can be minimized. This problem is a discrete case study of decision variables y_1, y_2, y_3 and y_4. Mathematically, the problem is stated as follows:

$$\begin{aligned} &\boldsymbol{Min}\ f_1(y) = \left(\frac{1}{6.931} - \frac{y_2 y_3}{y_1 y_4}\right)^2, \text{ where } y = (y_1, y_2, y_3, y_4) \\ &\boldsymbol{s.t.}\quad 12 \le y_1, y_2, y_3, y_4 \le 60 \end{aligned}$$

The ratio $\frac{y_2 y_3}{y_1 y_4}$ represents the gear ratio. The gear trains are shown in Fig. 3. This problem has been solved using the proposed SC-GWO with 30 independent runs. The recorded results are presented in Table 8. In the table, the results of proposed SC-GWO algorithm are compared with some state-of-the-art algorithms such as GWO [5], PSO [2], wPSO [60], ABC [3], cuckoo search (CS) [61], GA [62], augmented Lagrange multiplier (ALM) [63]. For a fair comparison, the results of SC-GWO are also compared with other algorithms that are taken in Sect. 4.2. From the table, it can be easily analyzed that the proposed algorithm SC-GWO is not able to provide a better solution but also consumes less function evaluation effort. Thus, the table recommends using the SC-GWO over other algorithms to solve the gear train design problem.

7 Design of three-bar truss problem

This case study considers a three-bar planar truss structure, which is shown in Fig. 4. Initially, this problem was introduced by Nowacki [64] with the aim of minimizing the volume of a statically loaded three-bar truss. The constraints are applied to the stress of each truss element. Mathematically, the problem is stated as follows:

$$\begin{aligned} &\boldsymbol{Min}\ f_2(y_1, y_2) = L \times \left(2\sqrt{2y_1} + y_2\right) \\ &\boldsymbol{s.t.}\quad g_1(y_1, y_2) = \frac{\sqrt{2}y_1 + y_2}{\sqrt{2}y_1^2 + 2y_1 y_2} R \le \sigma \\ &\qquad g_2(y_1, y_2) = \frac{y_2}{\sqrt{2}y_1^2 + 2y_1 y_2} R \le \sigma \\ &\qquad g_3(y_1, y_2) = \frac{1}{y_1 + \sqrt{2}y_2} R \le \sigma \\ &\qquad 0 \le y_1, y_2 \le 1 \end{aligned}$$

where

$$L = 100\,\text{c.m.}, \quad R = 2\,\text{KN/cm}^2 \text{ and } \sigma = 2\,\text{KN/cm}^2$$

Table 4 Comparison of best, median, average, worst and standard deviation of objective function values corresponding to the dimension 50

Test problem	Algorithm	Best	Median	Average	Worst	STD	Wilcoxon test
*F*1	SCA	4.79E−06	2.33E−02	4.02E+00	1.08E+02	1.98E+01	+
	PSO	1.56E−10	1.17E−08	3.52E−08	3.60E−07	7.48E−08	+
	GWO	6.70E−94	4.10E−92	1.44E−90	3.50E−89	6.40E−90	+
	HGWOSCA	2.77E−103	5.18E−100	4.83E−99	3.13E−98	8.71E−99	+
	SC-GWO (present study)	**0**	**0**	**0**	**0**	**0**	
*F*2	SCA	8.32E−09	8.50E−07	2.77E−05	6.28E−04	1.15E−04	+
	PSO	3.10E−07	1.18E−05	1.33E+00	1.00E+01	3.46E+00	+
	GWO	5.02E−55	5.85E−54	9.38E−54	4.85E−53	9.86E−54	+
	HGWOSCA	3.77E−62	5.83E−61	1.37E−60	8.49E−60	2.13E−60	+
	SC-GWO (present study)	**0**	**0**	**0**	**0**	**0**	
*F*3	SCA	5.80E+03	1.95E+04	2.45E+04	7.70E+04	1.54E+04	+
	PSO	1.59E+03	6.53E+03	8.67E+03	4.63E+04	8.83E+03	+
	GWO	1.53E−23	2.81E−20	1.22E−16	2.55E−15	4.98E−16	+
	HGWOSCA	1.37E−18	3.36E−13	2.99E−08	8.94E−07	1.63E−07	+
	SC-GWO (present study)	**0**	**0**	**0**	**0**	**0**	
*F*4	SCA	2.60E+01	4.82E+01	4.82E+01	6.91E+01	9.44E+00	+
	PSO	8.70E+00	1.11E+01	1.09E+01	1.40E+01	1.34E+00	+
	GWO	1.48E−21	6.03E−20	3.47E−19	2.22E−18	6.19E−19	+
	HGWOSCA	6.94E−17	3.36E−15	1.24E−07	3.70E−06	6.75E−07	+
	SC-GWO (present study)	**0**	**0**	**0**	**0**	**0**	
*F*5	SCA	5.97E+01	3.30E+03	6.01E+04	6.13E+05	1.39E+05	+
	PSO	**3.11E+01**	9.92E+01	3.11E+03	9.01E+04	1.64E+04	+
	GWO	4.54E+01	**4.71E+01**	**4.70E+01**	**4.86E+01**	**7.60E−01**	=
	HGWOSCA	4.52E+01	4.70E+01	4.69E+01	4.84E+01	8.19E−01	=
	SC-GWO (present study)	4.55E+01	**4.71E+01**	**4.70E+01**	**4.86E+01**	8.84E−01	
*F*6	SCA	8.67E+00	9.61E+00	1.04E+01	1.67E+01	1.89E+00	+
	PSO	**8.15E−10**	**1.15E−08**	**5.00E−08**	**4.07E−07**	**1.03E−07**	−
	GWO	1.25E+00	2.25E+00	2.22E+00	3.26E+00	5.35E−01	=
	HGWOSCA	9.81E−01	2.50E+00	2.44E+00	3.50E+00	5.73E−01	=
	SC-GWO (present study)	1.44E+00	2.48E+00	2.50E+00	3.50E+00	6.16E−01	
*F*7	SCA	3.92E−03	6.90E−02	1.29E−01	9.91E−01	1.97E−01	+
	PSO	3.01E−02	5.93E−02	5.79E−02	1.04E−01	1.60E−02	+
	GWO	1.86E−04	6.11E−04	6.67E−04	2.25E−03	4.16E−04	+
	HGWOSCA	2.01E−04	6.86E−04	7.47E−04	1.43E−03	3.25E−04	+
	SC-GWO (present study)	**2.72E−06**	**2.98E−05**	**3.81E−05**	**1.52E−04**	**3.20E−05**	
*F*8	SCA	−5.63E+03	−5.25E+03	−5.26E+03	−4.83E+03	**2.29E+02**	+
	PSO	**−1.63E+04**	**−1.40E+04**	**−1.40E+04**	**−1.23E+04**	8.39E+02	−
	GWO	−1.13E+04	−9.19E+03	−9.39E+03	−7.86E+03	9.37E+02	−
	HGWOSCA	−1.16E+04	−8.84E+03	−8.94E+03	−5.17E+03	1.45E+03	−
	SC-GWO (present study)	−1.09E+04	−8.75E+03	−7.91E+03	−3.84E+03	2.59E+03	
*F*9	SCA	3.08E−04	3.16E+01	3.77E+01	1.37E+02	3.96E+01	+
	PSO	4.68E+01	1.02E+02	1.05E+02	1.52E+02	2.23E+01	+
	GWO	**0**	**0**	1.89E−15	5.68E−14	1.04E−14	=
	HGWOSCA	**0**	**0**	6.12E+00	4.73E+01	1.25E+01	+
	SC-GWO (present study)	**0**	**0**	**0**	**0**	**0**	
*F*10	SCA	1.01E−03	2.04E+01	1.70E+01	2.05E+01	7.71E+00	+
	PSO	7.57E−06	2.88E−04	4.88E−01	1.47E+00	6.18E−01	+
	GWO	1.51E−14	1.51E−14	1.58E−14	2.22E−14	2.17E−15	+
	HGWOSCA	8.88E−16	8.88E−16	8.88E−16	8.88E−16	0	=
	SC-GWO (present study)	**8.88E−16**	**8.88E−16**	**8.88E−16**	**8.88E−16**	**0**	

Table 4 (continued)

Test problem	Algorithm	Best	Median	Average	Worst	STD	Wilcoxon test
*F*11	SCA	3.43E−05	4.36E−01	4.29E−01	1.97E+00	4.22E−01	+
	PSO	9.49E−10	2.35E−07	6.56E−03	3.44E−02	9.30E−03	+
	GWO	**0**	**0**	1.68E−03	1.30E−02	3.89E−03	=
	HGWOSCA	**0**	**0**	1.07E−03	1.83E−02	4.12E−03	+
	SC-GWO (present study)	**0**	**0**	**0**	**0**	**0**	
*F*12	SCA	1.22E+00	1.85E+01	1.53E+05	4.31E+06	7.86E+05	+
	PSO	**1.90E−10**	**4.49E−06**	8.10E−02	6.24E−01	1.48E−01	=
	GWO	3.56E−02	7.00E−02	8.83E−02	3.92E−01	6.71E−02	=
	HGWOSCA	4.52E−02	1.04E−01	1.15E−01	4.00E−01	6.29E−02	+
	SC-GWO (present study)	3.73E−02	7.13E−02	**7.38E−02**	**1.24E−01**	**2.41E−02**	
*F*13	SCA	4.82E+00	5.50E+02	5.66E+05	1.10E+07	2.04E+06	+
	PSO	**2.62E−07**	**1.10E−02**	**1.55E−01**	**1.61E+00**	4.27E−01	–
	GWO	9.56E−01	1.73E+00	1.70E+00	2.56E+00	**3.67E−01**	–
	HGWOSCA	1.51E+00	2.04E+00	2.08E+00	2.94E+00	3.14E−01	+
	SC-GWO (present study)	1.11E+00	1.86E+00	1.94E+00	2.50E+00	4.25E−01	

To find a solution to this problem, various studies have been done in the literature. This problem is solved using the same function evaluations as used in [61]. The recorded results on 30 independent trials are shown in Table 9. In the same table, the comparison of the SC-GWO is performed with other algorithms such as PSO [2], wPSO [60] and CS [61]. In the tables, the results of two other studied by Ray and Saini [65] and Tsai [66] are also presented. The table also compares the performance of the proposed SC-GWO algorithm with novel algorithms that are used in Sect. 4.2 for comparison. It can be analyzed from the table that the SC-GWO is able to provide a comparatively better solution than other algorithms.

8 Design of speed reducer problem

The speed reducer design problem introduced in [61] is a crucial benchmark design problem, which is shown in Fig. 5. The goal of this problem is to minimize the weight of the speed reducer. The decision variables involved in this problem are "face width (W)," "module of teeth (M)," "number of teeth on pinion (N)," "length of shaft 1 between bearing (L_1)," "length of shaft 2 between bearing (L_2)," "diameter of shaft 1 (D_1)" and "diameter of shaft 2 (D_2)." The constraints are applied on stress in shaft 1 and shaft 2, bending stress of gear teeth, surface stress, transverse deflections of shaft 1 and shaft 2 due to the effect of transmitted force. The mathematical expression of this problem is stated as follows:

$$
\begin{aligned}
Min \quad & f_3(y) = 0.7854y_1y_2^2\left(14.9334y_3 + 3.3333y_3^2 - 43.0934\right) \\
& \quad + 7.4777\left(y_6^3 + y_7^3\right) - 1.508y_1\left(y_6^2 + y_7^2\right) \\
& \quad + 0.7854\left(y_4y_6^2 + y_5y_7^2\right) \\
& x = \left(y_1, y_2, y_3, y_4, y_5, y_6, y_7\right) = \left(W, M, N, L_1, L_2, D_1, D_2\right) \\
s.t. \quad & g_1(y) = \frac{27}{y_1y_2^2y_3} \le 1 \\
& g_2(y) = \frac{397.5}{y_1y_2^2y_3^2} \le 1 \\
& g_3(y) = \frac{1.93y_5^3}{y_2y_3y_7^4} \le 1 \\
& g_4(y) = \frac{1.93y_4^3}{y_2y_3y_6^4} \le 1 \\
& g_5(y) = \frac{\sqrt{1.57\times10^8 + \left(\frac{745y_5}{y_2y_3}\right)^2}}{85y_7^3} \le 1 \\
& g_6(y) = \frac{\sqrt{1.69\times10^7 + \left(\frac{745y_4}{y_2y_3}\right)^2}}{110y_6^3} \le 1 \\
& g_7(y) = \frac{y_2y_3}{40} \le 1 \\
& g_8(y) = \frac{y_1}{12y_2} \le 1 \\
& g_9(y) = \frac{5y_2}{y_1} \le 1 \\
& g_{10}(y) = \frac{1.5y_6+1.9}{y_4} \le 1 \\
& g_{11}(y) = \frac{1.1y_7+1.9}{y_5} \le 1 \\
& 2.6 \le y_1 \le 3.6 \\
& 0.7 \le y_2 \le 0.8 \\
& 17 \le y_3 \le 28 \\
& 7.3 \le y_4 \le 8.3 \\
& 7.8 \le y_5 \le 8.3 \\
& 2.9 \le y_6 \le 3.9 \\
& 5 \le y_7 \le 5.5
\end{aligned}
$$

Table 5 Comparison of best, median, average, worst and standard deviation of objective function values corresponding to the dimension 100

Test problem	Algorithm	Best	Median	Average	Worst	STD	Wilcoxon test
*F*1	SCA	1.93E+02	1.39E+03	2.01E+03	7.10E+03	1.86E+03	+
	PSO	1.32E−01	5.58E−01	1.04E+00	7.35E+00	1.41E+00	+
	GWO	3.30E−65	1.06E−63	2.90E−63	1.73E−62	4.15E−63	+
	HGWOSCA	2.85E−69	8.28E−68	1.59E−67	9.63E−67	2.34E−67	+
	SC-GWO (present study)	**0**	**0**	**0**	**0**	**0**	
*F*2	SCA	1.89E−05	3.04E−02	1.14E−01	1.94E+00	3.49E−01	+
	PSO	7.11E−02	2.02E+01	1.87E+01	6.10E+01	1.65E+01	+
	GWO	2.74E−38	9.31E−38	1.29E−37	5.75E−37	1.27E−37	+
	HGWOSCA	7.48E−44	5.39E−43	1.06E−42	6.56E−42	1.37E−42	+
	SC-GWO (present study)	**0**	**0**	**0**	**0**	**0**	
*F*3	SCA	1.07E+05	1.70E+05	1.75E+05	2.69E+05	3.96E+04	+
	PSO	3.87E+04	6.78E+04	6.76E+04	1.20E+05	1.78E+04	+
	GWO	4.38E−09	5.14E−06	1.73E−02	4.27E−01	7.79E−02	+
	HGWOSCA	6.73E−03	1.25E+00	6.15E+00	9.31E+01	1.75E+01	+
	SC-GWO (present study)	**0**	**0**	**0**	**0**	**0**	
*F*4	SCA	6.77E+01	8.22E+01	8.08E+01	8.98E+01	4.63E+00	+
	PSO	2.50E+01	2.87E+01	2.91E+01	3.36E+01	2.33E+00	+
	GWO	2.01E−11	1.23E−09	2.48E−08	3.49E−07	7.68E−08	+
	HGWOSCA	1.56E−05	2.33E+01	2.24E+01	5.13E+01	1.52E+01	+
	SC-GWO (present study)	**0**	**0**	**0**	**0**	**0**	
*F*5	SCA	3.82E+06	1.97E+07	2.60E+07	7.90E+07	1.97E+07	+
	PSO	4.85E+02	7.19E+02	1.02E+03	3.90E+03	7.66E+02	+
	GWO	**9.51E+01**	**9.77E+01**	**9.75E+01**	**9.84E+01**	9.07E−01	=
	HGWOSCA	9.57E+01	9.76E+01	9.74E+01	9.85E+01	7.86E−01	=
	SC-GWO (present study)	9.58E+01	**9.77E+01**	**9.75E+01**	9.85E+01	**8.09E−01**	
*F*6	SCA	7.97E+01	1.23E+03	2.10E+03	9.85E+03	2.44E+03	+
	PSO	**2.89E−01**	**9.35E−01**	**2.57E+00**	3.92E+01	7.11E+00	−
	GWO	7.86E+00	9.36E+00	9.49E+00	**1.15E+01**	**7.46E−01**	−
	HGWOSCA	6.99E+00	8.97E+00	9.13E+00	1.16E+01	9.96E−01	−
	SC-GWO (present study)	7.86E+00	1.02E+01	1.00E+01	1.27E+01	1.07E+00	
*F*7	SCA	1.96E+00	2.54E+01	3.84E+01	1.48E+02	3.55E+01	+
	PSO	2.71E−01	4.42E−01	9.62E−01	1.12E+01	2.19E+00	+
	GWO	3.49E−04	8.38E−04	1.11E−03	2.69E−03	6.72E−04	+
	HGWOSCA	5.00E−04	1.44E−03	1.45E−03	3.00E−03	6.09E−04	+
	SC-GWO (present study)	**7.20E−06**	**3.07E−05**	**3.96E−05**	**1.09E−04**	**3.08E−05**	
*F*8	SCA	− 8.66E+03	− 7.64E+03	− 7.64E+03	− 6.68E+03	**5.21E+02**	+
	PSO	**− 2.65E+04**	**− 2.35E+04**	**− 2.37E+04**	**− 2.16E+04**	1.17E+03	−
	GWO	− 1.96E+04	− 1.65E+04	− 1.58E+04	− 6.49E+03	2.77E+03	−
	HGWOSCA	− 2.32E+04	− 1.68E+04	− 1.66E+04	− 1.35E+04	1.74E+03	−
	SC-GWO (present study)	− 1.95E+04	− 1.56E+04	− 1.28E+04	− 5.38E+03	5.49E+03	
*F*9	SCA	5.67E+01	1.54E+02	1.68E+02	2.87E+02	6.81E+01	+
	PSO	2.02E+02	2.73E+02	2.78E+02	3.52E+02	4.55E+01	+
	GWO	**0**	**0**	3.79E−15	1.14E−13	2.08E−14	=
	HGWOSCA	**0**	**0**	1.27E+00	1.28E+01	3.47E+00	+
	SC-GWO (present study)	**0**	**0**	**0**	**0**	**0**	
*F*10	SCA	3.47E−01	2.06E+01	1.94E+01	2.06E+01	4.63E+00	+
	PSO	1.40E+00	2.29E+00	2.35E+00	3.20E+00	4.15E−01	+
	GWO	1.87E−14	2.93E−14	2.88E−14	3.29E−14	3.58E−15	+
	HGWOSCA	8.88E−16	8.88E−16	8.88E−16	8.88E−16	0.00E+00	=
	SC-GWO (present study)	**8.88E−16**	**8.88E−16**	**8.88E−16**	**8.88E−16**	**0**	
*F*11	SCA	2.74E+00	1.36E+01	1.91E+01	6.49E+01	1.68E+01	+
	PSO	7.39E−02	3.33E−01	3.63E−01	8.02E−01	1.54E−01	+

Table 5 (continued)

Test problem	Algorithm	Best	Median	Average	Worst	STD	Wilcoxon test
	GWO	**0**	**0**	9.06E−04	1.36E−02	3.45E−03	=
	HGWOSCA	0.00E+00	0.00E+00	1.36E−03	1.68E−02	4.26E−03	+
	SC-GWO (present study)	**0**	**0**	**0**	**0**	**0**	
*F*12	SCA	2.82E+05	8.66E+07	9.25E+07	2.28E+08	6.89E+07	+
	PSO	1.34E+00	3.84E+00	3.82E+00	6.54E+00	1.22E+00	+
	GWO	1.74E−01	2.36E−01	2.49E−01	3.98E−01	6.00E−02	=
	HGWOSCA	1.25E−01	2.41E−01	2.53E−01	3.96E−01	6.23E−02	+
	SC-GWO (present study)	**1.54E−01**	**2.28E−01**	**2.36E−01**	**3.82E−01**	**4.92E−02**	
*F*13	SCA	1.04E+07	8.71E+07	1.21E+08	5.81E+08	1.27E+08	+
	PSO	4.57E+01	9.36E+01	1.13E+02	5.37E+02	8.69E+01	+
	GWO	**5.02E+00**	**6.05E+00**	**6.05E+00**	**6.79E+00**	**4.21E−01**	–
	HGWOSCA	5.62E+00	6.58E+00	6.58E+00	7.35E+00	4.81E−01	=
	SC-GWO (present study)	5.51E+00	6.63E+00	6.58E+00	7.21E+00	4.70E−01	

Table 6 Summary of statistical results obtained by Wilcoxon test

Dimension	SC-GWO vs SCA	SC-GWO vs PSO	SC-GWO vs GWO	SC-GWO vs HGWOSCA
30	13/0/0	9/1/3	6/7/0	10/1/2
50	13/0/0	9/1/3	6/5/2	9/3/1
100	13/0/0	11/0/2	6/4/3	8/3/2

The results obtained after the 30 independent trials of SC-GWO are shown in Table 10 using the same function evaluations as used in [61]. The results' comparison from other algorithms is also shown in the same table which demonstrates the superior search efficiency of the proposed SC-GWO than different algorithms.

9 Design of compression spring problem

The goal in this problem [70, 71] is to attain a minimum weight of a compression spring by determining the optimal values of variables, namely "wire diameter (d), "mean coil diameter (D)" and "number of active coils (N)." The constraints applied to the objective function are stress, surge frequency and deflection. Mathematical expression for this problem is stated as follows:

$$\begin{aligned} &\boldsymbol{Min}\ f_4(y) = \left(y_3 + 2\right) y_2 y_1^2 \\ &\boldsymbol{s.t.}\quad g_1(y) = 1 - \frac{y_2^3 y_3}{71785 y_1^4} \le 0 \\ &\qquad g_2(y) = \frac{4y_2^2 - y_1 y_2}{12566\left(y_2 y_1^3 - y_1^4\right)} + \frac{1}{5108 y_1^2} - 1 \le 0 \\ &\qquad g_3(y) = 1 - \frac{140.45 y_1}{y_2^2 y_3} \le 0 \\ &\qquad g_4(y) = \frac{y_1 + y_2}{1.5} - 1 \le 0 \\ &\qquad 0.05 \le y_1 \le 2, 0.25 \le y_2 \le 1.30, 2 \le y_3 \le 15. \end{aligned}$$

This problem is solved using PSO [2], gravitational search algorithm (GSA) [72], random walk grey wolf optimizer (RW-GWO) [50], GA (Coello) [73], PSO (He and Wang) [74], evolution strategy (ES) (Coello and Montes) [69], ray optimization (RO) [75] algorithms and by Arora [70] and Belegundu [71]. The optimal weight obtained by the proposed algorithm is compared with these studies in Table 11 by performing the 30 trials of an algorithm and using the same cost of function evaluations as used in [50]. In the table, the comparison is also performed with recent algorithms such as SCA, GWO, MFO, WOA, mGWO, wGWO, m-SCA, ISCA, OBSCA, SSA and chaotic SSA. The table clearly verifies the better performance of proposed SC-GWO algorithm than other algorithms.

10 Design of pressure vessel problem

The goal of this problem is to minimize the total cost in terms of material, forming and welding of the cylindrical pressure vessel as shown in Fig. 6. The ends of the vessel are capped while the head has a hemispherical shape. In these problems, the decision parameters are "thickness of the shell $\left(T_{\mathrm{SH}}\right)$," "thickness of the head $\left(T_{\mathrm{HD}}\right)$," "inner radius ($R$)" and "the length of the cylindrical shell (L)" without considering the head. In these problems, four constraints are involved. The mathematical formulation of the problem is defined as follows:

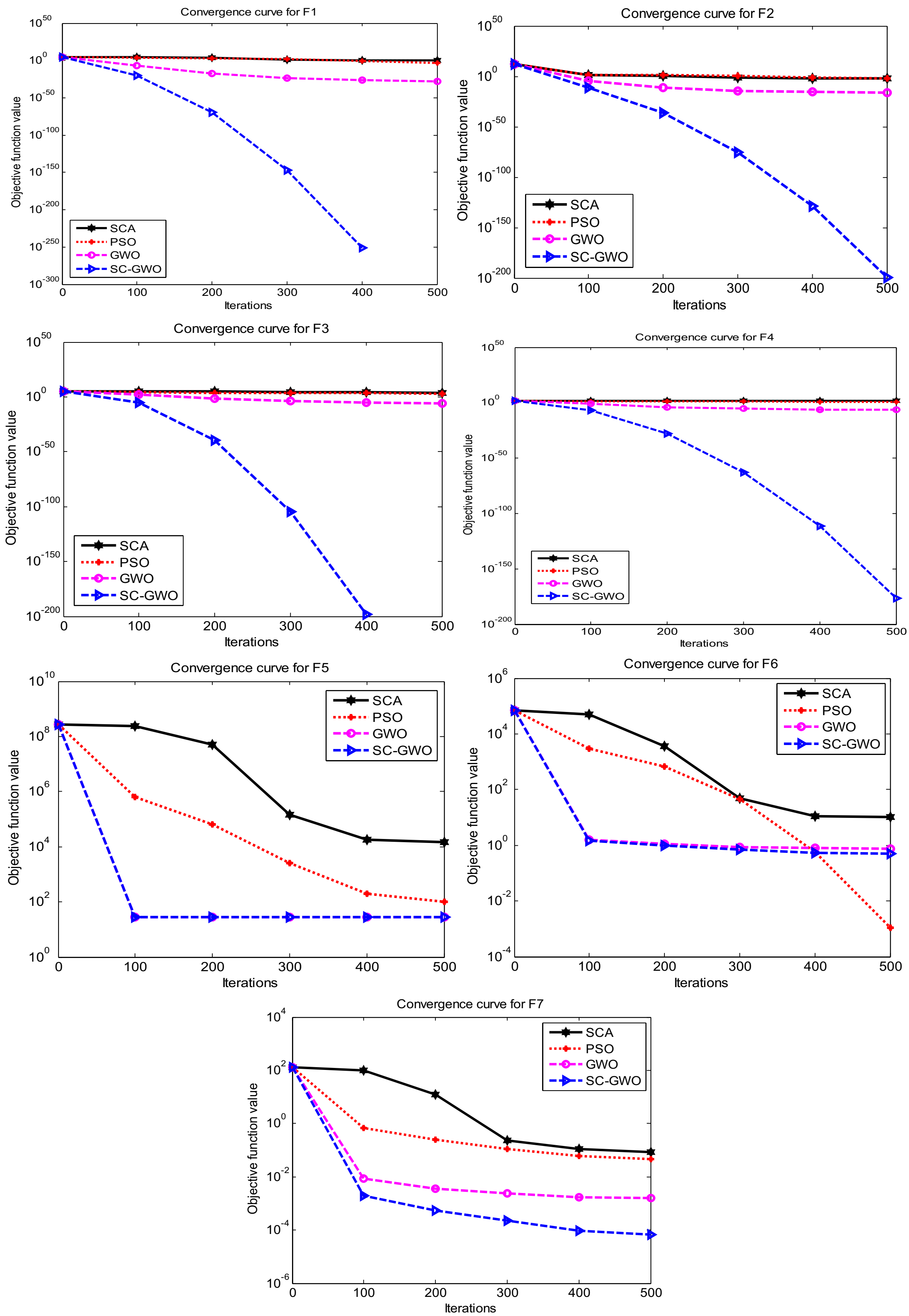

Fig. 1 Convergence curves for unimodal test functions corresponding to the dimension 30

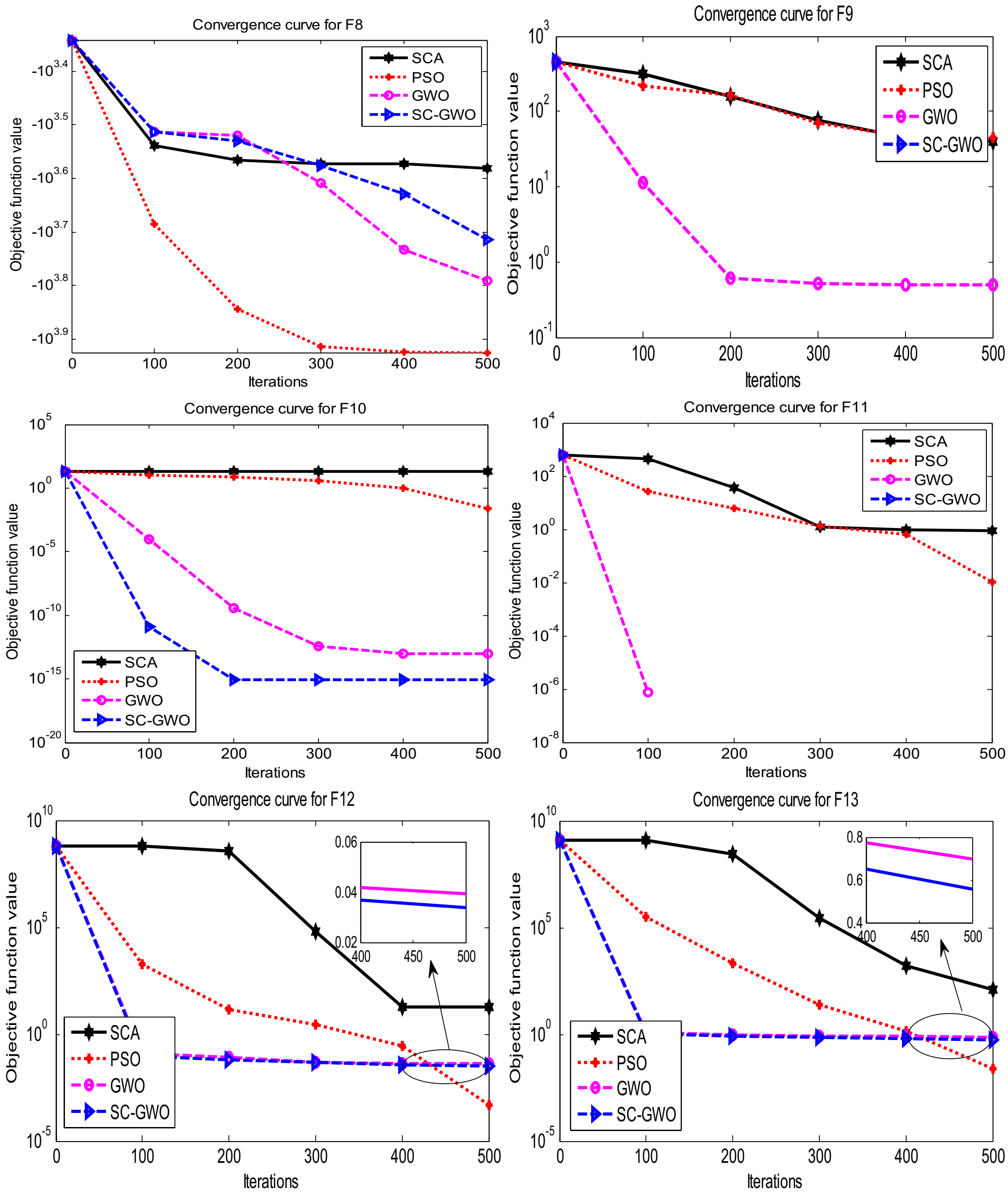

Fig. 2 Convergence curves for multimodal test functions corresponding to the dimension 30

$$
\begin{aligned}
&\textit{Min}\ f_5(y) = 0.6224y_1y_3y_4 + 1.7781y_2y_3^2 + 19.84y_1^2y_3 + 3.1661y_1^2y_4\\
&\qquad y = (y_1, y_2, y_3, y_4) = (T_{\mathrm{SH}}, T_{\mathrm{HD}}, R, L)\\
&s.t.\quad g_1(y) = 0.0193y_3 - y_1 \le 0\\
&\qquad g_2(y) = 0.00954y_3 - y_2 \le 0\\
&\qquad g_3(y) = 1296000 - \tfrac{4}{3}\pi y_3^3 - \pi y_3^2 y_4 \le 0\\
&\qquad g_4(y) = y_4 - 240 \le 0\\
&\qquad 1 \times 0.0625 \le y_1, y_2 \le 99 \times 0.0625\\
&\qquad 10 \le y_3, y_4 \le 200
\end{aligned}
$$

In order to find the optimal cost, the SC-GWO algorithm is implemented 30 times on this problem using the same function evaluations as used in [61] and the recorded results are shown in Table 12. To compare the results obtained by proposed algorithm, the results of some state-of-the-art algorithms such as GA, DE, ACO, PSO, ES, GWO, SCA, GSA, branch and bound method, Lagrangian multiplier method

Table 7 Comparison of mean objective function values between proposed SC-GWO and some recent algorithms

Test problem	mGWO	wGWO	OBSCA	m-SCA	ISCA	WOA	MFO	SSA	Chaotic SSA	SC-GWO
F1	6.80E−76	1.36E−60	1.88E−18	8.25E−02	2.01E−58	1.40E−148	2.33E+03	1.24E−08	1.90E−08	**0.00E+00**
F2	6.41E−45	4.21E−35	3.66E−17	2.13E−05	1.87E−36	9.90E−102	3.83E+01	1.54E+00	1.78E+00	**0.00E+00**
F3	1.91E−18	5.60E−14	1.35E+01	3.84E+03	2.52E−04	2.45E+04	1.75E+04	2.58E+02	3.54E+02	**0.00E+00**
F4	4.06E−20	2.39E−14	7.07E−01	2.11E+01	2.07E−15	4.17E+01	6.82E+01	8.73E+00	1.29E+01	**0.00E+00**
F5	2.64E+01	2.69E+01	2.83E+01	6.01E+03	**2.61E+01**	2.72E+01	5.34E+06	1.38E+02	1.02E+02	2.68E+01
F6	5.55E−01	6.40E−01	4.60E+00	4.64E+00	1.56E−01	4.06E−02	2.00E+03	**1.39E−08**	1.82E−08	7.09E−01
F7	7.10E−04	8.37E−04	2.15E−03	6.95E−02	1.13E−03	1.72E−03	3.99E+00	1.04E−01	9.92E−02	**4.94E−05**
F8	− 5.97E+03	− 6.08E+03	− 3.76E+03	− 3.95E+03	− 7.67E+03	**− 1.15E+04**	− 8.67E+03	− 7.49E+03	− 7.53E+03	− 5.66E+03
F9	1.68E−01	3.42E−01	5.26E−11	2.07E+01	**0.00E+00**	0.00E+00	1.75E+02	5.79E+01	6.15E+01	**0.00E+00**
F10	8.82E−15	1.51E−14	1.27E+01	1.64E+01	**8.88E−16**	4.80E−15	1.75E+01	2.04E+00	2.38E+00	**8.88E−16**
F11	6.82E−04	2.53E−03	4.34E−11	4.13E−01	0.00E+00	0.00E+00	2.11E+01	8.62E−03	7.30E−03	**0.00E+00**
F12	3.39E−02	6.49E−02	4.94E−01	1.58E+03	1.00E−02	**3.01E−03**	7.33E−01	5.58E+00	8.00E+00	4.07E−02
F13	4.26E−01	6.46E−01	2.46E+00	4.45E+03	**1.69E−01**	2.82E−01	4.40E−01	2.58E+00	7.34E+00	5.49E−01

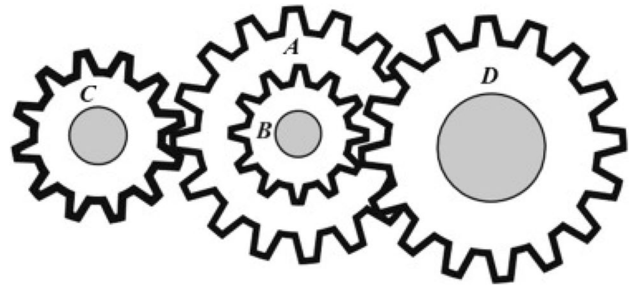

Fig. 3 Gear train design problem [61]

and some recent algorithms which are used for comparison in Sect. 4.2 are also presented in the same table. From the table, it can be observed that the proposed algorithm is better than all other reported algorithms.

11 Application of SC-GWO for finding the optimal setting of overcurrent relays

The relay coordination is a nonlinear and highly constrained optimization task, where the aim is to achieve the minimum operating time of all the relays. Overcurrent relays are protection devices [80] and can sense the flow of current only in one direction. These relays should operate for a fault for the functioning of the power system properly. Proper coordination of direction overcurrent relays is a very crucial task for the better performance of the power systems and to escape from the problem of equipment damages, and this relay coordination is a very time-consuming and tedious task. The protection of directional overcurrent relays consists of two types of setting—TDS (time dial setting) and PS (plug setting)—and by obtaining the optimal values of these settings, efficient coordination for overcurrent relays can be established. In the literature, various metaheuristics such as genetic algorithm (GA) [1], particle swarm optimization (PSO) [2] and differential evolution (DE) [81] are implemented to obtain an efficient solution for the coordination of overcurrent relays. In [82–84], differential evolution along with its modified variants is employed on the coordination of overcurrent relay problem to find the optimum setting of decision parameters TDS and PS. In [85], a random search technique is applied to find the decision parameters so that the operating time of all relays can be minimized.

11.1 Problem formulation

The relay operating time (T) of directional overcurrent relay is a nonlinear function that consists of decision parameters, namely variables TDS (time dial setting), PF (power flow)

Table 8 Results' comparison for the design of gear train problem

Algorithm	Optimal solution				$f_{1,\min}$	No. of function evaluations
	y_1	y_2	y_3	y_4		
SC-GWO	**43**	**16**	**19**	**49**	**2.7009E−12**	**500**
GWO	57	31	13	49	9.9398E−11	500
PSO	34	13	20	53	2.3078E−11	500
wPSO	54	17	22	48	1.1661E−10	820
SCA	51	15	26	53	2.3078E−11	500
ABC	44	16	19	49	1.0742E−05	40,000
CS	43	16	19	49	**2.7009E−12**	5000
ALM	33	13	15	41	2.4070E−08	NA
GA	33	14	17	50	1.3620E−09	NA
mGWO	59	15	21	37	3.0676E−10	500
wGWO	54	37	12	57	8.8876E−10	500
m-SCA	53	13	20	34	2.3078E−11	820
OBSCA	52	15	30	60	2.3576E−09	820
SSA	57	13	31	49	9.9399E−11	820
MFO	51	16	23	50	1.1834E−09	820
WOA	55	14	17	30	1.3616E−09	820
ISCA	47	25	16	59	9.7457E−10	500
Chaotic SSA	46	26	12	47	9.9216E−10	500

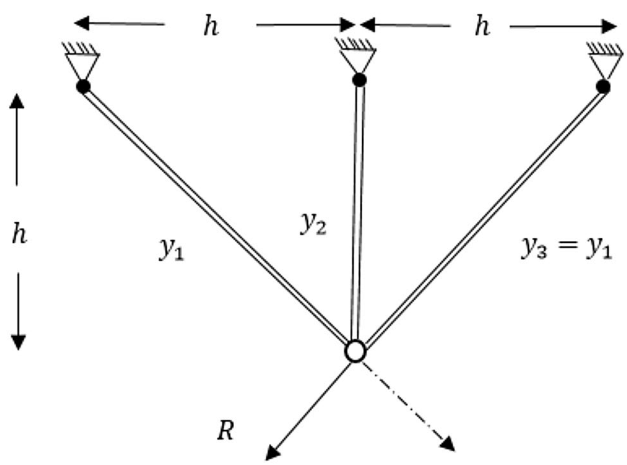

Fig. 4 Three-bar truss design [61]

and PS (plug setting). Mathematically, the operating time [86, 87] can be represented as:

$$T = \frac{\alpha \times \mathrm{TDS}}{\left(\frac{i_f}{\mathrm{PS} \times \mathrm{CT}_{\mathrm{pr\_rating}}}\right)^{\gamma} - \beta} \tag{19}$$

In the above equation, the constants α, γ, and β represent the characteristics of a relay and are chosen as 0.14, 0.02 and 1.0, respectively [88]. I_f is the fault current passing through a relay. $\mathrm{CT}_{\mathrm{pr\_rating}}$ is current transformer's primary rating. If the secondary rating of CT is 1.0, then the current sensed by the relay is

Table 9 Results' comparison for the design of three-bar truss problem

Algorithm	Decision variables		Objective function value ($f_{2,\min}$)
	y_1	y_2	
SC-GWO	**0.78941**	**0.40617**	**263.8963**
SCA	0.78394	0.42219	263.9506
PSO	0.86263	0.32534	263.8986
wPSO	0.58959	0.20568	263.8994
GWO	0.54083	0.41266	263.9497
CS	0.78867	0.40902	263.9716
Ray and Saini [65]	0.795	0.395	264.3
Tsai [66]	0.788	0.408	263.68 (infeasible)
mGWO	0.78953	0.40585	263.8967
wGWO	0.78919	0.40678	263.8964
m-SCA	0.79491	0.39113	263.9481
OBSCA	0.77457	0.46647	263.9463
SSA	0.78869	0.4082	263.8958
MFO	0.78901	0.44025	267.1922
WOA	0.79995	0.37725	263.9858
ISCA	0.79084	0.40217	263.9002
Chaotic SSA	0.78901	0.44025	267.1922

$$I_r = \frac{I_f}{\mathrm{CT}_{\mathrm{pr\_rating}}} \tag{20}$$

A fault that occurs close to relay is called near-end fault or close-in fault, and the fault which occurs at other end is called far-end fault or far-bus fault for the same relay. N_c represents the number of close-in fault, and N_f is the number of far-bus faults. The objective function of the problem is the addition of operating time of all primary relays which can be represented as follows:

$$O_f = \sum_{j=1}^{N_c} T^j_{\mathrm{pr\_cl\_in}} + \sum_{k=1}^{N_f} T^k_{\mathrm{pr\_far\_bus}} \tag{21}$$

where

$$T^j_{\mathrm{pr\_cl\_in}} = \frac{0.14 \times \mathrm{TDS}^j}{\left(\frac{i^j_f}{\mathrm{PS}^j \times \mathrm{CT}^j_{\mathrm{primary\ rating}}}\right)^{0.02} - 1} \tag{22}$$

and

$$T^k_{\mathrm{pr\_far\_bus}} = \frac{0.14 \times \mathrm{TDS}^k}{\left(\frac{i^k_f}{\mathrm{PS}^k \times \mathrm{CT}^k_{\mathrm{primary\ rating}}}\right)^{0.02} - 1} \tag{23}$$

are response time of relay j and k to clear close-in and far-bus fault, respectively.

11.2 Constraints of the problem

11.2.1 Bound constraints for variable TDS

The bound constraints for TDS variable are defined as:

$$\mathrm{TDS}^j_{\min} \le \mathrm{TDS}^j \le \mathrm{TDS}^j_{\max} \quad \forall j = 1, 2, \ldots, N_c \tag{24}$$

The values of $\mathrm{TDS}^j_{\min}$ and $\mathrm{TDS}^j_{\max}$ for each j are fixed as 0.05 s and 1.10 s.

11.2.2 Bound constraints for variable PS

The bound constraints for TDS variable are defined as:

$$\mathrm{PS}^j_{\min} \le \mathrm{PS}^j \le \mathrm{PS}^j_{\max} \quad \forall j = 1, 2, \ldots, N_f \tag{25}$$

The values of $\mathrm{PS}^j_{\min}$ and $\mathrm{PS}^j_{\max}$ for each j are fixed as 1.25 and 1.5.

11.2.3 Constraints for primary operation time

Every term of the objective function (O_f) should lie in the interval [0.05, 1].

11.2.4 Selectivity constraints for relay pairs

The backup relays handle the situation where primary relays fail to avoid the mal-operation. The constraints which maintain the selectivity of primary and backup relays are as follows:

$$\mathrm{TM} = T_{\mathrm{backup}} - T_{\mathrm{primary}} \ge \mathrm{CTI} \tag{26}$$

or

$$\mathrm{CTI} - T_{\mathrm{backup}} + T_{\mathrm{primary}} \le 0 \tag{27}$$

where

$$T^j_{\mathrm{primary}} = \frac{0.14 \times \mathrm{TDS}^m}{\left(\frac{i^j_f}{\mathrm{PS}^j \times \mathrm{CT}^j_{\mathrm{primary\ rating}}}\right)^{0.02} - 1} \tag{28}$$

and

$$T^j_{\mathrm{backup}} = \frac{0.14 \times \mathrm{TDS}^n}{\left(\frac{i^j_f}{\mathrm{PS}^j \times \mathrm{CT}^j_{\mathrm{primary\ rating}}}\right)^{0.02} - 1} \tag{29}$$

where (m, n) is the combination of a primary relay and backup relay n corresponding to primary relay m. The value of *CTI* is fixed and often set between 0.2 and 0.6 s.

11.3 Experimental details with results

In this study, the proposed hybrid method called SC-GWO is employed to solve the coordination overcurrent relay problem. The size of the population in any metaheuristics plays an essential and crucial role. The small size leads to less exploration, and the large population size may fail to determine the optimal of the problem because of the low exploitation. Therefore, in the present work, an equivalent size 10 times of the decision parameters is taken as population size for each of the bus systems.

To compare the optimal setting obtained by proposed method, random search technique (RST) [85], differential evolution (DE) [83], improved variants of differential

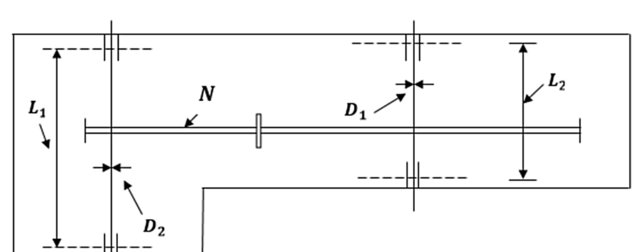

Fig. 5 Speed reducer [61]

Table 10 Results' comparison for the design of the speed reducer problem

Algorithm	Optimal decision parameters							Optimum value ($f_{3,\min}$)
	y_1	y_2	y_3	y_4	y_5	y_6	y_7	
SC-GWO	**3.50064**	**0.7**	**17**	**7.30643**	**7.80617**	**3.35034**	**5.28694**	**2996.9859**
PSO	3.58147	0.7	17.8282	7.98445	7.82083	3.15398	5.1873	3005.3248
GWO	3.6	0.8	28	7.3	8.3	2.9	5.0	3020.2331
SCA	3.51889	0.7	17	7.3	8.3	3.35899	5.30519	3028.8657
wPSO	3.50662	0.7	17	7.44735	7.88468	3.2725	5.38998	3003.7983
CS	3.50150	0.7	17	7.6050	7.8181	3.3520	5.2875	3000.9810
Ray and Saini [65]	3.51418	0.700005	17	7.497343	7.8346	2.9018	5.0022	2732.9006 (infeasible)
Akhtar et al. [67]	3.50612	0.700006	17	7.549126	7.85933	3.36558	5.289773	3008.08
Ku et al. [68]	3.6	0.7	17	7.3	7.8	3.4	5	2876.1176 (infeasible)
Montes and Coello [69]	3.50616	0.700831	17	7.46018	7.962143	3.3629	5.3090	3025.005
mGWO	3.50128	0.7	17	7.34965	7.80177	3.35087	5.28712	2997.7748
wGWO	3.50008	0.7	17	7.3193	7.81168	3.35072	5.28692	2997.085
m-SCA	3.52394	0.7	17	7.3	7.8	3.36280	5.32467	3033.2845
OBSCA	3.00576	0.72755	21.8423	7.30835	8.15455	3.36452	5.25164	3027.5130
MFO	3.59093	0.70554	19.7972	8.08267	7.84181	3.70621	5.48167	3836.2164
WOA	3.52111	0.7	17	7.3	7.8	3.35021	5.29533	3010.1480
SSA	3.50031	0.7	17	7.80001	7.85001	3.35247	5.2867	3002.5678
ISCA	3.50081	0.7	17	7.3	7.8	3.35129	5.28698	2997.1295
Chaotic SSA	3.50031	0.7	17	7.80001	7.85001	3.35247	5.2867	3002.5678

Table 11 Results' comparison for the design of compression string problem

Algorithms	Optimal decision vector			$f_{4,\min}$
	y_1	y_2	y_3	
SC-GWO	**0.051511**	**0.352376**	**11.552600**	**0.012672**
GWO	0.051550	0.353220	11.504050	0.012675
PSO	0.051728	0.357644	11.244543	0.012675
SCA	0.050822	0.335616	12.717800	0.012758
GSA	0.050276	0.323680	13.525410	0.012702
RW-GWO	0.051670	0.356130	11.330560	0.012674
GA	0.051480	0.351660	11.632200	0.012705
PSO (He and Wang)	0.051728	0.357644	11.244543	0.012675
ES	0.051989	0.363965	10.890522	0.012681
RO	0.051370	0.349096	11.76279	0.0126788
Constraint correction [70]	0.050000	0.315900	14.2500	0.012833
Mathematical optimization [71]	0.053400	0.399180	9.185400	0.012730
mGWO	0.0511667	0.344199	12.067	0.012676
wGWO	**0.0514616**	**0.35127**	**11.6221**	**0.012672**
m-SCA	0.0515532	0.353314	11.5518	0.012725
OBSCA	1.01280	0.54065	9.2424	0.012824
MFO	0.050822	0.335616	12.717800	0.012758
WOA	0.051207	0.345215	12.004032	0.012676
SSA	0.051207	0.345215	12.004032	0.012676
ISCA	0.052855	0.38523	9.8008	0.01270
Chaotic SSA	0.051285	0.347075	11.8777	0.012668

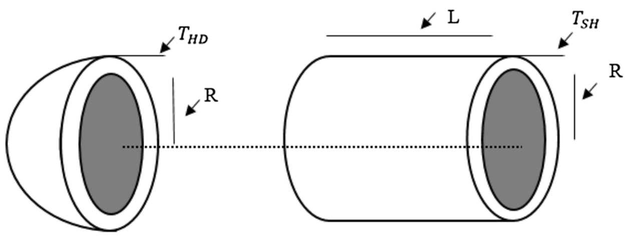

Fig. 6 Pressure vessel design [61]

evolution such as MDE1, MDE2, MDE3, MDE4, MDE5 [84], OCDE1 [82], OCDE2 [82], LX-PM [87] have been considered.

11.3.1 Test models: IEEE 3-Bus, 4-Bus, 6-Bus and IEEE 14-Bus system

In the present work, the 3-bus, 4-bus, 6-bus, and 14-bus systems are used to evaluate the performance of the proposed algorithm (SC-GWO). In the 3-bus system, the total number of decision parameters is 12 which are $\mathrm{TDS}^1 - \mathrm{TDS}^6$ and $\mathrm{PS}^1 - \mathrm{PS}^6$. In this model, eight selectivity constraints are involved. The 4-bus system contains 16 decision variables, namely $\mathrm{TDS}^1 - \mathrm{TDS}^8$ and $\mathrm{PS}^1 - \mathrm{PS}^{16}$. In the 4-bus system, the number of selectivity constraints is 9. In the 6-bus system, 28 decision variables $\mathrm{TDS}^1 - \mathrm{TDS}^{14}$ and $\mathrm{PS}^1 - \mathrm{PS}^{28}$ are involved. In this model, 48 selectivity constraints are involved corresponding to all possible near-end and far-end faults that are sensed by relays of the power system. In a 14-bus system, 80 decision variables ($\mathrm{TS}^1 - \mathrm{TS}^{40}$ and $\mathrm{PS}^1 - \mathrm{PS}^{40}$) are involved. The data for the 14-bus system can be found in [86, 87]. In this model, 145 selectivity constraints are involved corresponding to all possible near-end and far-end faults that are sensed by relays of the power system.

The input values of $I_{\mathrm{f}}^j, \mathrm{CT}_{\mathrm{pr\_rating}}^j, I_{\mathrm{f}}^k$ and $\mathrm{CT}_{\mathrm{pr\_rating}}^k$ for 3-bus, 4-bus and 6-bus systems can be accessed from [86, 87]. The values of CTI for 3-bus, 4-bus systems are fixed as 0.3 and for 6 and 14-bus systems; it is fixed as 0.2. For the 3-, 4- and 6-bus systems, values of I_{f} and $\mathrm{CT}_{\mathrm{pr\_rating}}$ corresponding to primary and backup relay can be accessed from [86, 87].

The obtained best optimal decision variables are presented in Tables 13, 14, 15 and 16, and corresponding objective function value (minimum operating time of all relays) is shown in Table 17 for all the bus systems. In Table 17, the comparison of the proposed SC-GWO algorithm is also performed with some recent algorithms like SCA [4], m-SCA [34], ISCA [34], OBSCA [33], GWO [5], mGWO [55], SSA [11], chaotic SSA [57] and WOA [56]. From Table 17, it

Table 12 Results' comparison for the design of pressure vessel problem

Algorithm	Optimum decision variables				$f_{5,\min}$
	y_1	y_2	y_3	y_4	
SC-GWO	0.8125	0.4375	42.0984	176.6370	6059.7179
GWO	0.8750	0.4375	44.9807	144.1081	6136.6600
PSO	0.8125	0.4375	42.0913	176.7465	6061.0777
SCA	0.8125	0.4375	42.0486	177.7078	6076.3651
GSA	1.1250	0.6250	55.9887	84.4542	8538.8360
PSO [74]	0.8125	0.4375	42.0913	176.7465	6061.0780
GA [73]	0.8125	0.4345	40.3239	20.0000	6288.7450
GA [76]	0.9375	0.5000	48.3290	112.6790	6410.3810
DE [77]	0.8125	0.4375	42.0984	176.6376	6059.7340
ACO [78]	0.8125	0.4375	42.1036	176.5727	6059.0888 (infeasible)
ES [69]	0.8125	0.4375	42.0980	176.6405	6059.7456
Branch and Bound [79]	1.1250	0.6250	47.7000	117.7010	8129.1040
Lagrangian Multiplier [63]	1.1250	0.6250	58.2910	43.6900	7198.043
mGWO	0.8125	0.4375	42.09829	176.6386	6059.7359
wGWO	0.8125	0.4375	42.09842	176.637	6059.7207
m-SCA	0.8125	0.4375	42.10389	176.5944	6060.0282
OBSCA	3.0000	0.8125	82.80484	175.3546	6528.3323
MFO	**0.8125**	**0.4375**	**42.0984**	**176.6366**	**6059.7143**
WOA	0.8125	0.4375	42.0982	176.6389	6059.7410
SSA	0.8125	0.4375	42.09836	176.6376	6059.7254
ISCA	0.8125	0.4375	42.09842	176.6382	6059.7457
Chaotic SSA	0.8750	0.4375	45.33679	140.2539	6090.527

Table 13 Optimal decision variables for 3-bus system obtained from various algorithms

Decision variable	RST	DE	MDE1	MDE2	MDE3	MDE4	MDE5	OCDE1	OCDE2	LX-PM	GWO	SCA	SC-GWO
TS^1	0.05006	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05016	0.0500	0.0500	0.05
TS^2	0.21073	0.219	0.1976	0.1976	0.1976	0.1976	0.1976	0.1976	0.1976	0.20446	0.1989	0.2469	0.2057
TS^3	0.05002	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05002	0.0500	0.0520	0.05
TS^4	0.21883	0.2135	0.209	0.209	0.209	0.209	0.209	0.209	0.209	0.21404	0.2155	0.2525	0.2099
TS^5	0.18814	0.1950	0.1812	0.1812	0.1812	0.1812	0.1812	0.1812	0.1812	0.19364	0.1835	0.1837	0.1813
TS^6	0.19538	0.1953	0.18067	0.18067	0.18067	0.18067	0.18067	0.18067	0.18067	0.19495	0.1841	0.2211	0.1846
PS^1	1.25123	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25071	1.2531	1.2500	1.25
PS^2	1.35344	1.25	1.4999	1.5	1.4999	1.5	1.5	1.4999	1.5	1.43549	1.4908	1.2713	1.4021
PS^3	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.2509	1.2510	1.2830	1.2506
PS^4	1.38177	1.4605	1.4999	1.5	1.4999	1.5	1.5	1.4999	1.5	1.46147	1.4163	1.5000	1.4917
PS^5	1.37434	1.25	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.29151	1.4598	1.5000	1.4993
PS^6	1.25019	1.25	1.4999	1.5	1.4999	1.5	1.5	1.4999	1.5	1.25908	1.4381	1.5000	1.4314

Table 14 Optimal decision variables for 4-bus system obtained from various algorithms

Decision variable	RST	DE	MDE1	MDE2	MDE3	MDE4	MDE5	OCDE1	OCDE2	LX-PM	GWO	SCA	SC-GWO
TS^1	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.0602	0.05
TS^2	0.2242	0.2248	0.2121	0.2123	0.2121	0.2121	0.2121	0.2122	0.2122	0.2249	0.2181	0.2554	0.2128
TS^3	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.0500	0.05
TS^4	0.1587	0.1515	0.1515	0.1515	0.1515	0.1515	0.1515	0.1516	0.1516	0.1637	0.1258	0.1993	0.1265
TS^5	0.1367	0.1264	0.1264	0.1264	0.1264	0.1262	0.1264	0.1262	0.1262	0.1318	0.1277	0.1621	0.1264
TS^6	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.0637	0.05
TS^7	0.1388	0.1337	0.1338	0.1371	0.1338	0.1337	0.1337	0.1338	0.1337	0.1371	0.1345	0.1513	0.134
TS^8	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.0500	0.05
PS^1	1.2910	1.2734	1.2733	1.2733	1.2733	1.25	1.2734	1.2734	1.25	1.2859	1.25	1.2500	1.255
PS^2	1.2645	1.25	1.4998	1.4959	1.50	1.50	1.4999	1.25	1.50	1.2502	1.3817	1.2500	1.4867
PS^3	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.2501	1.25	1.4770	1.2509
PS^4	1.3460	1.4997	1.4996	1.4997	1.4995	1.50	1.4999	1.4997	1.50	1.2501	1.4986	1.2500	1.4849
PS^5	1.2669	1.4997	1.50	1.50	1.4997	1.50	1.50	1.4997	1.50	1.3757	1.4679	1.2500	1.4969
PS^6	1.2512	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.2503	1.2563	1.2500	1.25
PS^7	1.3937	1.5	1.4997	1.4274	1.4995	1.4998	1.50	1.50	1.50	1.4304	1.4872	1.2500	1.4977
PS^8	1.2508	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.2523	1.2527	1.2500	1.2509

can be observed that the algorithm DE, modified variants of DE (MDE1, MDE2, MDE3, MDE4, MDE5, OCDE1 and OCDE2) [82, 84] and OBSCA [58] are unable to provide a feasible solution for any bus system. The algorithm RST [85], SCA [4] and m-SCA [34] offer the feasible solution for only 3- and 4-bus systems. The GWO [5], mGWO [55], ISCA [34], SSA [11], chaotic SSA [57] and WOA [56] are successful in providing the feasible solutions of all bus systems. By comparing the objective function value between the proposed SC-GWO algorithms and other algorithms, it can be observed that the SC-GWO algorithm provides a better objective function value compared to different comparative algorithms for each bus system. Therefore, the comparison analysis between state-of-the-art algorithms that are applied to solve relay coordination problems in the literature and some new optimization algorithms ensures better search efficiency of the proposed SC-GWO algorithm as compared to other comparative algorithms.

Table 15 Optimal decision variables for 6-bus systems obtained from various algorithms

Decision variable	DE	MDE1	MDE2	MDE3	MDE4	MDE5	OCDE1	OCDE2	LX-PM	GWO	SCA	SC-GWO
TS^1	0.1171	0.1171	0.1149	0.1034	0.1144	0.1144	0.1144	0.1144	0.11046	0.1082	0.1561	0.1161
TS^2	0.2082	0.1866	0.2037	0.1863	0.1684	0.1864	0.1864	0.1864	0.20821	0.1873	0.2552	0.1896
TS^3	0.0997	0.0965	0.0982	0.0961	0.0947	0.0947	0.0947	0.0947	0.08468	0.0974	0.1882	0.0981
TS^4	0.1125	0.1119	0.10367	0.1125	0.1006	0.1006	0.1006	0.1006	0.11336	0.1057	0.1375	0.1109
TS^5	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05019	0.0503	0.1022	0.0501
TS^6	0.058	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05004	0.0509	0.2004	0.0501
TS^7	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05022	0.05	0.0558	0.05
TS^8	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05028	0.0502	0.0509	0.05
TS^9	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05018	0.0504	0.0561	0.05
TS^{10}	0.0719	0.0706	0.0575	0.0703	0.0701	0.0701	0.0701	0.0701	0.06093	0.0666	0.1027	0.0663
TS^{11}	0.0649	0.0649	0.0667	0.0649	0.0649	0.0649	0.0649	0.0649	0.06679	0.0661	0.0872	0.0651
TS^{12}	0.0617	0.0617	0.0566	0.0509	0.0509	0.0509	0.0509	0.0509	0.05324	0.058	0.0652	0.0516
TS^{13}	0.05	0.05	0.0635	0.05	0.05	0.05	0.05	0.05	0.05034	0.05	0.076	0.05
TS^{14}	0.08560	0.086	0.0859	0.0857	0.0857	0.0709	0.0709	0.0709	0.08317	0.0783	0.0998	0.0729
PS^1	1.2505	1.2515	1.2635	1.4995	1.2602	1.2602	1.2602	1.2602	1.41006	1.4307	1.2595	1.2695
PS^2	1.25	1.4959	1.2993	1.4995	1.4987	1.4987	1.4987	1.4987	1.28584	1.4906	1.2946	1.4659
PS^3	1.2512	1.2525	1.2622	1.2575	1.2761	1.2761	1.2761	1.2761	1.47458	1.2561	1.2553	1.256
PS^4	1.2515	1.2632	1.4322	1.2508	1.4992	1.4992	1.4992	1.4992	1.26098	1.4356	1.3012	1.2842
PS^5	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25185	1.253	1.3883	1.2554
PS^6	1.25	1.3822	1.3885	1.381	1.3814	1.3814	1.3814	1.3814	1.40044	1.4824	1.2575	1.3925
PS^7	1.25	1.25	1.2508	1.25	1.25	1.25	1.25	1.25	1.25050	1.25	1.3634	1.25
PS^8	1.25	1.2501	1.25	1.25	1.2505	1.2505	1.2505	1.2505	1.25043	1.2517	1.3568	1.2501
PS^9	1.2502	1.25	1.2514	1.25	1.25	1.25	1.25	1.25	1.25009	1.2527	1.3016	1.2507
PS^{10}	1.2502	1.2501	1.497	1.2521	1.25	1.25	1.25	1.25	1.33302	1.2574	1.3271	1.258
PS^{11}	1.4998	1.4999	1.4759	1.4998	1.4999	1.4999	1.4999	1.4999	1.47820	1.4892	1.4072	1.5
PS^{12}	1.2575	1.2529	1.47	1.4997	1.5	1.5	1.5	1.5	1.44120	1.3438	1.3859	1.4775
PS^{13}	1.4805	1.4664	1.2728	1.4647	1.4615	1.1615	1.4615	1.1615	1.44245	1.4614	1.3264	1.4352
PS^{14}	1.2557	1.25	1.2624	1.254	1.4979	1.4979	1.4979	1.4979	1.32587	1.4038	1.4393	1.4706

12 Conclusions

In the present paper, a hybrid version of SCA and GWO called SC-GWO is proposed with the aim of maintaining an appropriate synergy between the operators' exploration and exploitation. In the proposed SC-GWO, first, the search equation of the SCA is modified to provide a better and promising direction of search. In the second step, this new search mechanism is hybridized with the GWO to establish a comparatively better balance between exploration and exploitation. For the performance evaluation, a well-known and standard benchmark set of 13 problems is adopted. The robustness of the proposed SC-GWO algorithm in terms of scalability is also examined by increasing the dimension of the issues from 30 to 100. The analysis of results through statistical test and convergence curves shows better searchability in the SC-GWO as compared to original SCA and the other optimization methods. Moreover, the comparison of the results on engineering design problems also demonstrates the superior searchability of the SC-GWO as compared to different algorithms. The performance comparison on the relay coordination problem also shows that the SC-GWO provides a better optimal setting for minimizing the operating time in 3-, 4-, 6- and 14-bus systems. In the future, we will evaluate the proposed SC-GWO algorithm on more complex benchmark problems. The binary and multi-objective version of the SC-GWO can be utilized in the future to solve more complex engineering cases such as vehicle scheduling, or aircraft streamline modeling problems. Various constraint handling techniques can also be integrated into the SC-GWO to introduce its efficient constrained version.

Table 16 Optimal decision variables for 14-bus systems obtained from various algorithms

Decision variable	GWO	SCA	SC-GWO	Decision variable	GWO	SCA	SC-GWO
TS^1	0.0661	0.0911	0.0518	PS^1	1.2523	1.2580	1.2554
TS^2	0.0500	0.1588	0.0500	PS^2	1.5000	1.4117	1.5
TS^3	0.0593	0.2481	0.0530	PS^3	1.3155	1.3020	1.3865
TS^4	0.0556	0.0520	0.0502	PS^4	1.2748	1.2584	1.2782
TS^5	0.0630	0.0782	0.0579	PS^5	1.2867	1.4149	1.2984
TS^6	0.0615	0.0564	0.0940	PS^6	1.3532	1.3653	1.2505
TS^7	0.0659	0.0563	0.0748	PS^7	1.4904	1.3965	1.2793
TS^8	0.1121	0.2568	0.0932	PS^8	1.3035	1.4203	1.2625
TS^9	0.0834	0.0618	0.0937	PS^9	1.3203	1.2944	1.4670
TS^{10}	0.1404	0.0835	0.1501	PS^{10}	1.2778	1.4064	1.2986
TS^{11}	0.1220	0.1723	0.1644	PS^{11}	1.3077	1.3655	1.2551
TS^{12}	0.2230	0.1747	0.2488	PS^{12}	1.4540	1.4499	1.3979
TS^{13}	0.1506	0.0948	0.1441	PS^{13}	1.2550	1.3499	1.2611
TS^{14}	0.1472	0.1329	0.1239	PS^{14}	1.2683	1.4212	1.2759
TS^{15}	0.1307	0.1171	0.1188	PS^{15}	1.3565	1.4941	1.3225
TS^{16}	0.2239	0.1543	0.2335	PS^{16}	1.3641	1.4139	1.3687
TS^{17}	0.1954	0.1703	0.1467	PS^{17}	1.3762	1.4460	1.2641
TS^{18}	0.2206	0.2159	0.2149	PS^{18}	1.2613	1.4158	1.3273
TS^{19}	0.1462	0.3214	0.1554	PS^{19}	1.3008	1.4245	1.2673
TS^{20}	0.1600	0.3816	0.1208	PS^{20}	1.2844	1.3766	1.4434
TS^{21}	0.2198	0.3703	0.2019	PS^{21}	1.2922	1.2513	1.4927
TS^{22}	0.4229	0.4361	0.4231	PS^{22}	1.3838	1.5000	1.3813
TS^{23}	0.0775	0.2535	0.0503	PS^{23}	1.3001	1.3236	1.3123
TS^{24}	0.3834	0.8389	0.3691	PS^{24}	1.3810	1.4107	1.4084
TS^{25}	0.0524	0.0625	0.0504	PS^{25}	1.3542	1.3402	1.3569
TS^{26}	0.2970	0.3216	0.2957	PS^{26}	1.4480	1.2770	1.4355
TS^{27}	0.3397	0.1982	0.3492	PS^{27}	1.3726	1.2848	1.2886
TS^{28}	0.0503	0.1379	0.0761	PS^{28}	1.3063	1.4281	1.3631
TS^{29}	0.1298	0.0506	0.1216	PS^{29}	1.4059	1.5000	1.4014
TS^{30}	0.2739	0.1631	0.2842	PS^{30}	1.4613	1.4894	1.3323
TS^{31}	0.2689	0.2052	0.2513	PS^{31}	1.2630	1.3884	1.3070
TS^{32}	0.4433	0.2164	0.4381	PS^{32}	1.3528	1.3124	1.3610
TS^{33}	0.2460	0.1187	0.2350	PS^{33}	1.3027	1.3881	1.3946
TS^{34}	0.2631	0.1385	0.2302	PS^{34}	1.3727	1.3942	1.3104
TS^{35}	0.3957	0.2229	0.3773	PS^{35}	1.2874	1.2648	1.3663
TS^{36}	0.3561	0.4726	0.3640	PS^{36}	1.4803	1.2869	1.2952
TS^{37}	0.2419	0.2728	0.2395	PS^{37}	1.3942	1.4367	1.2708
TS^{38}	0.4586	0.6090	0.4298	PS^{38}	1.2703	1.2728	1.3582
TS^{39}	0.2059	0.0612	0.1764	PS^{39}	1.3519	1.2712	1.3165
TS^{40}	0.3695	0.4185	0.3690	PS^{40}	1.4074	1.4711	1.3634

Table 17 Results' comparison of different bus systems with other metaheuristics

Algorithm	IEEE 3-bus system	IEEE 4-bus system	IEEE 6-bus system	IEEE 14-bus system
RST	4.8354	3.7050	–	–
DE	4.8422[a]	3.6774[a]	10.6272[a]	42.7843[a]
MDE1	4.8070[a]	3.6694[a]	10.5067[a]	–
MDE2	4.7873[a]	3.6734[a]	10.6238[a]	–
MDE3	4.7822[a]	3.6692[a]	10.4370[a]	–
MDE4	4.7806[a]	3.6674[a]	10.3812[a]	–
MDE5	4.7806[a]	3.6694[a]	10.3514[a]	–
OCDE1	4.7806[a]	3.6674[a]	10.3479[a]	37.3540[a]
OCDE2	4.7806[a]	3.6674[a]	10.3286[a]	37.4603[a]
LX-PM	4.8340	3.7029	10.4581	37.2881
SCA	5.4094	4.2348	15.8528[a]	35.4246[a]
m-SCA	5.4347	4.4143	29.5899[a]	45.3325[a]
ISCA	4.8161	3.5907	11.2142	39.0246
GWO	4.8002	3.5748	10.4537	38.3534
mGWO	4.8019	3.5728	10.4419	43.1888
wGWO	4.8015	3.5725	10.3687	42.8796
SSA	4.9489	3.8590	14.9644	46.0893
Chaotic SSA	4.9560	3.8883	15.6504	45.5356
OBSCA	2.8575[a]	3.0474[a]	11.9088[a]	53.8793[a]
WOA	4.9209	3.6358	12.3277	43.8405
SC-GWO	**4.7995**	**3.5688**	**10.3376**	**37.1387**

[a]Represents the infeasibility of the obtained solution

References

1. Goldberg DE (2006) Genetic algorithms. Pearson Education India, Bangalore
2. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks
3. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Global Optim 39:459–471
4. Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. Knowl-Based Syst 96:120–133
5. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
6. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl-Based Syst 89:228–249
7. Chen H, Jiao S, Wang M, Heidari AA, Zhao X (2020) Parameters identification of photovoltaic cells and modules using diversification-enriched Harris hawks optimization with chaotic drifts. J Clean Prod 244:118778
8. 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
9. Mafarja M, Aljarah I, Heidari AA, Hammouri AI, Faris H, Ala'M A-Z, Mirjalili S (2018) Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. Knowl-Based Syst 145:25–45
10. Gupta S, Deep K, Heidari AA, Moayedi H, Chen H (2019) Harmonized salp chain-built optimization. Eng Comput 36: https://doi.org/10.1007/s00366-019-00871-5
11. 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 Softw 114:163–191
12. Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. Adv Eng Softw 105:30–47
13. Zhang Z, Jiang D, Liu W, Chen J, Li E, Fan J, Xie K (2019) Study on the mechanism of roof collapse and leakage of horizontal cavern in thinly bedded salt rocks. Environ Earth Sci 78:292. https://doi.org/10.1007/s12665-019-8292-2
14. Qiao W, Yang Z (2019) Modified dolphin swarm algorithm based on chaotic maps for solving high-dimensional function optimization problems. IEEE Access 7:110472–110486. https://doi.org/10.1109/ACCESS.2019.2931910
15. Qiao W, Yang Z (2019) Solving large-scale function optimization problem by using a new metaheuristic algorithm based on quantum dolphin swarm algorithm. IEEE Access 7:138972–138989. https://doi.org/10.1109/ACCESS.2019.2942169
16. Qiao W, Tian W, Tian Y, Yang Q, Wang Y, Zhang J (2019) The forecasting of PM2.5 using a hybrid model based on wavelet transform and an improved deep learning algorithm. IEEE Access 7:142814–142825. https://doi.org/10.1109/ACCESS.2019.2944755
17. Liu W, Zhang X, Fan J, Li Y, Wang L (2020) Evaluation of Potential for Salt Cavern Gas Storage and Integration of Brine Extraction: Cavern Utilization, Yangtze River Delta Region. Nat Resour Res 29. https://doi.org/10.1007/s11053-020-09640-4
18. Qiao W, Huang K, Azimi M, Han S (2019) A novel hybrid prediction model for hourly gas consumption in supply side based on improved whale optimization algorithm and relevance vector machine. IEEE Access 7:88218–88230. https://doi.org/10.1109/ACCESS.2019.2918156
19. Fan J, Jiang D, Liu W, Wu F, Chen J, Daemen J (2019) Discontinuous fatigue of salt rock with low-stress intervals. Int J Rock Mech Min Sci 115:77–86. https://doi.org/10.1016/j.ijrmms.2019.01.013
20. Liu W, Zhang Z, Chen J, Fan J, Jiang D, Jjk D, Li Y (2019) Physical simulation of construction and control of two butted-well horizontal cavern energy storage using large molded rock

salt specimens. Energy 185:682–694. https://doi.org/10.1016/j.energy.2019.07.014
21. Chen J, Lu D, Liu W, Fan J, Jiang D, Yi L, Kang Y (2020) Stability study and optimization design of small-spacing two-well (SSTW) salt caverns for natural gas storages. J Energy Storage 27:101131. https://doi.org/10.1016/j.est.2019.101131
22. Zhou G, Moayedi H, Foong LK (2020) Teaching-learning-based metaheuristic scheme for modifying neural computing in appraising energy performance of building. Eng Comput. https://doi.org/10.1007/s00366-020-00981-5
23. Zhou G, Moayedi H, Bahiraei M, Lyu Z (2020) Employing artificial bee colony and particle swarm techniques for optimizing a neural network in prediction of heating and cooling loads of residential buildings. J Clean Prod. https://doi.org/10.1016/j.jclepro.2020.120082
24. Liu W, Zhang ZX, Fan JY, Jiang DY, Daemen JJK (2020) Research on the stability and treatments of natural gas storage caverns with different shapes in bedded salt rocks. IEEE Access 8:000507. https://doi.org/10.1109/ACCESS.2020.2967078
25. Jinlong L, Wenjie X, Jianjing Z, Wei L, Xilin S, Chunhe Y (2020) Modeling the mining of energy storage salt caverns using a structural dynamic mesh. Energy 193:116730. https://doi.org/10.1016/j.energy.2019.116730
26. Gao W, Wu H, Siddiqui MK, Baig AQ (2018) Study of biological networks using graph theory. Saudi J Biol Sci 25:1212–1219
27. Gao W, Wang W, Dimitrov D, Wang Y (2018) Nano properties analysis via fourth multiplicative ABC indicator calculating. Arab J Chem 11:793–801
28. Gao W, Guirao JLG, Basavanagoud B, Wu J (2018) Partial multi-dividing ontology learning algorithm. Inf Sci 467:35–58
29. Gao W, Guirao JLG, Abdel-Aty M, Xi W (2019) An independent set degree condition for fractional critical deleted graphs. Discrete Cont Dyn Syst-S 12:877–886
30. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1:67–82
31. Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. IEEE Trans Evol Comput 3:82–102
32. Rizk-Allah RM (2019) An improved sine–cosine algorithm based on orthogonal parallel information for global optimization. Soft Comput 23:7135–7161
33. Elaziz MA, Oliva D, Xiong S (2017) An improved opposition-based sine cosine algorithm for global optimization. Expert Syst Appl 90:484–500
34. Gupta S, Deep K (2019) Improved sine cosine algorithm with crossover scheme for global optimization. Knowl-Based Syst 165:374–406
35. Nayak DR, Dash R, Majhi B, Wang S (2018) Combining extreme learning machine with modified sine cosine algorithm for detection of pathological brain. Comput Electr Eng 68:366–380
36. Zhang J, Zhou Y, Luo Q (2018) An improved sine cosine water wave optimization algorithm for global optimization. J Intell Fuzzy Syst 34:2129–2141
37. Nenavath H, Jatoth RK (2018) Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking. Appl Soft Comput 62:1019–1043
38. Zamli KZ, Din F, Ahmed BS, Bures M (2018) A hybrid Q-learning sine-cosine-based strategy for addressing the combinatorial test suite minimization problem. PLoS ONE 13:e0195675
39. Chen H, Heidari AA, Zhao X, Zhang L, Chen H (2020) Advanced orthogonal learning-driven multi-swarm sine cosine optimization: framework and case studies. Expert Syst Appl 144:113113
40. Gupta S, Deep K (2019) A novel hybrid sine cosine algorithm for global optimization and its application to train multilayer perceptrons. Appl Intell 50: https://doi.org/10.1007/s10489-019-01570-w
41. Muangkote N, Sunat K, Chiewchanwattana S (2014) An improved grey wolf optimizer for training q-Gaussian Radial Basis Functional-link nets. In: 2014 international computer science and engineering conference (ICSEC)
42. Jayabarathi T, Raghunathan T, Adarsh B, Suganthan PN (2016) Economic dispatch using hybrid grey wolf optimizer. Energy 111:630–641
43. Rodríguez L, Castillo O, Soria J, Melin P, Valdez F, Gonzalez CI, Martinez GE, Soto J (2017) A fuzzy hierarchical operator in the grey wolf optimizer algorithm. Appl Soft Comput 57:315–328
44. Castillo O, Amador-Angulo L (2018) A generalized type-2 fuzzy logic approach for dynamic parameter adaptation in bee colony optimization applied to fuzzy controller design. Inf Sci 460:476–496
45. Castillo O, Melin P, Ontiveros E, Peraza C, Ochoa P, Valdez F, Soria J (2019) A high-speed interval type 2 fuzzy system approach for dynamic parameter adaptation in metaheuristics. Eng Appl Artif Intell 85:666–680
46. Ochoa P, Castillo O, Soria J (2020) Optimization of fuzzy controller design using a Differential Evolution algorithm with dynamic parameter adaptation based on Type-1 and Interval Type-2 fuzzy systems. Soft Comput 24:193–214
47. Olivas F, Valdez F, Melin P, Sombra A, Castillo O (2019) Interval type-2 fuzzy logic for dynamic parameter adaptation in a modified gravitational search algorithm. Inf Sci 476:159–175
48. Sánchez D, Melin P, Castillo O (2017) A grey wolf optimizer for modular granular neural networks for human recognition. Comput Intell Neurosci 2017:1–24. https://doi.org/10.1155/2017/4180510
49. Tawhid MA, Ali AF (2017) A hybrid grey wolf optimizer and genetic algorithm for minimizing potential energy function. Memetic Computing 9:347–359
50. Gupta S, Deep K (2019) Enhanced leadership-inspired grey wolf optimizer for global optimization problems. Eng Comput 36:1–24. https://doi.org/10.1007/s00366-019-00795-0
51. Gupta S, Deep K (2018) Cauchy Grey Wolf Optimiser for continuous optimisation problems. J Exp Theor Artif Intell 30:1051–1075
52. Long W, Jiao J, Liang X, Tang M (2018) An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization. Eng Appl Artif Intell 68:63–80
53. Gupta S, Deep K (2019) An efficient grey wolf optimizer with opposition-based learning and chaotic local search for integer and mixed-integer optimization problems. Arab J Sci Eng 44:7277–7296
54. Singh N, Singh S (2017) A novel hybrid GWO-SCA approach for optimization problems. Engineering Science and Technology, an International Journal 20:1586–1601
55. Mittal N, Singh U, Sohi BS (2016) Modified grey wolf optimizer for global engineering optimization. Appl Comput Intell Soft Comput 2016:1–1. https://doi.org/10.1155/2016/7950348
56. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
57. Sayed GI, Khoriba G, Haggag MH (2018) A novel chaotic salp swarm algorithm for global optimization and feature selection. Appl Intell 48:3462–3481
58. Deb K (2000) An efficient constraint handling method for genetic algorithms. Comput Methods Appl Mech Eng 186:311–338
59. Sandgren E (1988) Nonlinear integer and discrete programming in mechanical design. In: Proceeding of the ASME design technology conference
60. Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In: 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360)
61. Gandomi AH, Yang X-S, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29:17–35

62. Wu S-J, Chow P-T (1995) Genetic algorithms for nonlinear mixed discrete-integer optimization problems via meta-genetic parameter optimization. Eng Optim+ A35 24:137–159
63. Kannan B, Kramer SN (1994) An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. J Mech Des 116(2):405–411. https://doi.org/10.1115/1.2919393
64. Nowacki H (1973) Optimization in pre-contract ship design, Computer Applications in the Automation of Shipyard Operation and Ship Design,IFIP/IFAC/JSNA, Tokyo, Japan
65. Ray T, Saini P (2001) Engineering design optimization using a swarm with an intelligent information sharing among individuals. Eng Optim 33:735–748
66. Tsai J-F (2005) Global optimization of nonlinear fractional programming problems in engineering design. Eng Optim 37:399–409
67. Akhtar S, Tai K, Ray T (2002) A socio-behavioural simulation model for engineering design optimization. Eng Optim 34:341–354
68. Ku KuJ, Rao SS, Chen L (1998) Taguchi-aided search method for design optimization of engineering systems. Eng Optim 30:1–23
69. Mezura-Montes E, Coello CAC (2008) An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. Int J Gen Syst 37:443–473
70. Arora J (2004) Optimum design concepts: optimality conditions. Introduction to optimum design. Elsevier, Amsterdam
71. Belegundu AD, Arora JS (1985) A study of mathematical programming methods for structural optimization. Part I: theory. Int J Numer Meth Eng 21:1583–1599
72. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179:2232–2248
73. Coello CAC (2000) Use of a self-adaptive penalty approach for engineering optimization problems. Comput Ind 41:113–127
74. He Q, Wang L (2007) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng Appl Artif Intell 20:89–99
75. Kaveh A, Khayatazad M (2012) A new meta-heuristic method: ray optimization. Comput Struct 112:283–294
76. Deb K (1997) GeneAS: A Robust Optimal Design Technique for Mechanical Component Design. In: Dasgupta D, Michalewicz Z (eds) Evolutionary Algorithms in Engineering Applications. Springer, Berlin, Heidelberg, pp 497–514. https://doi.org/10.1007/978-3-662-03423-1_27
77. Huang F-z, Wang L, He Q (2007) An effective co-evolutionary differential evolution for constrained optimization. Appl Math Comput 186:340–356
78. Kaveh A, Talatahari S (2010) An improved ant colony optimization for constrained engineering design problems. Eng Comput 27:155–182. https://doi.org/10.1108/02644401011008577
79. Sandgren E (1990) Nonlinear integer and discrete programming in mechanical design optimization
80. Blackburn J, Domin T (2006) Symmetrical components: a review. Protective relaying: principles and applications, CRC Press is an imprint of Taylor & Francis Group, an Informa business, ISBN 10:1-57444-716-5
81. Storn R, Price K (1997) Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11:341–359
82. Chelliah TR, Thangaraj R, Allamsetty S, Pant M (2014) Coordination of directional overcurrent relays using opposition based chaotic differential evolution algorithm. Int J Electr Power Energy Syst 55:341–350
83. Thangaraj R, Pant M, Abraham A (2010) New mutation schemes for differential evolution algorithm and their application to the optimization of directional over-current relay settings. Appl Math Comput 216:532–544
84. Thangaraj R, Pant M, Deep K (2010) Optimal coordination of over-current relays using modified differential evolution algorithms. Eng Appl Artif Intell 23:820–829
85. Birla D, Maheshwari RP, Gupta HO, Deep K, Thakur M (2006) Application of random search technique in directional overcurrent relay coordination. Int J Emerg Electric Power Syst 7:1. https://doi.org/10.2202/1553-779X.1271
86. Thakur M (2007) New real coded genetic algorithms for global optimization. Ph.D thesis, Department of Mathematics, Indian Institute of Technology
87. Qiao W, Yang Z (2020) An improved dolphin swarm algorithm based on Kernel Fuzzy C-means in the application of solving the optimal problems of largescalefunction. IEEE Access. https://doi.org/10.1109/ACCESS.2019.2958456
88. Benmouyal G, Meisinger M, Burnworth J, Elmore W, Freirich K, Kotos P, Leblanc P, Lerley P, McConnell J, Mizener J (1999) IEEE standard inverse-time characteristic equations for overcurrent relays. IEEE Trans Power Delivery 14:868–872