Journal of Soft Computing in Civil Engineering 6-4 (2022) 1-42

Contents lists available at SCCE

Journal of Soft Computing in Civil Engineering

Journal homepage: www.jsoftcivil.com

A New Enhanced Hybrid Grey Wolf Optimizer (GWO) Combined with Elephant Herding Optimization (EHO) Algorithm for Engineering Optimization

Zeynab Hoseini[1], Hesam Varaee[1*], Mahdi Rafieizonooz[2], Jang-Ho Jay Kim[2]

1. Department of Engineering, Ale Taha Institute of Higher Education, 14888-36164, Tehran, Iran
2. School of Civil and Environmental Engineering, Yonsei University, Yonsei-ro 50, Seodaemun-gu, Seoul 03722, South Korea

Corresponding author: *varaee.hesam@aletaha.ac.ir*

https://doi.org/10.22115/SCCE.2022.342360.1436

ARTICLE INFO

Article history:
Received: 14 May 2022
Revised: 18 July 2022
Accepted: 31 August 2022

Keywords:
Optimization;
Grey wolf optimizer (GWO);
Elephant herding optimization (EHO);
Convergence speed;
Constrained engineering problems.

ABSTRACT

Although the exploitation of GWO advances sharply, it has limitations for continuous implementing exploration. On the other hand, the EHO algorithm easily has shown its capability to prevent local optima. For hybridization and by considering the advantages of GWO and the abilities of EHO, it would be impressive to combine these two algorithms. In this respect, the exploitation and exploration performances and the convergence speed of the GWO algorithm are improved by combining it with the EHO algorithm. Therefore, this paper proposes a new hybrid Grey Wolf Optimizer (GWO) combined with Elephant Herding Optimization (EHO) algorithm. Twenty-three benchmark mathematical optimization challenges and six constrained engineering challenges are used to validate the performance of the suggested GWOEHO compared to both the original GWO and EHO algorithms and some other well-known optimization algorithms. Wilcoxon's rank-sum test outcomes revealed that GWOEHO outperforms others in most function minimization. The results also proved that the convergence speed of GWOEHO is faster than the original algorithms.

How to cite this article: Hoseini Z, Varaee, H, Rafieizonooz, M, Jay Kim JH. A new enhanced hybrid grey wolf optimizer (GWO) combined with elephant herding optimization (EHO) algorithm for engineering optimization. J Soft Comput Civ Eng 2022;6(4): 1–42. https://doi.org/10.22115/scce.2022.342360.1436

1. Introduction

The goal of optimization is to seek the best acceptable solution, given the constraints and limitations of the problem. Each optimization problem has several independent variables called design variables, represented by the n-dimensional vector $\boldsymbol{X}$. There may be different solutions for a problem, and a function called the objective function is defined to compare these solutions and choose the best vector $\boldsymbol{X}$ as an optimal solution [1–5]. In general, various optimization techniques may be categorized into the two main local and global optimization methods. Among them, metaheuristics as a global optimization algorithms have a much better chance than the local algorithms to search out the global or near-global optimum [6–11]. Since the last twenty years, metaheuristic algorithms have become extremely popular thanks to their efficient and robust performance in addressing high-dimensional nonlinear optimization problems [12–14]. Genetic algorithm (GA) [15], flying squirrel optimizer (FSO) [16], cuckoo search (CS) [17], differential evolution (DE) [18], artificial bee colony (ABC) [19], grey wolf optimizer (GWO) [20], bat algorithm (BA) [21], elephant herding optimization (EHO) [22], moth search algorithm [23], ideal gas molecular movement (IGMM) algorithm [24] and particle swarm optimization (PSO) [25] are some of the metaheuristic algorithms. These algorithms are also classified into evolutionary algorithms, swarm-based algorithms, and trajectory-based algorithms. For instance, GA, DE, and HS are classified as evolutionary algorithms [26]. PSO is classed as a swarm-based algorithm and, ant colony optimization (ACO) is classed as a trajectory-based method [27,28].

Recently, a new metaheuristic algorithm called GWO, motivated by the hierarchy of leadership and also the hunting mechanism of gray wolves, has been developed by Mirjalili et al. [20]. The results of their study have shown that the GWO can deliver very competitive outcomes compared to the known algorithms. Nevertheless, the biggest issue in GWO is the liable to inactivity in local optima [29]. Therefore, the main objective of current study is to boost the performance of GWO when a new hybridizing approach is presented.

2. Theoretical background

Nowadays the utilization of GWO for various applications has been grown rapidly [30]. Medjahed et al. [31] employed the original GWO algorithm within the band selection problem to decrease the dimensionality of hyperspectral images. In [32], Emary et al. while minimizing the chosen specifications, recommended a new binary version of the GWO algorithm for finding a specific subset maximizing the categorization precision.

Besides of the better performance of GWO on real-world problems than many other papulation-based algorithms, it also encounters some challenging problems as well. For instance, the original GWO algorithm may be simply locked in within the local optima when searching multi-modal functions, and also the convergence level will reduce considerably in the further iterations [33]. Hence, several alternatives of GWO are developed to solve the above-mentioned aspects.

In order to set the bidding strategy for a producing company in a consistent price spot market, [31] and [34] developed a new modified version of the GWO algorithm. In 2018, Long et al. [35]

utilized EEGWO as an improved version of GWO for overcoming some engineering optimization challenges. The suggested EEGWO developed the improved position-updating equation to modify the exploration performance of the original GWO algorithm. In addition, Gupta and Deep [36] suggested a random walk strategy for a modified version of GWO to increase the global search performance of the original GWO algorithm. Furthermore, Mittal et al. [37] introduced an improved GWO (mGWO) employing an exponential function to decay parameters over iterations to balance exploitation and exploration.

In the field of hybrid metaheuristics, GWO has also achieved much consideration. For example, for overcoming the feature selection and global optimization challenges, a GWO (CGWO) algorithm and a hybrid harmony search with an opposition learning strategy is presented in [38]. In 2017, Sanjay et al. [39] optimized distributed generator units' configuration using an original hybrid GWO approach based on mutation and crossover operators. Regarding to discover the best feature subset, [40] suggested a binary version of hybrid PSOGWO. In [41] and [42], a hybridized version of GWO with DE is presented for nonstop optimization and test scheduling. In order to minimize the potential energy functions, Tawhid and Ali [43] have hybridized GWO with GA. In addition, Gaidhane and Nigam [44] used an artificial bee colony (ABC) and a hybridized GWO to enhance the development system's functioning. Besides, [43] proposed a further hybrid method named GWOSCA, using the sine and cosine algorithm (SCA) and the GWO algorithm. For combining the algorithms' strengths to produce promising alternative solutions for achieving the efficient global optima, [45] proposed a hybrid GWO with CSA that is named GWOCSA. The above-mentioned researches have presented that in comparison with other global or local search methods, the hybrid methods achieved much better.

On the other hand, Wang et al. [22] introduced an EHO algorithm motivated by the herding behavior of the elephant group. Although elephants are social animals, they have complex social behaviors. A group of elephants is comprised of several clans under the leadership of a matriarch. In recent years, many variants of EHO have been presented for continuous, combinatorial, constrained, and multiobjective optimization. A comprehensive review of the EHO-based algorithms and their applications is submitted in [46]. Tuba et al. [47] suggested a novel chaotic-based EHO algorithm called CEHO to overcome unconstrained worldwide optimization challenges. Another study, ElShaarawy et al. [48], introduced an enhanced EHO algorithm to solve the fast convergence of EHO. Separating operators with balanced control was utilized to develop the exploitation and exploration performance of the proposed algorithm.

Li et al. [49] suggested a hybrid algorithm (EHGWO) that unites the advantages of EHO and GWO. EHGWO uses a newly devised fitness function to select the optimal virtual machines (VMs). Exploring the search space and exploiting the optimal solutions found are two inconsistent instructions to be considered when modeling or utilizing a metaheuristic [50]. Reasonably balancing exploitation and exploration will improve the search algorithm's performance. One alternative is to employ a hybrid method where two or more algorithms are united to enhance each algorithm's ability, and the final hybrid method can be called as a mimetic method [51].

In this study, with integrating the features of GWO and EHO algorithms and using a new separating operator, we recommended a hybrid algorithm named GWOEHO. Although the exploitation of GWO advances sharply, it has limitations for continuous implementing exploration. Therefore, in some cases, GWO cannot successfully deal with the problem always and fails to discover the optimal global solution [45]. The exploration and exploitation performances of the GWO algorithm are enhanced by embedding the EHO futures and using a novel separating operator. The convergence speed of GWO is also increased by combining it with the EHO algorithm. Utilizing a new separating operator is valuable to help the population to jump out of the local optima. The performance of the improved algorithm is then evaluated by twenty-three mathematical benchmark functions and six constrained engineering problems. The statistical test outcomes present the superiority of the suggested hybrid GWOEHO algorithm over the other well-known optimization algorithms.

3. Grey wolf optimizer (GWO)

GWO algorithm that first was suggested by Mirjalili et al. [20] mimics the leadership and hunting characteristics of the grey wolves that live in a group of 5-12 individuals. Each group of wolves is divided into alpha, beta, delta, and omega subgroups to simulate the leadership hierarchy characteristics.

Alpha wolves are the pack leaders and make decisions about wake time, hunting and sleep place. Beta wolves are alpha's assistants in making decisions, and their primary responsibility is reaction suggestions. Delta wolves known as caretakers, hunters, elders, sentinels, and scouts control omega wolves by following alpha and beta. The lowest ranking grey wolf is an omega that always must follow wolves of other levels. In the hunting process, alpha, beta, and delta are pack leaders, respectively, and omega wolves must conform. Track and approach hunting, siege, and harassment of the hunt until the movement stops, and finally, attack on prey are three main stages for wolves hunting. The GWO algorithm has been designed based on the wolves' hunting method and their social hierarchy. The mathematical model of the GWO algorithm is described as follows.

Mirjalili suggests two equations to mathematically model the encircling behavior of grey wolves [20]:

$$| (t) - \vec{X}(t)\ \vec{X}_P .\ \vec{C}\ | = \vec{D} \quad (1)$$

$$\vec{D}\ (t) - \vec{A} .\ \vec{X}_P\ (t+1) = \vec{X} \quad (2)$$

where $\vec{A}$ and $\vec{C}$ are coefficient vectors while t shows the present iteration. $\vec{X}_P$ vector is related to the position of the prey and $\vec{X}$ vector indicates the grey wolf position. The vectors $\vec{A}$ and $\vec{C}$ can be calculated following equations 3 and 4:

$$\vec{A} = 2\vec{a}.\vec{r}_1 - \vec{a} \quad (3)$$

$$\vec{C}=2.\vec{r}_2 \quad (4)$$

where elements of $\vec{a}$ are linearly reduced from 2 to 0 over the course of iterations and $\vec{r}_1, \vec{r}_2$ are random vectors in [0, 1].

Grey wolves can recognize the location of prey, encircle it and finally hunt it. The pack leader, alpha, usually guides the hunting process; however, beta and delta wolves sometimes take part in this process. Consequently, we presume the alpha, beta, and delta as the first three best candidate solutions to mathematically reproduce the hunting behavior of grey wolves. And this is due to their better knowledge about prey's potential location than other kinds of wolves known as omega. In present work, the first three best solutions were saved for mathematic modeling and other agents were forced to update their positions by those three solutions. The following formulas are suggested in this regard:

$$| -\vec{X}\ \vec{X}_\delta\ .\vec{C}_3\ |\ = \vec{D}_\delta\ ,\ | -\vec{X}\ \vec{X}_\mathcal{B}\ .\ \vec{C}_2\ |\ = \vec{D}_\mathcal{B}\ ,\ \ | -\vec{X}\ \vec{X}_\propto\ .\ \ \vec{C}_1\ | = \ \vec{D}_\propto \quad (5)$$

$$-\vec{A}_3.\vec{D}_\delta\ \ \vec{X}_\delta\ \ = \vec{X}_3\ ,\ -\vec{A}_2.\vec{D}_\mathcal{B}\ \ \vec{X}_\mathcal{B}\ \ = \vec{X}_2\ ,\ -\vec{A}_1.\vec{D}_\propto\ \ \vec{X}_\propto\ \ = \vec{X}_1 \quad (6)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_1 + \ \vec{X}_1}{3} \quad (7)$$

where the best solution is believed as the alpha (α); therefore, the second and third- fittest solutions are called beta (β) and delta (δ), respectively. The rest of the alternative solutions are recognized to be omega (ω).

As mentioned before, when the prey stops moving, the grey wolves attack. To mathematically model approaching the prey, we can reduce the value of $\vec{a}$. As $\vec{A}$ is dependent on $\vec{a}$, when random values of $\vec{A}$ are in [-1,1], the next location of a explore agent can be in any location between its present location and the location of the prey.

In the exploration process, alpha, beta, and delta wolves separate from each other to explore for prey, and after finding a suitable prey, they converge to attack it. For modeling the separation mathematically, $\vec{A}$ is used with random values bigger than 1 or less than -1 to force the explore agents to diverge from the prey. In addition, grey wolves have to separate from the prey so that they can find a fitter prey as a result of values $|A| > 1$.

$\vec{C}$ vector is another component of the GWO algorithm that contains random values in [0,2] and consequently provides random weight for prey. As a result, the effect of prey in identifying the distance is stochastically emphasized or deemphasized when the value of $\vec{C}$ vector is respectively greater than 1 and less than -1. The pseudocode of the GWO is indicated as Algorithm 1:

Algorithm 1

Begin
Initialization
  Generate the grey wolf population X_i (i=1,2,…,n)
  Set initial parameters $\boldsymbol{a}$, $\boldsymbol{A}$, and $\boldsymbol{C}$
Evaluation
  Compute the fitness of each wolf
  $X_\propto$= the fittest wolf
  X_β= the second-fittest wolf
  X_δ= the third-fittest wolf
  while ($t <$ Max number of iterations)
    for each wolf
      Update the location of the present wolf by Eq. 7
    end for
  Update $\boldsymbol{a}$, $\boldsymbol{A}$, and $\boldsymbol{C}$
  Compute the fitness of all wolves
  Update $X_\propto$. X_β , and X_δ
  $t = t + 1$
  end while
  return $X_\propto$
End

4. Elephant herding optimization

The elephant herding optimization (EHO) algorithm was first introduced by Wang et al. [22]. In the wild, elephants are social animals. The oldest matriarchal elephant leads a group of elephants, and females prefer to live in their families. In contrast, males favor leaving their tribe when they grow up and, if necessary, have the extraordinary ability to communicate through low-frequency vibrations with their tribe members [52] using the herding behavior of elephants, global optimization problems solved by ideal laws we describe below.

- Elephants live in a tribe where each tribe has a fixed number of elephants.
- A certain number of male elephants have left their tribe at the beginning of each generation and live alone.
- In each tribe, the group is led to a matriarchy, which is seen as the eldest and best elephant for the optimization problem in the tribe.

4.1. Clan updating operator

Since all tribes are influenced by one matriarch, each elephant in clan ci, its next location is controlled by matriarch ci. For the elephant j in clan ci, it can be revised by:

$$x_{new,ci,j} = x_{ci,j} + \alpha \times (x_{best,ci} - x_{ci,j}) \times r \quad (8)$$

where $x_{new,ci,j}$ and $x_{ci,j}$ are recently revised and old location for elephant j in clan ci, respectively. $\alpha \in [0,1]$ is a scale factor that specifies the effect of matriarch ci on $x_{ci,j}$. $x_{best,ci}$

demonstrates matriarch ci, which is the best elephant individual in clan ci. $r \in [0, 1]$. Uniform distribution is utilized in this study.

$$x_{new,ci,j} = \beta \times x_{center,ci} \tag{9}$$

where $\beta \in [0,1]$ is a factor that controls the effect of the $x_{center,ci}$ on $x_{new,ci,j}$. It can be seen that, the new individual $x_{new,ci,j}$ in Eq. (9) is produced by the data collected by all the elephant individuals in clan ci. $x_{center,ci}$ is the center of clan ci, and for the dth dimension it can be evaluated as:

$$x_{center,ci,d} = \frac{1}{n_{ci}} \times \sum_{j=1}^{n_{ci}} x_{ci,j,d} \tag{10}$$

where $1 \leq d \leq D$ specifies the dth dimension, and D is its total dimension. n_{ci} shows the number of elephants in clan ci and $x_{ci,j,d}$, is the dth of the elephant individual $x_{ci,j}$. The center of clan ci, $x_{center,ci}$ can be evaluated through D evaluations according to Eq. (10).

According to the above explanation, the clan updating operator can be formulated like Algorithm 2.

Algorithm 2 Clan updating operator

```
for ci=1 to nClan
    for j=1 to nci
        Update xci,j and produce xnew,ci,j by Eq. (8)
        if xci,j = xbest,ci then
            Update xci,j and produce xnew,ci,j by Eq. (9)
        end if
    end for j
end for ci
```

4.2. Separating operator

In elephant tribes, adult male elephants have left their family group and are living alone. This separation procedure can be simulated on the separator operator when solving optimization challenges so that to improve further the study for the EHO approach, the male elephants, in any case, implement the separator operator in each generation as presented in Eq. (11):

$$x_{worst,ci} = x_{min} + (x_{max} - x_{min} + 1) \times rand \tag{11}$$

where x_{min} and x_{max} are lower and upper border of the location of elephant individual, respectively. $x_{worst,ci}$ is the worst elephant individual in clan ci. $rand \in [0,1]$ is a random number uniformly distributed in the range [0, 1] [52]. Accordingly, algorithm 3 can model the separating operator.

The schematic description of the EHO algorithm is shown in Algorithm 4.

Algorithm 3 Separating operator

for c_i=1 to nClan
    Substitute the worst elephant in clan c_i by Eq. (11)
end for c_i

Algorithm 4 EHO algorithm

Begin
Initialization
    Generate the elephant population
    Set nKEL, MaxGen, $\propto$, β, nClan, and n_{ci}
    while t< MaxGen
        Sort all elephants based on their fitness.
        Save nKEL the elephant individuals.
        Implement clan updating based on Algorithm 2.
        Implement separating operator based on Algorithm 3.
        Evaluate the population according to the newly updated positions.
        Replace the worst elephant individuals with the nKEL saved ones.
        t=t+1.
    end while
    Report the best solution.
End

5. Proposed GWOEHO

This part presents a new hybrid algorithm by merging the features of GWO and EHO algorith. As aforesaid, the characteristic of the GWO algorithm is the use of the hierarchical structure of wolves and their behavior in hunting. Notwithstanding, the clan life of elephants and the subordination of the members of each clan to the leader, further the separation of male elephants from the group of elephants, have been inspiring point in the introduction of optimization algorithms. In the hybrid GWOEHO algorithm, the clan life of elephants is used to group wolves so that the wolf population is divided into a certain number of clans. The process of updating the condition of wolves is created using the proposed relationships in the GWO algorithm and after determining the alpha, beta, and gamma wolves for each clan. While updating the condition of alpha wolves in each clan, the text is done with alpha, beta, and gamma wolves among all wolves.

Next, the separation process is described by a new separation operator. One or two clans are randomly selected from among the wolf clans in the proposed new operator and the ,following formula changes the worst wolves from these clans.

$$x_{worst,ci} = t \times (x_{max}) \times (-1)^{Cindex} \tag{12}$$

The variable t in the above equation is reduced linearly using the Eq. (13) and proportional to $\mathrm{Max}_{\mathrm{iter}}$ from the value of one to zero.

$$t = 1 - Iter \times \left(\frac{2}{\mathrm{Max}_{\mathrm{iter}}}\right) \tag{13}$$

$$Iter \in [0, \mathrm{Max}_{\mathrm{iter}}/2]$$

In the above relation, the current counter iter Max_{iter} is equivalent to the maximum repetition cycle of the algorithm and x_{max} is the upper bound of the wolves and will be Cindex = 1 or cindex = 2. The pseudocode of the separation process is shown in Algorithm 5.

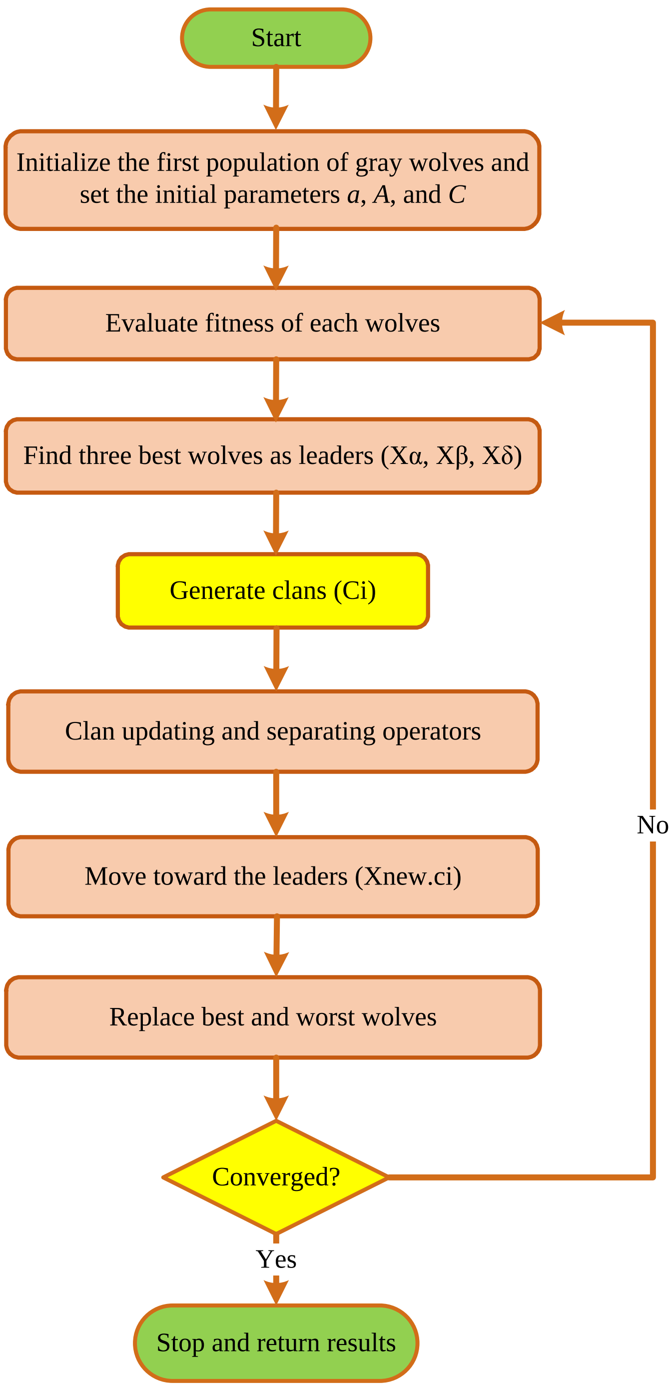


Fig. 1. flowchart of proposed GWO-EHO algorithm.

Algorithm 5

for C_i=1 to nClan (1 or 2 clans in elephant population)
    Substitute the worst elephant in clan C_i by Eq. (12)
end for ci

According to the described contents, the update of tribal operators of the GWOEHO algorithm is according to the pseudocode provided below as Algorithm 6.

Algorithm 6

```
Begin
Initialization
for c_i=1 to nClan (for all clans in wolves population)
    for j =1 to n_ci (for all wolves in clan c_i)
        Update x_ci,j and produce x_new,ci,j in accordance with GWOEHO by Eqs. (5), (6), (7)
        if x_ci,j=x_best,ci then
            Update x_ci,j and produce x_new,ci,j by Eq. (8)
        end if
    end for j
end for c_i
End
```

The flowchart of the proposed GWOEHO algorithm is displayed in Fig. 1.

6. Results and discussion

The GWOEHO algorithm is benchmarked on 23 benchmark functions in this part of the research. Many researchers [7,14,53,54] employed the first 23 benchmark functions for evaluating their proposed algorithms and methods. In spite of the simplicity, these test functions were selected to compare our outcomes to those of the present well-known meta-heuristics. Tables 1-3 indicated these benchmark functions where n demonstrates the dimension of the function, Range is the limitation of the function's explore area, and f_{min} is the target value. Additionally, these functions are the rotated, shifted, combined, and expanded deviations of the classical functions, which deliver the highest complexity between the present benchmark functions [55].

The employed benchmark functions are minimization functions and can be separated into three categories named unimodal, multi-modal, and fixed-dimension multi-modal. The GWOEHO algorithm was run 20 times on each benchmark function. The GWOEHO algorithm is compared to GA, DE, PSO, EHO, and GWO algorithms for verifying the results. The following parameters are also used in the algorithms for benchmark functions and engineering problems proposed in Table 4.

Table 1
Unimodal benchmark functions.

Test Function	n	S	f_{min}
$F_1(X) = \sum_{i=1}^{n} x_i^2$	30	$[-100.100]^n$	0
$F_2(X) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	$[-10.10]^n$	0
$F_3(X) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$	30	$[-100.100]^n$	0
$F_4(X) = \max_i \{\lvert x_i \rvert . 1 \leq i \leq n\}$	30	$[-100.100]^n$	0
$F_5(X) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i{}^2)^2 + (x_i - 1)^2]$	30	$[-30.30]^n$	0
$F_6(X) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	$[-100.100]^n$	0
$F_7(X) = \sum_{i=1}^{n} i x_i{}^4 + random[0.1)$	30	$[-1.28.1.28]^n$	0

Table 2
Multi-modal benchmark functions.

Test Function	n	S	f_{min}
$F_8(X) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	30	$[-500.500]^n$	-418.983×30
$F_9(X) = \sum_{i=1}^{n} [x_i{}^2 - 10\cos(2\pi x_i) + 10]$	30	$[-5.12.5.12]^n$	0
$F_{10}(X) = -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i{}^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)\right) + 20 + e$	30	$[-32.32]^n$	0
$F_{11}(X) = \frac{1}{4000}\sum_{i=1}^{n} x_i{}^2 - \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600.600]^n$	0
$F_{12}(X) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i.10.100.4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i.a.k.m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	$[-50.50]^n$	0
$F_{13}(X) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n} (x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\right\}$	30	$[-50.50]^n$	0

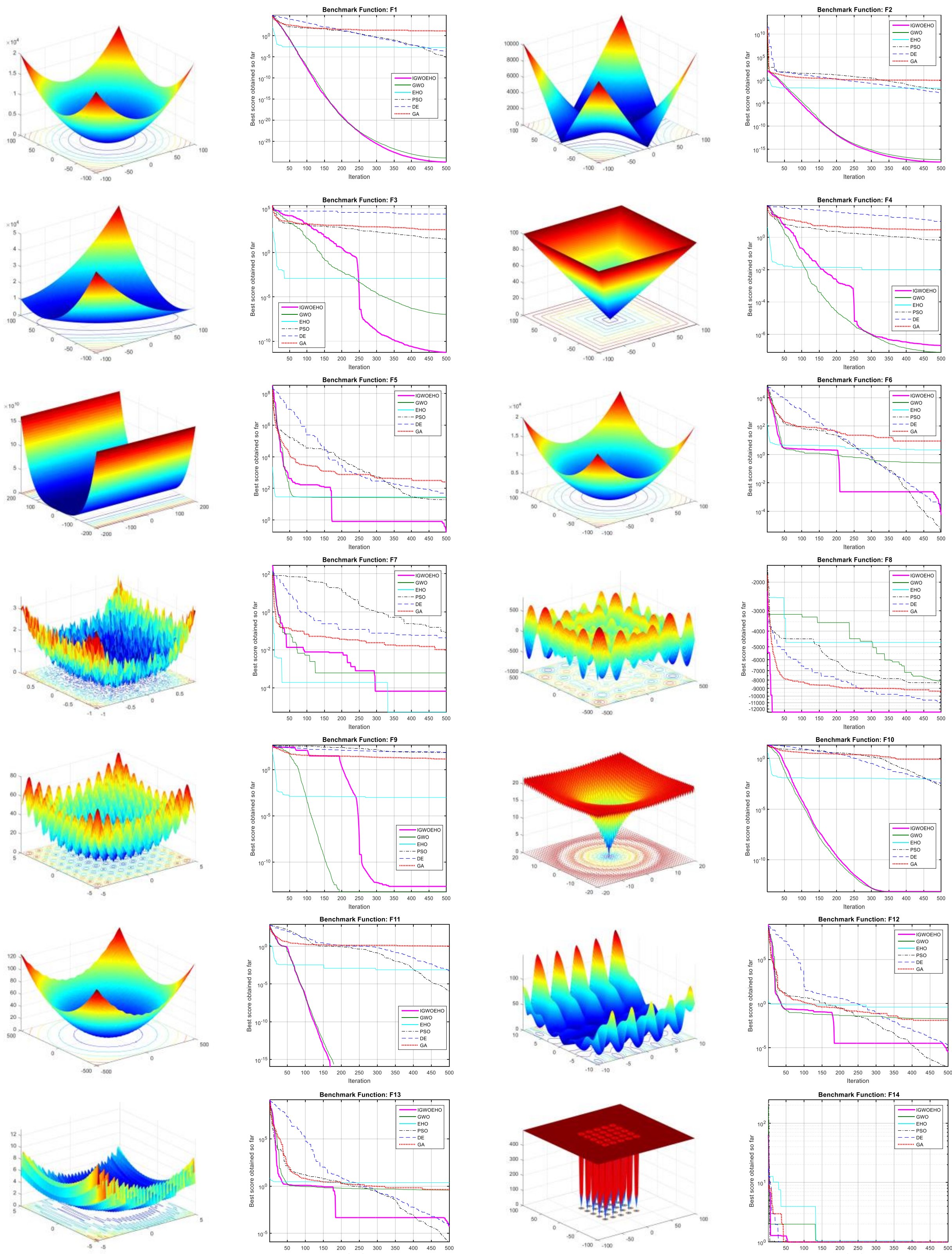
Table 3
Fixed-dimension multi-modal benchmark functions.

Test Function	n	S	f_{min}
$F_{14}(X) = (\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6})^{-1}$	2	$[-65,65]$	1
$F_{15}(X) = \sum_{i=1}^{11} [a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	4	$[-5,5]$	0.00030
$F_{16}(X) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	$[-5,5]$	-1.0316
$F_{17}(X) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10\left(1 - \frac{1}{8\pi}\right) cos x_1 + 10$	2	$[-5,5]$	0.398
$F_{18}(X) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)]$	2	$[-2,2]$	3
$F_{19}(X) = -\sum_{i=1}^{4} c_i\, exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	3	[1,3]	-3.86
$F_{20}(X) = -\sum_{i=1}^{4} c_i\, exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	6	[0,1]	-3.32
$F_{21}(X) = -\sum_{i=1}^{5} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.1532
$F_{22}(X) = -\sum_{i=1}^{7} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.4028
$F_{23}(X) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.5363

Table 4
Parameter settings of optimization algorithms.

GA [56]		DE [57]		PSO [58]		EHO Wang et al. [22]		GWOEHO	
Pc	0.9	Beta-min	0.2	V_{max}	6	Alpha	0.5		
		Beta-max	0.1	W_{max}, W_{min}	0.2-0.9			Beta	0.1
Pm	0.1					Beta	0.1		
		PCR	0.2	C1,C2	2				

Fig. 2 displays the 2D versions of the benchmark functions and convergence rates of the best answer for GWOEHO, GWO, EHO, PSO, DE, GA algorithms.


Benchmark Function: F1
Benchmark Function: F2
Benchmark Function: F3
Benchmark Function: F4
Benchmark Function: F5
Benchmark Function: F6
Benchmark Function: F7
Benchmark Function: F8
Benchmark Function: F9
Benchmark Function: F10
Benchmark Function: F11
Benchmark Function: F12
Benchmark Function: F13
Benchmark Function: F14
Best score obtained so far
Iteration
IGWOEHO
GWO
EHO
PSO
DE
GA

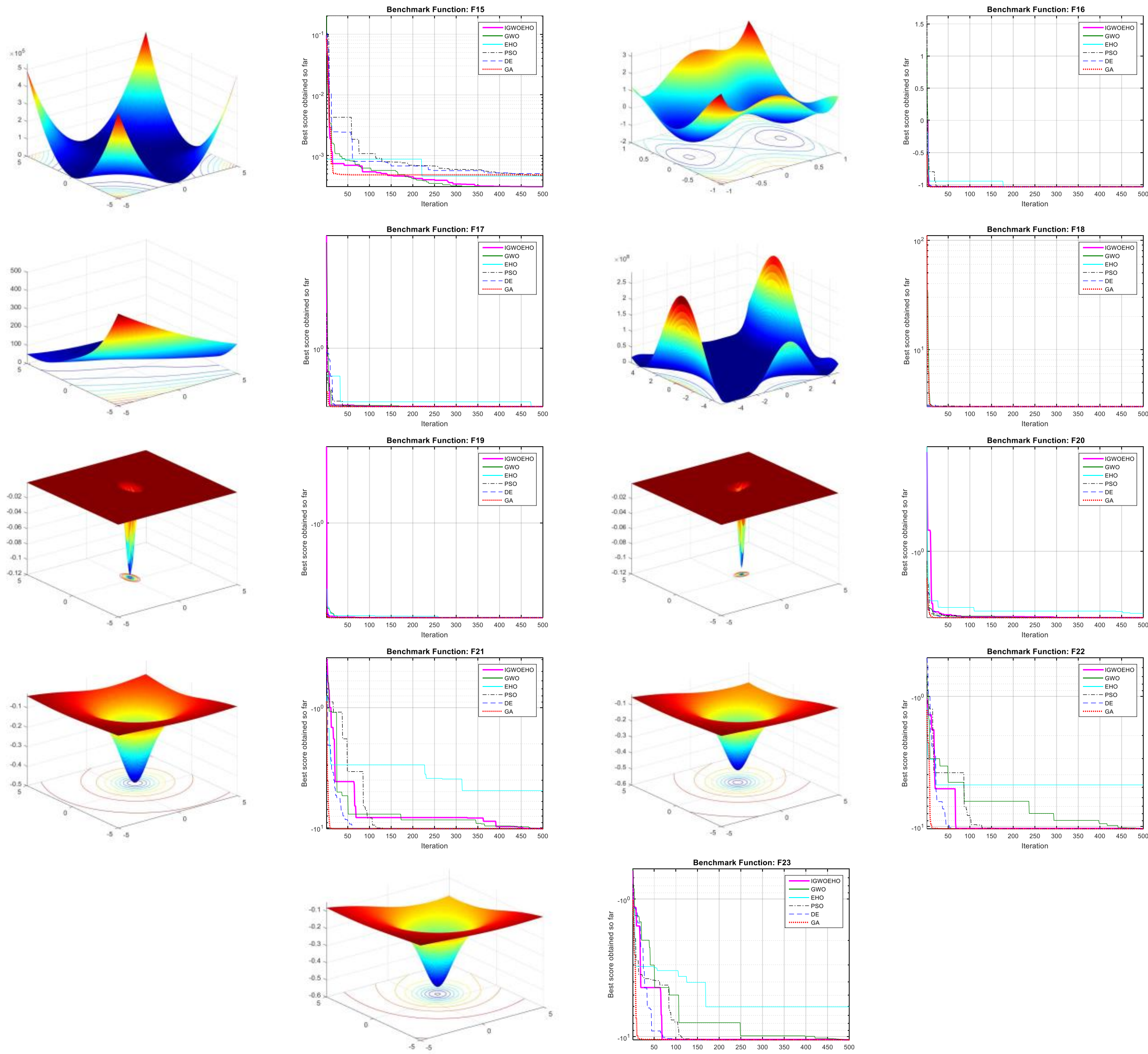


Fig. 2. Convergence history of optimization problems.

The statistical results (standard deviation and average) are stated in Tables 5–7. According to the results, GWOEHO can provide very competitive results.

Table 5
Outcomes of the unimodal benchmark functions.

		GA	DE	PSO	EHO	GWO	EHOGWO
	Ave	1.76E+01	4.07E-04	2.09E-04	1.76E-03	1.12E-27	1.18E-28
	Std.	5.99E+00	1.20E-04	2.01E-04	9.61E-05	1.75E-27	3.92E-28
F1	Best	9.18E+00	1.68E-04	8.11E-06	1.59E-03	1.27E-29	1.41E-30
	Worst	2.75E+01	5.78E-04	7.60E-04	1.92E-03	6.68E-27	1.77E-27
	Rank	6	4	3	5	2	1
	Ave	1.39E+00	2.16E-03	5.03E+00	1.93E-02	8.80E-17	1.93E-09
	Std.	3.06E-01	3.87E-04	6.90E+00	7.49E-04	5.41E-17	1.77E-09
F2	Best	9.27E-01	1.46E-03	7.78E-03	1.79E-02	5.17E-18	1.61E-18
	Worst	2.15E+00	2.95E-03	2.02E+01	2.05E-02	2.10E-16	4.40E-09
	Rank	6	3	4	5	2	1
	Ave	7.66E+02	3.11E+04	7.80E+01	2.93E-03	1.38E-05	4.12E-11
	Std.	2.43E+02	3.71E+03	2.74E+01	7.55E-04	2.42E-05	3.16E-11
F3	Best	4.00E+02	2.30E+04	3.38E+01	1.22E-03	1.09E-07	5.52E-12
	Worst	1.38E+03	3.53E+04	1.42E+02	4.28E-03	1.05E-04	1.12E-10
	Rank	5	6	4	3	2	1
	Ave	3.78E+00	1.32E+01	1.07E+00	1.31E-02	6.73E-07	2.52E-06
	Std.	3.94E-01	1.83E+00	1.77E-01	1.35E-03	7.60E-07	2.10E-06
F4	Best	6.57E-01	9.70E+00	6.57E-01	9.97E-03	7.68E-08	2.05E-07
	Worst	4.33E+00	1.57E+01	1.36E+00	1.56E-02	2.93E-06	6.35E-06
	Rank	4	5	4	3	1	2
	Ave	5.21E+02	1.41E+02	7.33E+01	2.88E+01	2.70E+01	2.37E+00
	Std.	1.99E+02	5.32E+01	5.14E+01	2.17E-02	9.00E-01	4.86E+00
F5	Best	2.55E+02	4.58E+01	1.92E+01	2.88E+01	2.59E+01	1.69E-01
	Worst	9.49E+02	2.37E+02	2.04E+02	2.89E+01	2.87E+01	1.92E+01
	Rank	6	5	2	4	3	1
	Ave	1.84E+01	4.08E-04	2.10E-04	3.12E+00	8.96E-01	8.21E-05
	Std.	6.47E+00	1.25E-04	4.45E-04	3.49E-01	3.16E-01	2.68E-05
F6	Best	8.86E+00	2.32E-04	3.29E-06	2.10E+00	2.57E-01	5.51E-05
	Worst	2.95E+01	5.94E-04	1.84E-03	3.63E+00	1.51E+00	1.51E-04
	Rank	6	3	1	5	4	2
	Ave	2.62E-02	5.57E-02	3.61E+00	1.22E-04	2.34E-03	1.32E-03
	Std.	8.01E-03	9.93E-03	4.09E+00	1.25E-04	1.43E-03	1.02E-03
F7	Best	1.03E-02	4.26E-02	8.28E-02	5.24E-06	6.05E-04	6.61E-05
	Worst	4.46E-02	7.60E-02	1.36E+01	4.46E-04	5.73E-03	4.56E-03
	Rank	4	5	6	1	3	2
Average Rank		5.28	4.43	3.43	3.71	2.43	1.43
Overall Rank		6	5	3	4	2	1

Table 6
Outcomes of the multi-modal benchmark functions.

		GA	DE	PSO	EHO	GWO	EHOGWO
	Ave	-8.32E+03	-9.77E+03	-4.87E+03	-3.68E+03	-5.98E+03	-1.26E+04
	Std.	6.15E+02	4.98E+02	1.42E+03	3.79E+02	1.08E+03	3.71E+00
F8	Best	-9.48E+03	-1.09E+04	-8.30E+03	-4.69E+03	-8.13E+03	-1.26E+04
	Worst	-6.94E+03	-8.60E+03	-2.90E+03	-3.18E+03	-3.11E+03	-1.26E+04
	Rank	3	2	4	6	5	1
	Ave	2.30E+01	8.61E+01	1.11E+02	1.17E-03	2.68E+00	3.92E-13
	Std.	5.85E+00	8.93E+00	2.96E+01	1.27E-04	5.71E+00	8.83E-14
F9	Best	1.38E+01	7.24E+01	6.87E+01	9.57E-04	5.68E-14	2.27E-13
	Worst	3.22E+01	1.05E+02	1.91E+02	1.40E-03	2.50E+01	5.68E-13
	Rank	4	6	5	3	1	2
	Ave	1.89E+00	5.52E-03	2.11E-01	1.05E-02	9.97E-14	2.31E-13
	Std.	3.87E-01	1.00E-03	4.04E-01	4.45E-04	2.13E-14	4.27E-13
F10	Best	8.36E-01	3.69E-03	2.05E-03	9.53E-03	6.84E-14	7.55E-14
	Worst	2.52E+00	7.22E-03	1.16E+00	1.13E-02	1.36E-13	1.99E-12
	Rank	6	4	3	5	1	2
	Ave	1.14E+00	7.05E-03	8.29E-03	2.24E-03	5.60E-03	1.01E-15
	Std.	4.41E-02	7.94E-03	8.12E-03	7.26E-04	1.00E-02	1.82E-15
F11	Best	1.06E+00	3.43E-04	1.16E-06	7.64E-04	0.00E+00	0.00E+00
	Worst	1.23E+00	2.60E-02	2.22E-02	3.42E-03	3.62E-02	4.66E-15
	Rank	5	3	2	4	1	1
	Ave	3.93E-02	5.73E-05	1.04E-02	5.97E-01	4.42E-02	6.51E-06
	Std.	2.46E-02	3.21E-05	4.64E-02	8.91E-02	2.07E-02	2.09E-06
F12	Best	1.28E-02	2.24E-05	7.45E-08	4.10E-01	1.98E-02	3.06E-06
	Worst	1.21E-01	1.47E-04	2.07E-01	7.74E-01	1.05E-01	1.05E-05
	Rank	4	3	1	6	5	2
	Ave	7.95E-01	2.93E-04	6.02E-03	2.84E+00	7.30E-01	1.28E-04
	Std.	2.56E-01	1.56E-04	1.09E-02	1.97E-01	3.08E-01	7.62E-05
F13	Best	4.38E-01	7.70E-05	1.18E-06	2.38E+00	3.75E-01	5.18E-05
	Worst	1.50E+00	6.59E-04	4.40E-02	2.98E+00	1.72E+00	3.39E-04
	Rank	5	3	1	6	4	2
Average Rank		4.5	3.5	2.67	5	3	1.67
Overall Rank		5	4	2	6	3	1

Table 7
Outcomes of the fixed-dimension multi-modal benchmark functions.

		GA	DE	PSO	EHO	GWO	EHOGWO
F14	Ave	9.98E-01	1.24E+00	2.67E+00	2.17E+00	4.82E+00	9.98E-01
	Std.	4.30E-11	1.10E+00	2.75E+00	7.34E-01	4.06E+00	1.40E-10
	Best	9.98E-01	9.98E-01	9.98E-01	1.01E+00	9.98E-01	9.98E-01
	Worst	9.98E-01	5.93E+00	1.08E+01	3.97E+00	1.27E+01	9.98E-01
	Rank	1	1	1	1	1	1
F15	Ave	2.68E-03	7.14E-04	1.02E-02	1.42E-03	4.40E-03	4.72E-04
	Std.	6.05E-03	9.77E-05	9.78E-03	7.58E-04	8.19E-03	1.93E-04
	Best	4.82E-04	5.06E-04	4.67E-04	4.60E-04	3.07E-04	3.08E-04
	Worst	2.04E-02	7.86E-04	2.26E-02	3.46E-03	2.04E-02	1.22E-03
	Rank	4	6	5	3	1	2
F16	Ave	-1.03E+00	-1.03E+00	-1.03E+00	-9.88E-01	-1.03E+00	-1.03E+00
	Std.	3.74E-08	2.28E-16	2.10E-16	4.62E-02	1.54E-08	2.36E-08
	Best	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
	Worst	-1.03E+00	-1.03E+00	-1.03E+00	-8.47E-01	-1.03E+00	-1.03E+00
	Rank	1	1	1	1	1	1
F17	Ave	3.98E-01	3.98E-01	3.98E-01	4.07E-01	3.98E-01	3.99E-01
	Std.	3.68E-07	0.00E+00	0.00E+00	1.09E-02	2.68E-06	2.76E-03
	Best	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01
	Worst	3.98E-01	3.98E-01	3.98E-01	4.45E-01	3.98E-01	4.08E-01
	Rank	1	1	1	2	1	1
F18	Ave	3.00E+00	3.00E+00	3.00E+00	3.23E+00	3.00E+00	3.00E+00
	Std.	7.81E-07	4.78E-16	2.19E-15	5.42E-01	3.82E-05	1.91E-05
	Best	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
	Worst	3.00E+00	3.00E+00	3.00E+00	4.93E+00	3.00E+00	3.00E+00
	Rank	1	1	1	1	1	1
F19	Ave	-3.86E+00	-3.86E+00	-3.86E+00	-3.81E+00	-3.86E+00	-3.86E+00
	Std.	3.67E-08	2.28E-15	2.89E-03	2.12E-02	3.14E-03	3.04E-03
	Best	-3.86E+00	-3.86E+00	-3.86E+00	-3.85E+00	-3.86E+00	-3.86E+00
	Worst	-3.86E+00	-3.86E+00	-3.85E+00	-3.76E+00	-3.85E+00	-3.85E+00
	Rank	1	1	1	2	1	1
F20	Ave	-3.29E+00	-3.32E+00	-3.23E+00	-2.90E+00	-3.26E+00	-3.21E+00
	Std.	5.59E-02	9.22E-06	1.23E-01	1.65E-01	8.07E-02	1.55E-01
	Best	-3.32E+00	-3.32E+00	-3.32E+00	-3.07E+00	-3.32E+00	-3.32E+00
	Worst	-3.20E+00	-3.32E+00	-2.84E+00	-2.49E+00	-3.10E+00	-2.84E+00
	Rank	1	1	1	2	1	1
F21	Ave	-7.65E+00	-9.24E+00	-7.01E+00	-3.87E+00	-9.14E+00	-9.63E+00
	Std.	3.54E+00	2.18E+00	3.34E+00	4.88E-01	2.08E+00	1.17E+00
	Best	-1.02E+01	-1.02E+01	-1.02E+01	-4.89E+00	-1.02E+01	-1.02E+01
	Worst	-2.63E+00	-2.68E+00	-2.63E+00	-3.22E+00	-5.06E+00	-5.69E+00
	Rank	1	1	1	2	1	1
F22	Ave	-8.27E+00	-1.03E+01	-8.25E+00	-4.04E+00	-1.04E+01	-1.02E+01
	Std.	3.37E+00	6.72E-01	3.08E+00	3.88E-01	1.10E-03	5.49E-01
	Best	-1.04E+01	-1.04E+01	-1.04E+01	-4.79E+00	-1.04E+01	-1.04E+01
	Worst	-2.75E+00	-7.40E+00	-2.77E+00	-3.38E+00	-1.04E+01	-8.24E+00
	Rank	1	1	1	2	1	1
F23	Ave	-7.62E+00	-1.05E+01	-8.46E+00	-4.23E+00	-1.03E+01	-1.04E+01
	Std.	3.69E+00	2.27E-01	2.97E+00	4.79E-01	1.21E+00	2.98E-01
	Best	-1.05E+01	-1.05E+01	-1.05E+01	-6.07E+00	-1.05E+01	-1.05E+01
	Worst	-2.42E+00	-9.52E+00	-2.42E+00	-3.69E+00	-5.13E+00	-9.56E+00
	Rank	1	1	1	2	1	1
Average rank		3.348	2.913	2.348	3.217	1.913	1.348
Overall Rank		6	4	3	5	2	1

6.1. Exploitation analysis

From Table 5, it is clear that GWOEHO can produce very competitive outcomes. According to the results, the suggested GWOEHO algorithm has the highest overall ranking of the whole benchmark suites. It can be said that the unimodal functions are usually utilized to evaluate the algorithms' exploitation ability. GWO, PSO, EHO, DE, and GA are in the following ranks, respectively. Hence, these outcomes present the extreme ability of GWOEHO in terms of exploiting the optimum. And this is because of the previous dissection of the suggested exploitation operators.

6.2. Exploration analysis

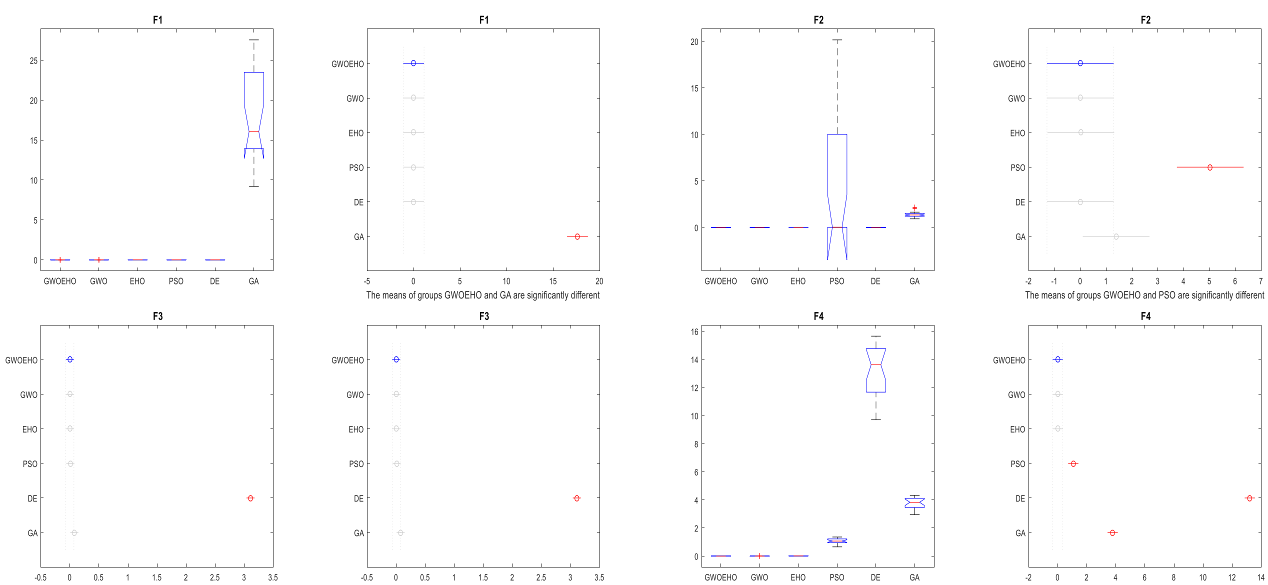
As opposed to the unimodal functions, multi-modal functions have many local optima, expanding exponentially with dimension. This ability makes them appropriate for benchmarking the exploration performance of an algorithm. Tables 6 and 7 present the outcomes for multi-modal benchmark functions and those with fixed dimensions. From the results, it is clear that the suggested GWOEHO has the best ranking when compared to the other algorithms. Meanwhile, PSO, GWO, DE, GA, and EHO are in the following ranks in relation to the multi-modal benchmark functions. In the case of multi-modal benchmark functions with rigid dimensions, GWOEHO still got the best rank, and GWO, PSO, DE, EHO, and GA are after GWOEHO, respectively.

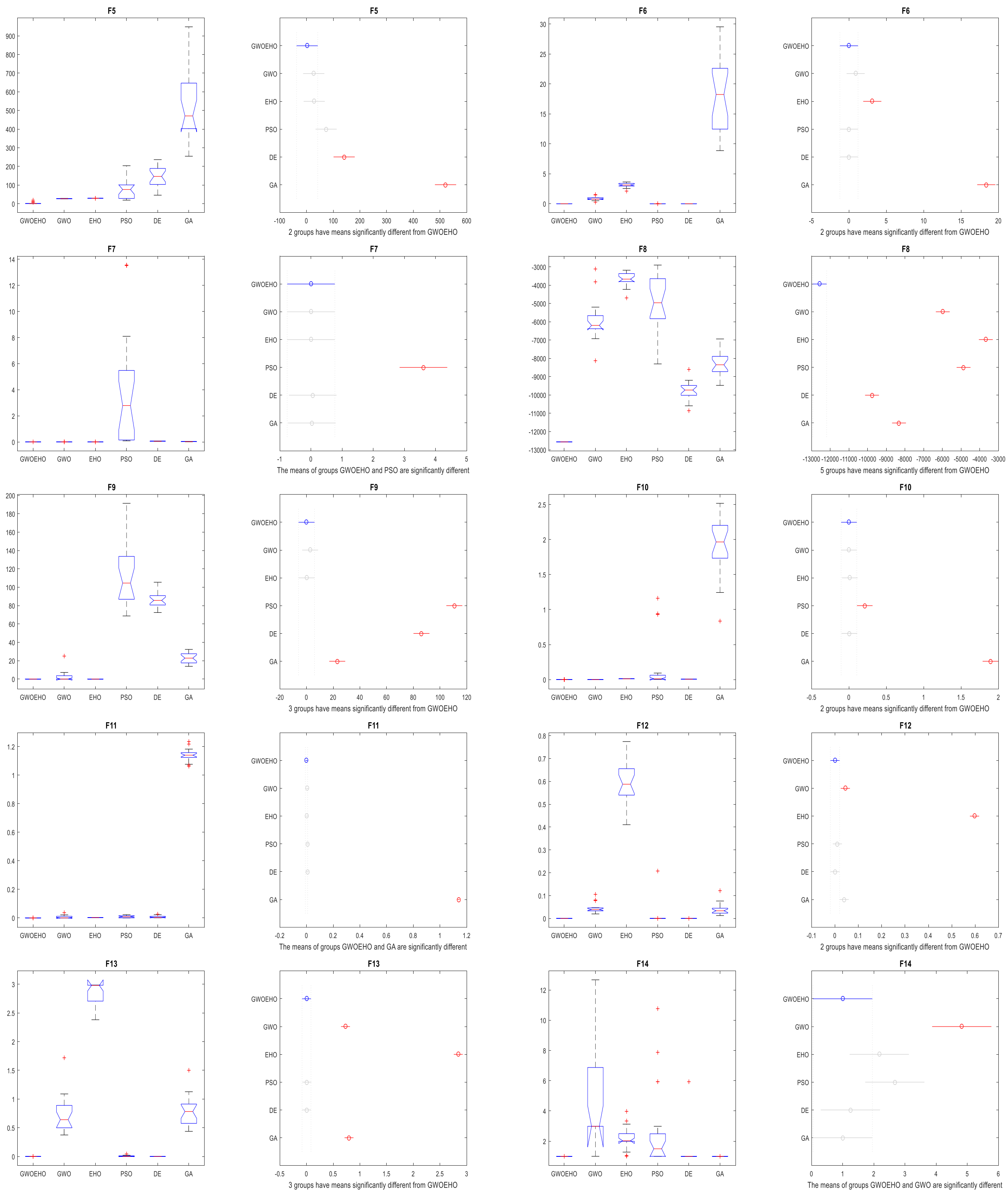
6.3. Statistical testing

Wilcoxon's rank-sum test as a nonparametric statistical test [59] is performed to determine the efficiency of the suggested GWOEHO algorithm when compared to the other algorithms. The test was accomplished by utilizing the outcomes of the suggested GWOEHO in each test function and in comparison with other algorithms at 5% importance. Table 8 displays the p-values attained by the test, where the p-values less than 0.05 indicates that the null hypothesis is not acceptable and therefore must be declined, i.e., there is a considerable variance at a level of 5%. In contrast, the p-values (bigger than 0.05) are underlined, meaning no considerable variance exists among the compared values. By analyzing the results obtained, given that in most comparisons, the values of the p-values are less than 0.05, which confirms that the improvement obtained by the suggested GWOEHO is statistically considerable. The box-plot and MCT results of the considered problems are presented in Fig. 3.

Table 8
p-values obtained from the rank-sum test on different benchmark functions

Function	GWO	EHO	PSO	DE	GA
1	2.04E-05	6.80E-08	6.80E-08	6.80E-08	6.80E-08
2	1.20E-06	6.80E-08	6.80E-08	6.80E-08	6.80E-08
3	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08
4	0.000758	6.80E-08	6.80E-08	6.80E-08	6.80E-08
5	6.70E-08	6.70E-08	7.79E-08	6.70E-08	6.70E-08
6	6.80E-08	6.80E-08	0.323482	6.80E-08	6.80E-08
7	0.009045	6.01E-07	6.80E-08	6.80E-08	6.80E-08
8	5.37E-08	5.37E-08	5.37E-08	5.37E-08	5.37E-08
9	0.673626	6.38E-08	6.38E-08	6.38E-08	6.38E-08
10	0.415586	6.67E-08	6.67E-08	6.67E-08	6.67E-08
11	0.399648	2.96E-08	2.96E-08	2.96E-08	2.96E-08
12	6.80E-08	6.80E-08	7.41E-05	6.80E-08	6.80E-08
13	6.80E-08	6.80E-08	0.285305	0.000179	6.80E-08
14	5.87E-06	6.80E-08	1	2.78E-07	1.19E-06
15	1	6.92E-07	4.41E-07	3.07E-06	2.06E-06
16	0.036048	6.80E-08	1.94E-08	8.01E-09	0.002319
17	0.218406	9.75E-06	8.01E-09	8.01E-09	6.76E-07
18	2.28E-05	0.470696	1.98E-08	5.86E-09	0.855798
19	0.323482	6.80E-08	6.49E-05	8.01E-09	6.80E-08
20	0.364842	5.87E-06	0.093541	2.66E-07	0.002139
21	0.946074	6.76E-08	1	0.040991	0.13321
22	0.00148	6.79E-08	0.101388	4.98E-06	0.038506
23	0.027447	6.75E-08	0.105926	1.92E-06	0.424775




F5
GWOEHO
GWO
EHO
PSO
DE
GA
F5
2 groups have means significantly different from GWOEHO
F6
F6
2 groups have means significantly different from GWOEHO
F7
F7
The means of groups GWOEHO and PSO are significantly different
F8
F8
5 groups have means significantly different from GWOEHO
F9
F9
3 groups have means significantly different from GWOEHO
F10
F10
2 groups have means significantly different from GWOEHO
F11
F11
The means of groups GWOEHO and GA are significantly different
F12
F12
2 groups have means significantly different from GWOEHO
F13
F13
3 groups have means significantly different from GWOEHO
F14
F14
The means of groups GWOEHO and GWO are significantly different

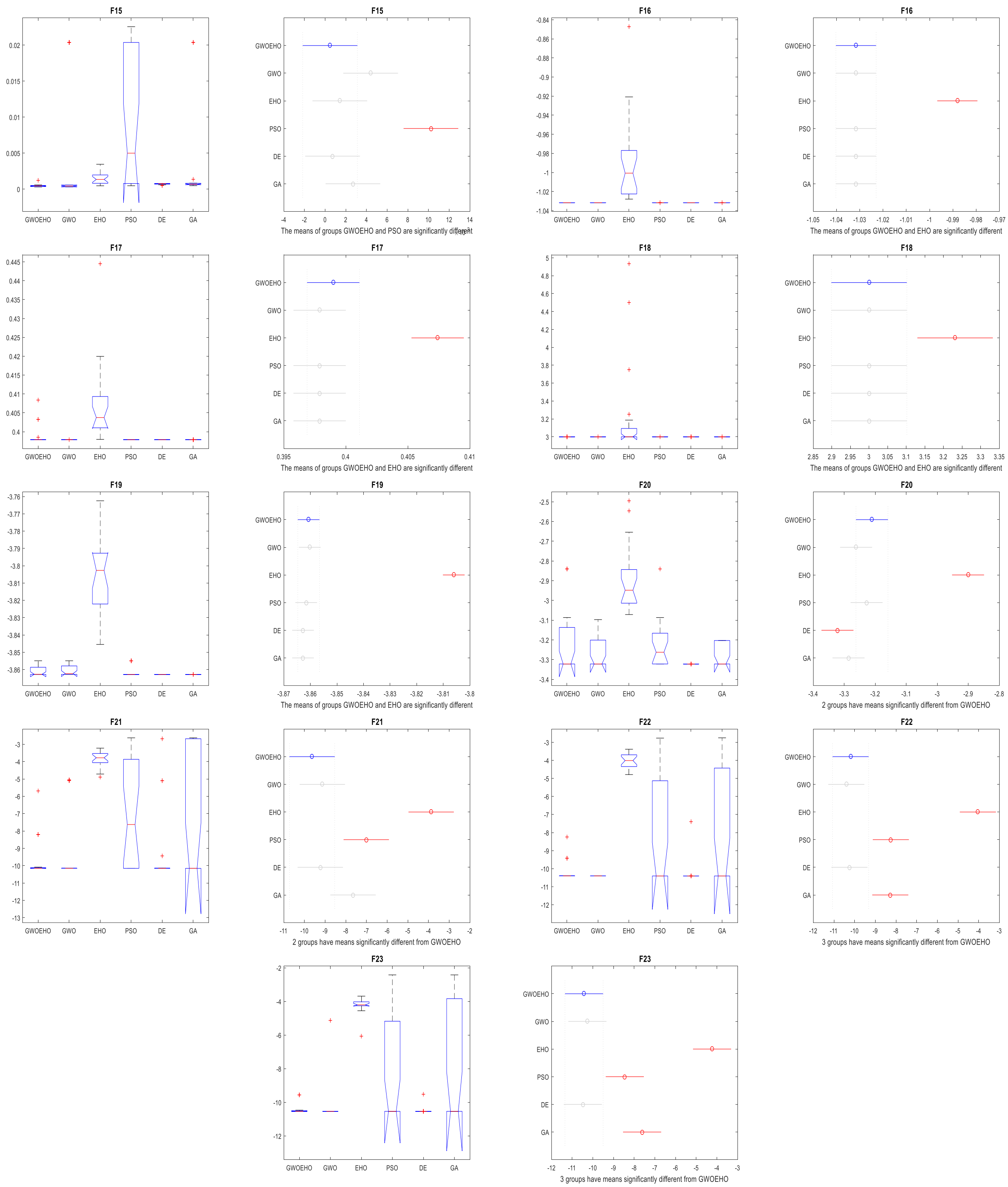


Fig. 3. Box-plot of F1 to F23 objective function using the reported optimizers.

7. GWOEHO for classical engineering challenges

In order to solve the classical engineering challenges, six constrained engineering design challenges including welded beam, three bar truss design, pressure vessel, tension/compression spring, speed reducer design, and tabular design column, are employed. Updating search agents can be very challenging due to the constraints and the possibility of affecting the amount of function. However, the search agents are updated without changing the algorithm and the direct relationship between the search agents and the main function. In this process, by violating any constraints, a large value is allocated to the main function's fitness value. Thus, if the best search agents include penalty functions, the algorithm will be replaced automatically in the following process. A variety of penalty functions can be used to penalize based on the amount of the violation.

The GWOEHO algorithm is compared to GA, DE, PSO, EHO, GWO, and algorithms for verifying the results. The previously mentioned parameters in Table 4 are also used in the algorithms for engineering design problems.

7.1. Tension/compression spring design

This problem minimizes tension/compression spring weight, as illustrated in Fig. 4 [60–62].

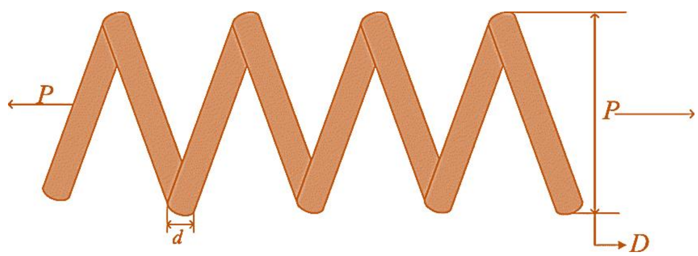


Fig. 4. Schematic of the Tension/compression spring.

Wire diameter (d), mean coil diameter (D), and the number of active coils (N) are three variables of the problem and, shear stress, surge frequency, and minimum deflection are defined as design constraints. The mathematical formulation of this challenge is as follows:

Consider $\vec{X} = [X_1 \ X_2 \ X_3] = [dDN]$

Minimize $f(\vec{X}) = (X_3 + 2)X_3X_1^2$

Subject to (14)

$$g_1 = 1 - \frac{X_3X_2^3}{71785X_1^4} \leq 0$$

$$g_2 = 1 - \frac{4X_2^2 + X_1X_2}{12566(X_2X_1^3 - X_1^4)} + \frac{1}{5108X_1^2} \leq 0$$

$$g_3 = 1 - \frac{140.45X_1}{X_2^2X_3} \leq 0$$

$$g_4 = \frac{X_1 + X_2}{1.5} - 1 \leq 0$$

Variable range $0.05 \leq X_1 \leq 2$

$0.25 \leq X_2 \leq 1.3$

$2 \leq X_3 \leq 15$

This problem was tackled by both heuristic and mathematical methods. With utilizing PSO, Ha and Wang attempted to solve this problem [63]. Besides, there are various algorithms have been utilized as heuristic optimizers for solving this problem including the Harmony Search (HS) [64], Differential Evolution (DE) [65] , GA [66], and Evolution Strategy (ES) [67]. In addition, the mathematical optimization technique [61] and the numerical optimization technique (constraints correction at constant cost) [60] are two mathematical methods that have been employed to solve this problem. Table 9 presented the results of GWO in comparison with the outcomes of the above-mentioned techniques.

Table 9
Comparison of results for tension/compression spring design problem.

Algorithm	Optimum variables			Optimum weight
	d	D	N	
GWO	0.05169	0.356737	11.28885	0.0126660
GSA	0.050276	0.32368	13.52541	0.0127022
PSO	0.051728	0.357644	11.24454	0.0126747
ES	0.051989	0.363965	10.89052	0.0126810
GA	0.05148	0.351661	11.6322	0.0127048
HS	0.051154	0.349871	12.07764	0.0126706
DE	0.051609	0.354714	11.41083	0.0126702
Mathematical optimization	0.053396	0.39918	9.1854	0.0127303
Constraint correction	0.05	0.3159	14.25	0.0128334
GWOEHO	0.0508	0.3350	12.7033	0.0127035

Note that in evaluating the algorithm, 30 algorithm populations have been selected. The number of tribes in the algorithm is set to five, and the maximum number of iterations is equal to 500. The results and outputs of the program after twenty independent runs are presented in the table 10. Similar penalty function for all algorithms were employed to perform a fair comparison [68]. This table shows and compares the results of the six algorithms GA, DE, PSO, EHO, GWO, GWOEHO. The typical convergence history of the six algorithms is displayed in Fig. 5.

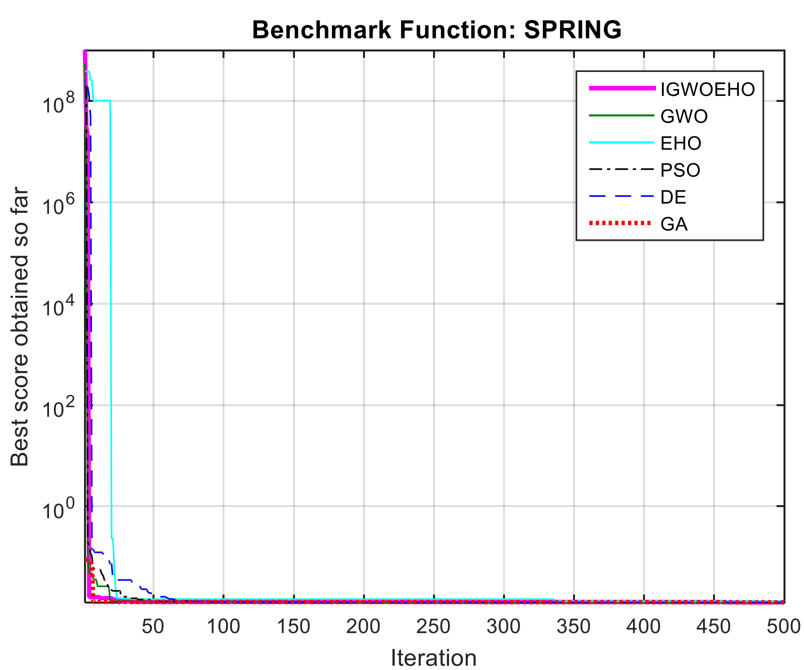


Fig. 5. Typical convergence history.

Table 10
Comparison of outputs for tension/compression spring design problem.

	GA	DE	PSO	EHO	GWO	GWOEHO
Ave	0.0151251	0.0130328	0.0140182	0.0142334	0.0127631	0.0128155
Std.	0.0016294	0.0003045	0.0019070	0.0003761	0.0001007	0.0001535
Best	0.0127486	0.0127103	0.0126970	0.0134147	0.0126794	0.0127034
Worst	0.0176103	0.0138408	0.0177731	0.0150044	0.0131131	0.0131679
Rank	5	4	2	6	1	3

7.2. Welded beam design

This challenge aims to reduce the fabrication cost of a welded beam, as presented in Fig. 6 [66]. The following constraints were used:

- Shear stress (τ).
- Bending stress in the beam (σ).
- Buckling load on the bar (P).
- End deflection of the beam (δ).
- Side constraints.

Four variables including the thickness of the bar (h), the height of the bar (d), the length of an attached part of the bar (L), and the thickness of weld (w) are the main factors affecting this problem. The mathematical formulation can be found as follows:

Minimize

$$min\, f(w.L.d.h) = 1.1047w^2L + 0.04811dh(14.0 + L) \quad (15)$$

Subject to

$$g_1 = \tau - 13600 \leq 0$$
$$g_2 = \sigma - 30000 \leq 0$$
$$g_3 = w - h \leq 0$$
$$g_4 = 0.1047w^2 + 0.04811hd(14 + L) - 0.5 \leq 0 \quad (16)$$
$$g_5 = 0.125 - w \leq 0$$
$$g_6 = \delta - 0.25 \leq 0$$
$$g_7 = 6000 - P \leq 0$$

where

$$\sigma = \frac{504000}{hd^2}$$
$$Q = 6000\left(14 + \frac{L}{2}\right)$$
$$D = \frac{1}{2}\sqrt{(L^2 + (w + d)^2}$$
$$J = \sqrt{2}wL\left[\frac{L^2}{6} + \frac{(w + d)^2}{2}\right]$$
$$\delta = \frac{65856}{30000hd^3}$$
$$\beta = \frac{QD}{J}$$
$$\alpha = \frac{6000}{\sqrt{2}wL}$$

$$\tau = \sqrt{\alpha^2 + \frac{\alpha\beta L}{D} + \beta^2}$$

$$P = 0.61423 \times 10^6 \frac{dh^3}{6}\left(1 - \frac{d\sqrt{30/48}}{28}\right)$$

$$\sigma = \frac{504000}{hd^2}$$

$$Q = 6000\left(14 + \frac{L}{2}\right)$$

$$D = \frac{1}{2}\sqrt{(L^2 + (w+d)^2}$$

$$J = \sqrt{2}wL\left[\frac{L^2}{6} + \frac{(w+d)^2}{2}\right]$$

$$\delta = \frac{65856}{30000hd^3}$$

$$\beta = \frac{QD}{J}$$

$$\alpha = \frac{6000}{\sqrt{2}wL}$$

$$\tau = \sqrt{\alpha^2 + \frac{\alpha\beta L}{D} + \beta^2}$$

$$P = 0.61423 \times 10^6 \frac{dh^3}{6}\left(1 - \frac{d\sqrt{30/48}}{28}\right)$$

Variable range

$0.1 \le L.d \le 10.0$

$0.1 \le w.h \le 2.0$

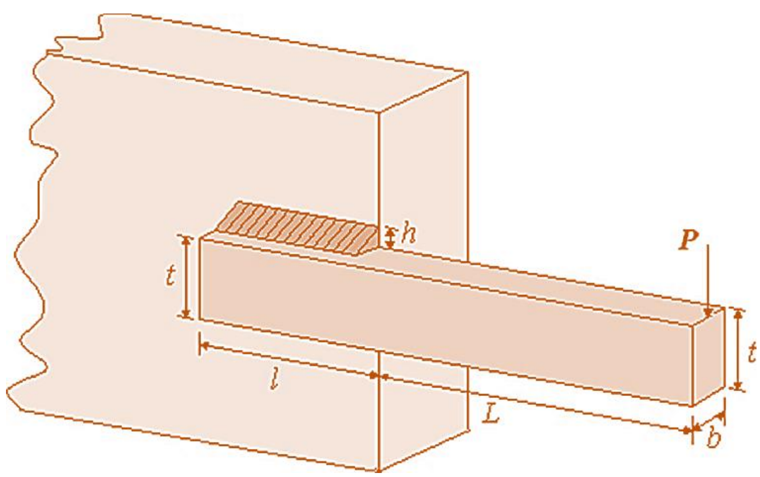


Fig. 6. Schematic of the welded beam.

Coello [69] and Deb [70,71] utilized GA, on the other hand Lee and Geem [72] employed HS for solving this problem. Ragsdell and Philips [73] to solve this problem, employed five different mathematical methods named: Stewart's successive linear approximation, Simplex method, Griffith, Richardson's random method, and Davidon-Fletcher-Powell. Table 11 displayed the comparison outputs for these mathematical approaches.

Table 11
Comparison outputs of the welded beam design problem.

Algorithm	Optimum variables				Optimum cost
	h	l	t	b	
GWO	0.205676	3.478377	9.03681	0.205778	1.72624
GSA	0.182129	3.856979	10.0000	0.202376	1.879952
GA	N/A	N/A	N/A	N/A	1.8245
GA	N/A	N/A	N/A	N/A	2.3800
GA	0.2489	6.1730	8.1789	0.2533	2.4331
HS	0.2442	6.2231	8.2915	0.2443	2.3807
Random	0.4575	4.7313	5.0853	0.6600	4.1185
simplex	0.2792	5.6256	7.7512	0.2796	2.5307
David	0.2434	6.2552	8.2915	0.2444	2.3841
APPROX	0.2444	6.2189	8.2915	0.2444	2.3815
GWOEHO	0.2044	3.2813	9.0357	0.2058	1.697348

Note that in evaluating the algorithm, 30 algorithm populations have been selected by example.

The number of tribes in the algorithm is considered to be five tribes. The number of loops of the algorithm is equal to 500 cycles.

The results and outputs of the program after twenty runs from the program are presented in the following tables. A comparable penalty function for GWOEHO was utilized for performing a fair comparison [68]. Table 12 shows the results of the six algorithms GA, DE, PSO, EHO, GWO, GWOEHO.

Table 12
Comparison of outputs for Welded beam design problem.

	GA	DE	PSO	EHO	GWO	GWOEHO
Ave	1.9080	1.9346	1.7157	2.9186	1.7016	1.7038
Std.	0.1462	0.1423	0.0402	0.3243	0.0038	0.0061
Best	1.7071	1.7287	1.6952	2.3160	1.6976	1.6973
Worst	2.1410	2.2020	1.7938	3.4147	1.7100	1.7189
Time	2.7980	1.8342	0.4185	3.7557	0.5500	3.6043
Rank	4	5	1	6	3	2

Fig. 7 presents the typical convergence history of the six algorithms GA, DE, PSO, EHO, GWO, GWOEHO and are compared.

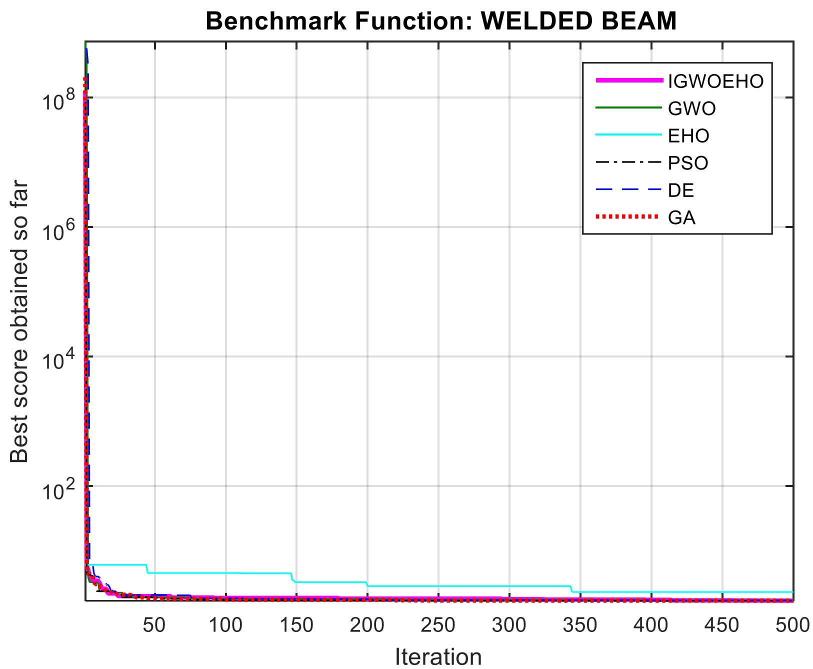


Fig. 7. Typical convergence history

7.3. Pressure vessel design

As presented in Fig. 8, pressure vessel design minimizes the total cost of forming, welding of a cylindrical vessel, and material. The following four variables were employed for pressure vessel design in this study.

- Thickness of the shell (T_s).
- Thickness of the head (T_h).
- Inner radius (R).
- Length of the cylindrical section without considering the head (L).

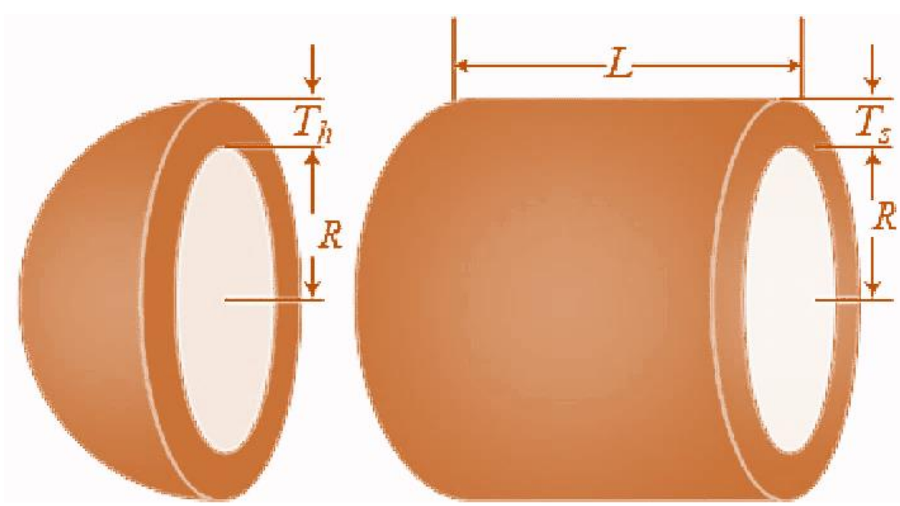


Fig. 8. Schematic of Pressure vessel.

The pressure vessel design is subject to four constraints. All the constraints and the problem are formulated as follows:

$$Minimize\ f(T_s, T_h, R, L) = 0.6224T_sRL + 1.7781T_hR^2 + 3.1661T_s^2L + 19.84T_s^2R$$

Subject to

$$g_1 = -T_s + 0.0193R \leq 0$$
$$g_2 = -T_h + 0.0095R \leq 0$$
$$g_3 = -\pi R^2 L - \frac{4}{3}\pi R^3 + 1.296.000 \leq 0 \quad (17)$$
$$g_4 = L - 240 \leq 0$$

Variable range

$1 \times 0.0625 \ \leq T_s \leq 99 \times 0.0625$
$1 \times 0.0625 \ \leq T_h \leq 99 \times 0.0625$
$10 \leq R \leq 200$
$10 \leq L \leq 200$

Pressure vessel design is a popular problem between researchers and there are so many solutions for this problem in several studies. The heuristic approaches that have been employed to improve this problem are: PSO [63], GA [62,66,74], ES [67], DE [65], and ACO [75]. In addition, Branch-and-bound [76] and augmented Lagrangian Multiplier [77] are two mathematical approaches which utilized in this study. The outputs of pressure vessel design are presented in Table 13.

Table 13
Comparison outputs for pressure vessel design problem.

Algorithm	Optimum variables				Optimum cost
	T_s	T_h	R	L	
GWO	0.812500	0.4345	42.089181	176.758731	6051.563900
GSA	1.125000	0.625	55.9886598	84.454203	8538.8359
PSO	0.812500	0.4375	42.091266	176.746500	6061.077700
GA	0.812500	0.4345	40.323900	200.000000	6288.744500
GA	0.812500	0.4375	42.097398	176.654050	6059.946300
GA	0.937500	0.5	48.329000	112.679000	6410.381100
ES	0.812500	0.4375	42.098087	176.640518	6059.745600
DE	0.812500	0.4375	42.098411	176.637690	6059.734000
ACO	0.812500	0.4375	42.103624	176.572656	6059.088800
Larangian Multiplier	1.125000	0.625	58.291000	43.690000	7198.042800
Branch-bound	1.125000	0.625	47.700000	117.701000	8129.103600
GWOEHO	0.8125	0.4375	42.097985	176.647321	6059.876173

Note that in evaluating the algorithm, 30 algorithm populations have been selected by example.

The number of tribes in the algorithm is considered to be five tribes. The number of loops of the algorithm is equal to 500 cycles.

The results and outputs of the program after twenty runs from the program are presented in the following tables. A comparable penalty function for GWOEHO was utilized for performing a fair comparison. Table 14 indicates the comparison outputs of the six algorithms GA, DE, PSO, EHO, GWO, GWOEHO.

Table 14
Comparison of outputs for Pressure vessel design problem.

	GA	DE	PSO	EHO	GWO	GWOEHO
Ave	6.737E+03	6.319E+03	6.526E+03	1.095E+04	6.369E+03	6.207E+03
Std.	4.034E+02	2.474E+02	3.634E+02	1.727E+03	4.801E+02	3.181E+02
Best	6.085E+03	6.067E+03	6.090E+03	7.971E+03	6.061E+03	6.0598E+03
Worst	7.399E+03	6.824E+03	7.333E+03	1.381E+04	7.375E+03	7.2839E+03
Rank	4	3	5	6	2	1

Fig. 9 presents the typical convergence history of the six algorithms GA, DE, PSO, EHO, GWO, GWOEHO and are compared.

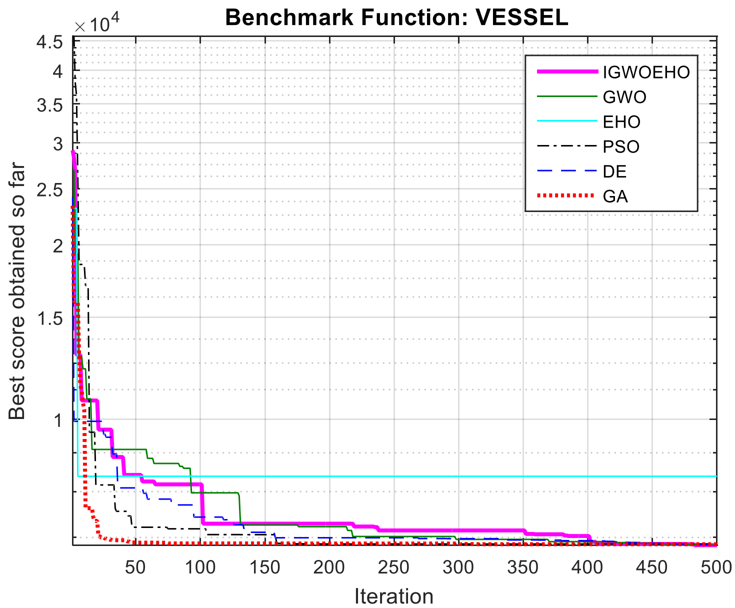


Fig. 9. Typical convergence history.

7.4. Three bar truss design

The structure of three bar truss design is presented in Fig. 10. The reduction of the volume related to the stress constraints of the truss members (on each side) is the main objective of this design. In this regard, the following mathematical approach is employed in current work:

$$
\begin{aligned}
&\text{Minimize } f(A_1, A_2) = \left(2\sqrt{2}\,A_1 + A_2\right) \times L \\
&g_1 = \frac{\sqrt{2}\,A_1 + A_2}{\sqrt{2}\,A_1^2 + 2A_1A_2} P - \sigma \leq 0 \\
&g_2 = \frac{A_2}{\sqrt{2}\,A_1^2 + 2A_1A_2} P - \sigma \leq 0 \\
&g_3 = \frac{1}{A_1 + \sqrt{2}\,A_2} P - \sigma \leq 0 \\
&0 \leq A_1, A_2 \leq 1
\end{aligned}
\tag{18}
$$

where l = 100 cm, P = 2 KN/cm 2, σ= 2 KN/cm2

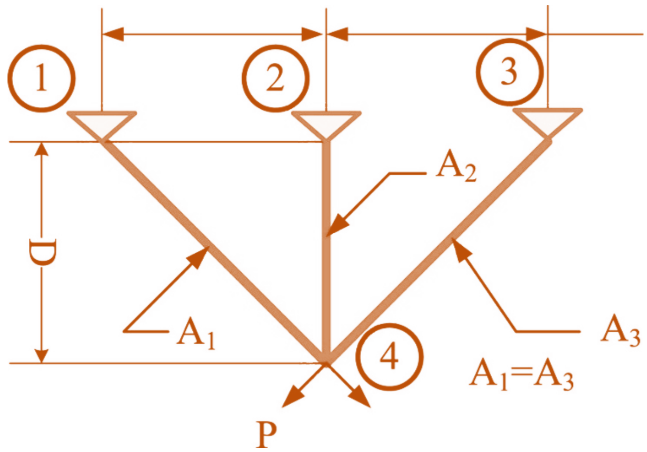


Fig. 10. Three bar truss design.

Various techniques including a swarm-like based approach [78,79], cuckoo search [17], dynamic stochastic selection differential evolution [80], evolutionary computational technique [81], GSA-GA [82] and convexification strategies [83] were employed by researchers to solve this benchmark problem. Table 15 presents these techniques and summarizes their results along. In addition of that, the statistical measures of these approaches are provided in Table 16.

Table 15
Comparison of the best solution for the three-bar truss design problem.

	A_1	A_2	g_1	g_2	g_3	f
Hernandez [84]	0.788	0.408	NA[a]	NA	NA	263.9
Ray and Saini [79]	0.795	0.395	-0.00169	-0.26124	-0.74045	264.3
Ray and Liew [78]	0.788621037	0.408401334	NA	-1.46392765	-0.536072358	263.8958466
Raj et al. [81]	0.78976441	0.40517605	NA	-1.4675992	-0.53240078	263.89671
Tsai [85]	0.788	0.408	NA	-0.2674	-0.73178	263.68
Zhang et al. [86]	0.788675136	0.408248287	NA	-1.46410161	-0.5358983	263.8958434
Gandomi et al. [17]	0.78867	0..40902	-0.00029	-0.26853	-0.73176	263.9716
GSA-GA [82]	0.788676171	0.408245358	NA	-1.4641049	-0.535895	263.8958433
GWOEHO	0.788353026	0.409169566	-0.0000	-1.4631	-0.7552	263.8968650

[a] not available

Table 16
Statistically result of different methods for the truss-bar problem**.**

Method	Best	Mean	Worst	Std.	Median
Ray and Liew	263.8958	263.9033	263.9698	1.26E-02	263.8989
Zhang et al.	263.8958	263.8958	263.8958	9.72E-07	263.8958
Gandomi et al.	263.9716	264.0669	NA[a]	9.00E-05	NA
GSA-GA	263.8958	263.8958	263.8958	5.34E-07	263.8958
GWOEHO	263.8968	263.9074	263.9314	0.009777	NA

[a] not available

Note that in evaluating the algorithm, 30 algorithm populations have been selected by example. The number of tribes in the algorithm is considered to be five tribes. The number of loops of the algorithm is equal to 500 cycles.

The results and outputs of the program after twenty runs from the program are presented in the following tables. A comparable penalty function for GWOEHO was utilized for performing a fair comparison. Table 17 shows the results of the six algorithms GA, DE, PSO, EHO, GWO, GWOEHO and are compared.

Table 17
Comparison of the outputs for three bar truss design.

	GA	DE	PSO	EHO	GWO	GWOEHO
Ave	263.9076	263.8961	264.8443	264.4008	263.9016	263.9074
Std.	0.0154	0.0002	4.2364	0.4353	0.0059	0.0098
Best	263.8964	263.8959	263.8959	264.0066	263.8962	263.8969
Worst	263.9459	263.8964	282.8427	265.7766	263.9170	263.9314
Rank	3	1	1	5	2	4

Fig. 11 presents the typical convergence history of the six algorithms GA, DE, PSO, EHO, GWO, GWOEHO and are compared.

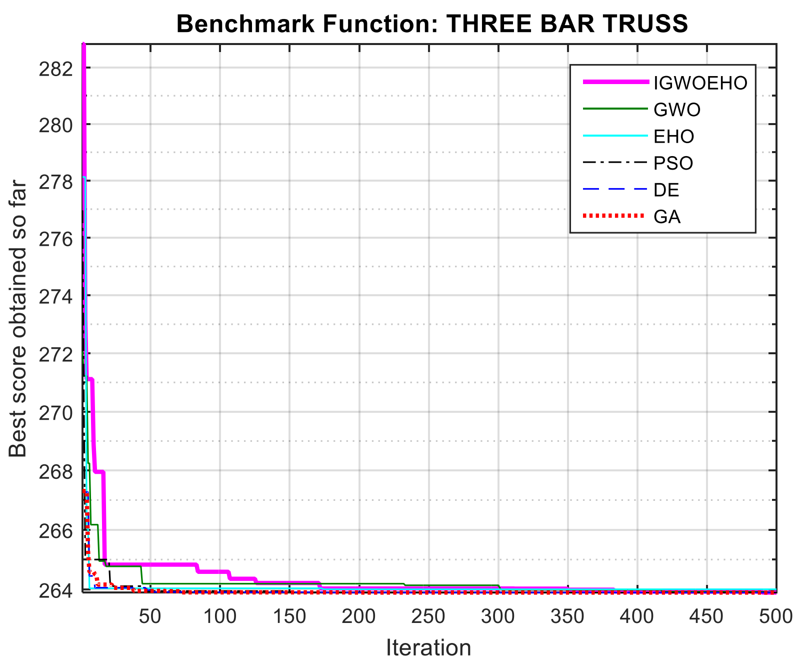


Fig. 11. Typical convergence history.

7.5. Speed reducer design

One of the most critical benchmark design problems is speed reducer problem that was proposed by [87] and presented in Fig. 12. The main focus of this benchmark design is to decrease the speed reducer total weight. Seven decision variables called, module of teeth (m), number of teeth on pinion (z), diameter of shaft1 (d_1), face width (b), diameter of shaft 2 (d_2), length of shaft one between bearing (l_1), and length of shaft two between bearing (l_2) were employed in speed reducer design problems.

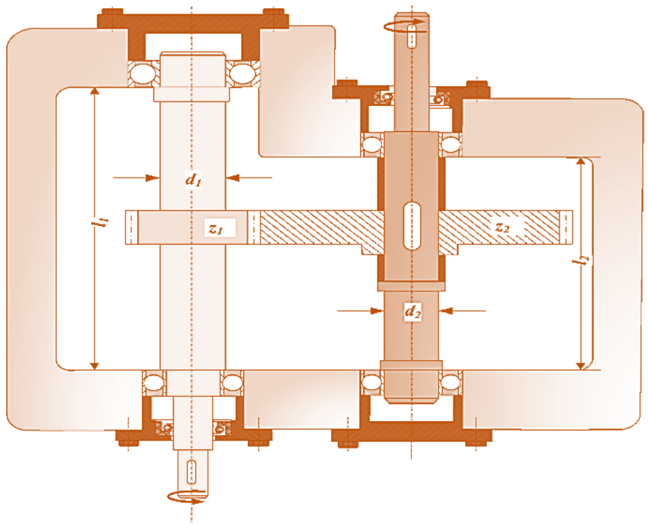


Fig. 12. Speed reducer design.

The optimization model of this problem is given as follows:

$$Minimize\ f(b.m.z.l_1.l_2.d_1.d_2) = 0.7854bm^2(3.3333z^2 + 14.9334z - 43.0934) - 1.508b(d_1^2 + d_2^2) + 7.4777(d_1^3 + d_2^3) + 0.7854(l_1d_1^2 + l_2d_2^2) \quad (19)$$

Subject to:

$$g_1 = \frac{27}{bm^2z} - 1 \leq 0$$

$$g_2 = \frac{397.5}{bm^2z^2} - 1 \leq 0$$

$$g_3 = \frac{1.93l_1^3}{mzd_1^4} - 1 \leq 0$$

$$g_4 = \frac{1.93l_2^3}{mzd_2^4} - 1 \leq 0$$

$$g_5 = \frac{\sqrt{(\frac{745l_1}{mz})^2 + 16.9 \times 10^6}}{110d_1^3} - 1 \leq 0$$

$$g_6 = \frac{\sqrt{\left(\frac{745l_2}{mz}\right)^2 + 157.5 \times 10^6}}{85d_2^3} - 1 \leq 0$$

$$g_7 = \frac{mz}{40} - 1 \leq 0$$

$$g_8 = \frac{5m}{b} - 1 \leq 0$$

$$g_9 = \frac{b}{12m} - 1 \leq 0$$

$$g_{10} = \frac{1.5d_1 + 1.9}{l_1} - 1 \leq 0$$

$$g_{11} = \frac{1.1d_2 + 1.9}{l_2} - 1 \leq 0$$

where 2.6 ≤y_1 ≤3.6, 0.7 ≤y_2 ≤0.8, 17 ≤y_3 ≤28, 7.3 ≤y_4 ≤8.3, 7.8 ≤y_5 ≤8.3, 2.9 ≤y_6 ≤3.9, 5.0 ≤y_7 ≤5.5. Many researchers [88–91] have proposed solutions to speed reducer design problem, presented in Table 18. Their statistical evaluations of these methods are indicated in Table 19; the answer given by the authors is infeasible as they violate the g_6 constraints.

Table 18
Comparison of the best solution for speed reducer problem**.**

	Kuang et al. [77]	**Ray and Saini [71]**	**Akhtar et al. [78]**	**Ray and Liew [70]**	**Raj et al. [69]**	**Montes et al. [80]**	**Cagnina et al. [79]**	**Gandomi et al. [73]**	**present study**
y_1	3.6	3.51419	3.50612	3.5000068	3.500071	3.50001	3.5	3.5015	3.4997
y_2	0.7	0.70001	0.70001	0.7	0.7	0.7	0.7	0.7	0.6999
y_3	17	17	17	17	17	17	17	17	16.999
y_4	7.3	7.49734	7.54913	7.3276021	7.3	7.3	7.3	7.605	7.3004
y_5	7.8	7.8346	7.85933	7.7153218	7.820728	7.8	7.8	7.8181	7.7994
y_6	3.4	2.9018	3.36558	3.350267	2.900173	3.350214	3.350214	3.352	2.8997
y_7	5	5.0022	5.28977	5.2866545	5.000005	5.286683	5.286683	5.2875	5.2867
g_1	-0.01	-0.0777	-0.0755	-0.0739171	-0.073934	-0.073915	-0.073915	-0.0743	-0.074
g_2	-0.22	-0.2012	-0.1994	-0.1980001	-0.198015	-0.197998	-0.197998	-0.1983	-0.1981
g_3	-0.528	-0.036	-0.4562	-0.9999967	-0.999994	-0.499172	-0.499172	-0.4349	-0.1009
g_4	-0.877	-0.8754	-0.8994	-0.9999995	-0.999999	-0.901471	-0.901471	-0.9008	-0.9015
g_5	-0.043	-0.4857	-0.0132	-0.6667294	-0.48645	0	0	-0.0011	-0.4862
g_6	0.1821[a]	0.1805[a]	-0.0017	-1.95E-08	0.1820623[a]	-5.00E-16	-5.00E-16	-0.0004	-0.0002
g_7	-0.703	-0.7025	-0.7025	-0.7024999	-0.7025	-0.7025	-0.7025	-0.7025	-0.7025
g_8	-0.028	-0.004	-0.0017	-0.0000019	-2.03E-05	-1.00E-16	-1.00E-16	-0.0004	-0.0001
g_9	-0.571	-0.5816	-0.5826	-0.5833325	-0.583325	-0.583333	-0.583333	-0.5832	-0.5833
g_{10}	-0.041	-0.166	-0.0796	-0.0548885	-0.1438	-0.051325	-5.13E-02	-0.089	-0.1461
g_{11}	-0.051	-0.0552	-0.0179	-2.33E-07	-0.053796	-0.010852	-0.010852	-0.013	-0.0108
$f(y)$	2876.1	2732.9	3008.08	2994.7442	2724.055	2996.3482	2996.3482	3001	2895.8317

[a] violate constraint;

Table 19
Statistical data for speed reducer design problem.

Algorithm	Best	Median	Mean	Worst	Std.
Kuang et al. [30]	2876.117623	NA [a]	NA	NA	NA
Ray and Saini	2732.9006	NA	2741.5642	2757.8581	NA
Akhtar et al. [2]	3008.08	NA	3012.12	3028	NA
Montes et al. [35]	3025.005	NA	3088.7778	3078.5918	NA
Ray and Liew [42]	2994.744241	3001.758	3001.758226	3009.964736	4.009142
Montes et al. [34]	2996.356689	NA	2996.36722	NA	8.20E-03
Cagnina et al. [5]	2994.471066	NA	2996.3482	NA	0
Zhang et al. [50]	3000.981	2994.471	2994.3482	2994.471066	3.58E-12
Gandomi et al. [50]	2894.73832	NA	3007.1997	NA	4.9634
GSA-GA [75]	2894.73832	2894.971	2894.71248	2895.03219	4.96E-04
GWOEHO	2895.831781	NA	2903.158785	2911.287901	3.98

[a] not available

Note that in evaluating the algorithm, 30 algorithm populations have been selected by example. The number of tribes in the algorithm is considered to be five tribes. The number of loops of the algorithm is equal to 500 cycles.

The results and outputs of the program after twenty runs from the program are presented in the following tables. A comparable penalty function for GWOEHO was utilized for performing a fair comparison [59]. Table 20 presents the results of the six algorithms GA, DE, PSO, EHO, GWO, GWOEHO and are compared.

Table 20
Comparison of outputs for speed reducer design.

	GA	DE	PSO	EHO	GWO	GWOEHO
Ave	2895.42	2895.33	2932.86	3364.06	2902.82	2903.16
Std.	0.07	0.00	15.88	153.15	3.19	3.98
Best	2895.34	2895.33	2895.33	3020.77	2897.83	2895.83
Worst	2895.62	2895.33	2955.62	3683.38	2910.67	2911.29
Time	2.44	1.73	0.41	3.14	0.53	3.63
Rank	2	1	1	5	4	3

Fig. 13 presents the typical convergence history of the six algorithms GA, DE, PSO, EHO, GWO, GWOEHO.

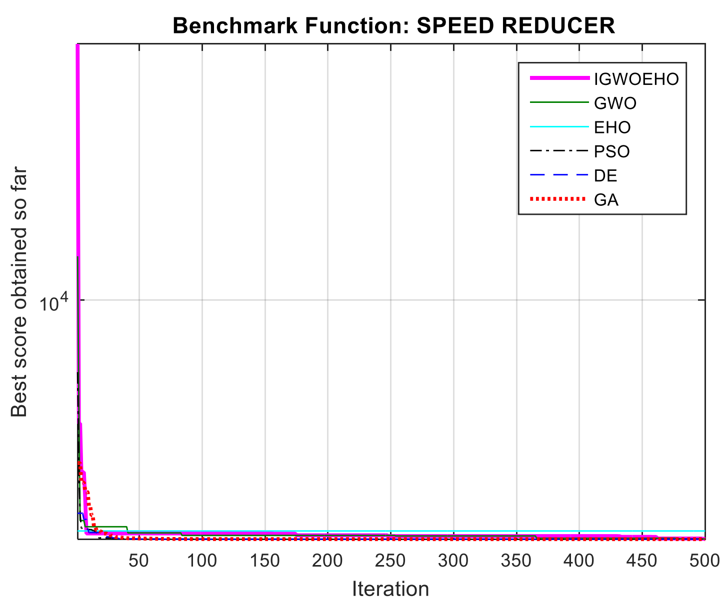


Fig. 13. Typical convergence history.

7.6. Tabular column design

In tabular column design problems, the target is to develop an identical column of a tabular section, with length (L) 250 cm, at lowest cost, that comprises construction and material cost, to tolerate a compressive load P = 2500 kgf presented in Fig. 14 [92]. The mean diameter (d) of the column is limited among 2 and 14 cm while thickness (t) restricted between 0.2 - 0.8 cm. The column is constructed of a material with a yield stress (σ_y = 500 kgf/cm2), a modulus of elasticity (E = 0.85 ×10 6 kgf/cm2), and a density (ρ= 0.0025 kgf/cm2). Based on these specifications, the optimization approach is formulated as

$$\text{Minimize} \quad f(Y) = 9.82dt + 2d \tag{20}$$
$$g_1 = \frac{p}{\pi dt\sigma_y} - 1 \leq 0$$
$$g_2 = \frac{8pl^2}{\pi^3 Edt(d^2 + t^2)} - 1 \leq 0$$
$$g_3 = \frac{2}{d} - 1 \leq 0$$
$$g_4 = \frac{d}{14} - 1 \leq 0$$
$$g_5 = \frac{0.2}{t} - 1 \leq 0$$
$$g_6 = \frac{t}{0.8} - 1 \leq 0$$

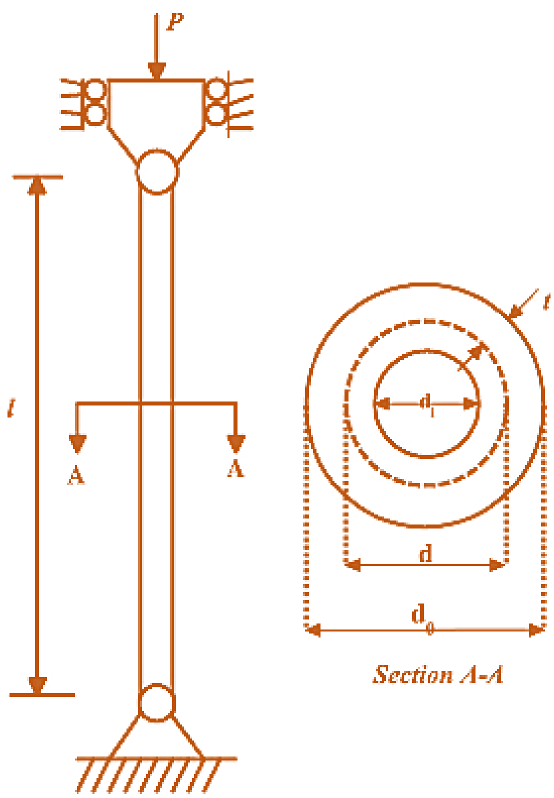


Fig. 14. Tabular column design.

Previous studies [17,92,93] dealt with tabular column design problems and solved them. Among these researches, the best solutions were proposed by Hsu and Liu [93], Rao [92], but their reported solutions are infeasible. However, by utilizing the proposed method in the current work

(GSA-GA algorithm) to solve tabular column design problems, the best solution to it can be achieved. Table 21 indicates the results of existing solutions and our proposed method. According to this table, the reported solution (GSA-GA algorithm) is much better than the current outputs. In addition, to present their constancy, the statistical analysis outcomes are indicated in Table 22, showing that the variation in the optimal outcomes is reasonably low in comparison with other methods [17,82,92,93].

Table 21
Comparison of the best solution for the tabular column design problem.

	Hsu and Liu [25]	Rao [40]	Gandomi et al. [16]	GSA-GA [75]	GWOEHO
d	5.4507	5.44	5.45139	5.45115623	5.451156
t	0.292	0.293	0.29196	0.29196548	0.291965
g_2	0.1317a	0.0026[a]	-0.1095	-7.50E-09	-7.50E-09
g_3	-0.6331	-0.8571	-0.6331	-0.633105	-0.63311
g_4	-0.6107	0	-0.6106	-0.610631	-0.61063
g_5	-0.3151	-0.75	-0.315	-0.314987	-0.31499
g_6	-0.635	0	-0.6351	-0.635043	-0.63504
f(y)	25.5316	26.5323	26.5321	26.531328	26.53133

[a] Violate constraint

Table 22
Statistical data for the tubular column problem.

Algorithms	Best	Median	Mean	Worst	Std.
Gandomi et al.	26.53217	NA [a]	26.53504	26.53972	0.00193
GSA-GA	26.531328	26.53133	26.531332	26.55315	3.94E-04
GWOEHO	26.53201094	NA	26.536874	26.54626	3.52E-03

[a] not available

Note that in evaluating the proposed algorithm's performance, the population size is set to 30, The number of tribes is set to 5, and the maximum number of iterations is equal to 500.
Results of the optimization algorithm after twenty independent runs are displayed in the following tables. Table 23 shows the results of six algorithms, GA, DE, PSO, EHO, GWO, GWOEHO, using the same penalty function.

Table 23
Comparison of results for tabular column design.

	GA	DE	PSO	EHO	GWO	GWOEHO
Ave	26.5399	26.5313	26.5313	27.4663	26.5377	26.5369
Std.	0.0114	0.0000	0.0000	0.4606	0.0033	0.0035
Best	26.5313	26.5313	26.5313	26.5914	26.5328	26.5320
Worst	26.5748	26.5313	26.5313	28.2970	26.5436	26.5463
Time	2.47	1.72	0.23	3.15	0.24	2.72
Rank	1	1	1	4	3	2

Fig. 15 presents the typical convergence history of the six algorithms GA, DE, PSO, EHO, GWO, GWOEHO.

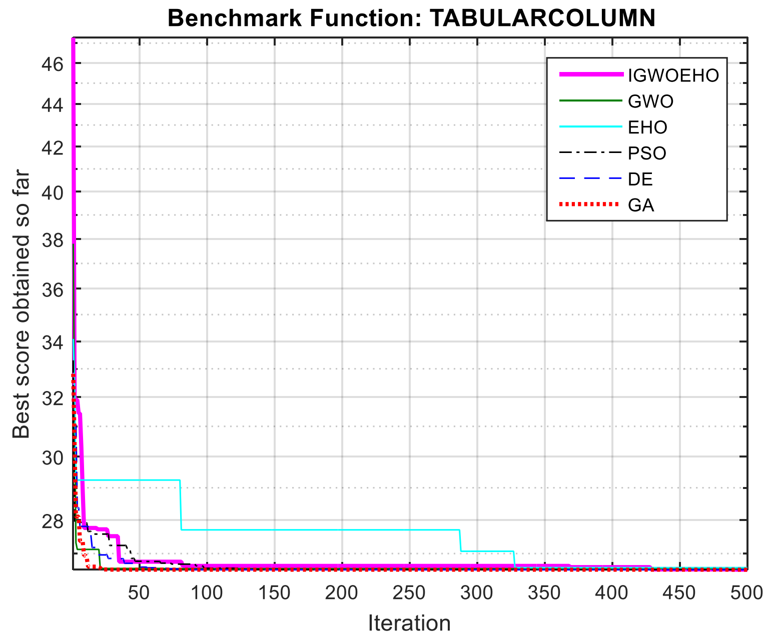


Fig. 15. Typical convergence history.

8. Conclusion

A new hybrid algorithm named GWOEHO is introduced in the present study by combining GWO and EHO algorithms' features and adding a new separating operator. The main idea is to integrate the strength of GWO in exploitation and the ability of EHO in exploration and avoid getting trapped in local optima. The convergence speed and accuracy of the proposed algorithm are also improved by embedding a new separating operator. Twenty-three benchmark mathematical functions and six constrained engineering problems are used to validate the proposed GWOEHO compared to the original GWO and EHO and some other well-known algorithms. The results show that GWOEHO outperforms both GWO and EHO algorithms in most problems. The GWOEHO algorithm has gained the first rank in fifteen benchmark mathematical functions and achieved the second best rank in 5 remained examples. Results obtained from Wilcoxon's rank-sum test as a nonparametric statistical test confirmed that the suggested GWOEHO overcomes the other algorithms, significantly.

The small standard deviation of the best solutions in most examples shows that the proposed GWOEHO algorithm exhibits robust performance in different independent runs of the same problem. This feature is especially important in complex and large-scale engineering problems. The application of the proposed algorithm in 6 different engineering problems and the comparison of the results with several well-known algorithms and the results from the technical literature show that GWOEHO can provide comparable results in this type of problem.

Funding

This work is supported by the Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (Grant 21NANO-B158359-02).

Conflicts of interest

The authors declare no conflict of interest.

Authors contribution statement

ZH, HV: Conceptualization; ZH, HV: Data curation; ZH: Formal analysis; ZH: Investigation; ZH, HV: Methodology; HV: Project administration; ZH: Resources; ZH, HV: Software; ZH, HV, MR, JJ: Supervision; HV: Validation; ZH, HV: Visualization; ZH, HV: Roles/Writing – original draft; MR, JJ: Writing – review & editing.

References

[1] Varaee H, Shishegaran A, Ghasemi MR. The life-cycle cost analysis based on probabilistic optimization using a novel algorithm. J Build Eng 2021;43:103032. https://doi.org/10.1016/j.jobe.2021.103032.

[2] Varaee H, Safaeian Hamzehkolaei N, Safari M. A hybrid generalized reduced gradient-based particle swarm optimizer for constrained engineering optimization problems. J Soft Comput Civ Eng 2021;5:86–119. https://doi.org/10.22115/SCCE.2021.282360.1304.

[3] Shishegaran A, Karami B, Safari Danalou E, Varaee H, Rabczuk T. Computational predictions for predicting the performance of steel 1 panel shear wall under explosive loads. Eng Comput (Swansea, Wales) 2021;ahead-of-p. https://doi.org/10.1108/EC-09-2020-0492.

[4] Shishegaran A, Varaee H, Rabczuk T, Shishegaran G. High correlated variables creator machine: Prediction of the compressive strength of concrete. Comput Struct 2021;247:106479. https://doi.org/10.1016/j.compstruc.2021.106479.

[5] Wang W, Rivard H, Zmeureanu R. Floor shape optimization for green building design. Adv Eng Informatics 2006;20:363–78.

[6] Varaee H, Ahmadi-Nedushan B. Minimum cost design of concrete slabs using particle swarm optimization with time varying acceleration coefficients. World Appl Sci J 2011;13:2484–94.

[7] Ghasemi MR, Varaee H. Damping vibration-based IGMM optimization algorithm: fast and significant. Soft Comput 2019;23:451–81. https://doi.org/10.1007/s00500-017-2804-3.

[8] Ghasemi MR, Varaee H. A fast multi-objective optimization using an efficient ideal gas molecular movement algorithm. Eng Comput 2017;33:477–96. https://doi.org/10.1007/s00366-016-0485-7.

[9] Ghasemi MR, Ghiasi R, Varaee H. Probability-Based Damage Detection of Structures Using Surrogate Model and Enhanced Ideal Gas Molecular Movement Algorithm. Adv Struct Multidiscip Optim 2018;11:1657–74. https://doi.org/10.1007/978-3-319-67988-4_124.

[10] Shabakhty N, Enferadi MH, Ghasemi MR, Varaee H. Application of Shape Memory Alloy Tuned Mass Damper in Vibration Control of Jacket type Offshore Structures. Iran J Mar Sci Technol 2020;7:64–75.

[11] Hwang S-F, He R-S. A hybrid real-parameter genetic algorithm for function optimization. Adv Eng Informatics 2006;20:7–21.

[12] Yang X-S, Gandomi AH, Talatahari S, Alavi AH. Metaheuristics in water, geotechnical and transport engineering. Newnes; 2012.

[13] Gandomi AH, Alavi AH. Multi-stage genetic programming: a new strategy to nonlinear system modeling. Inf Sci (Ny) 2011;181:5227–39.

[14] Ghasemi MR, Varaee H. Enhanced IGMM optimization algorithm based on vibration for numerical and engineering problems. Eng Comput 2018;34:91–116. https://doi.org/10.1007/s00366-017-0523-0.

[15] Goldberg D, Holland J. Genetic Algorithms and Machine Learning. Mach Learn 1988;3:95–9. https://doi.org/10.1023/A:1022602019183.

[16] Miarnaeimi F, Rashki M. Flying Squirrel Optimizer (FSO): A novel SI-based optimization algorithm for Flying Squirrel Optimizer (FSO): A novel SI-based optimization algorithm for engineering problems 2018.

[17] Hossein A, Yang GX, Gandomi AH, Yang XS, Alavi AH. Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems. Eng Comput 2013;29:17–35. https://doi.org/10.1007/s00366-011-0241-y.

[18] Chakraborty UK. Advances in differential evolution. vol. 143. Springer; 2008.

[19] Karaboga D, Basturk B. Artificial bee colony (ABC) optimization algorithm for solving constrained optimization. Probl. LNCS Adv. Soft Comput. Found. Fuzzy Log. Soft Comput. Springer-Verlag, IFSA (2007, Citeseer; 2007, p. 789–98.

[20] Mirjalili S, Mirjalili SM, Lewis A. Grey Wolf Optimizer. Adv Eng Softw 2014;69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007.

[21] Gandomi AH, Yang X-S, Alavi AH, Talatahari S. Bat algorithm for constrained optimization tasks. Neural Comput Appl 2013;22:1239–55.

[22] Wang G-GG-G, Deb S, Coelho LDS. Elephant Herding Optimization. 3rd Int. Symp. Comput. Bus. Intell., IEEE; 2015, p. 1–5. https://doi.org/10.1109/ISCBI.2015.8.

[23] Wang G. Moth search algorithm : a bio-inspired metaheuristic algorithm for global optimization problems. Memetic Comput 2016. https://doi.org/10.1007/s12293-016-0212-3.

[24] Varaee H, Ghasemi MR. Engineering optimization based on ideal gas molecular movement algorithm. Eng Comput 2017;33:71–93. https://doi.org/10.1007/s00366-016-0457-y.

[25] Eberhart R, Kennedy J. A new optimizer using particle swarm theory. MHS'95. Proc. Sixth Int. Symp. Micro Mach. Hum. Sci., Ieee; 1995, p. 39–43. https://doi.org/10.1109/mhs.1995.494215.

[26] Yang XS. Harmony search as a metaheuristic algorithm. Stud Comput Intell 2009;191:1–14. https://doi.org/10.1007/978-3-642-00185-7_1.

[27] Dorigo M, Caro G Di. Ant colony optimization: a new meta-heuristic. Proc 1999 Congr Evol Comput (Cat No 99TH8406) 1999;2. https://doi.org/10.1109/CEC.1999.782657.

[28] Sedki A, Ouazar D. Hybrid particle swarm optimization and differential evolution for optimal design of water distribution systems. Adv Eng Informatics 2012;26:582–91.

[29] Jiao J, Xu M. A Novel Grey Wolf Optimizer Algorithm With Refraction Learning. IEEE Access 2019;7:57805–19. https://doi.org/10.1109/ACCESS.2019.2910813.

[30] Faris H, Aljarah I, Al-Betar MA, Mirjalili S. Grey wolf optimizer: a review of recent variants and applications. Neural Comput Appl 2018;30:413–35.

[31] Medjahed SA, Saadi TA, Benyettou A, Ouali M. Gray wolf optimizer for hyperspectral band selection. Appl Soft Comput 2016;40:178–86.

[32] Emary E, Zawbaa HM, Hassanien AE. Binary grey wolf optimization approaches for feature selection. Neurocomputing 2016;172:371–81.

[33] Long W, Jiao J, Liang X, Tang M. Inspired grey wolf optimizer for solving large-scale function optimization problems. Appl Math Model 2018;60:112–26. https://doi.org/10.1016/j.apm.2018.03.005.

[34] Saxena A, Soni BP, Kumar R, Gupta V. Intelligent Grey Wolf Optimizer–Development and application for strategic bidding in uniform price spot energy market. Appl Soft Comput 2018;69:1–13.

[35] Long W, Jiao J, Liang X, Tang M. An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization. Eng Appl Artif Intell 2018;68:63–80.

[36] Gupta S, Deep K. A novel random walk grey wolf optimizer. Swarm Evol Comput 2019;44:101–12.

[37] Mittal N, Singh U, Sohi BS. Modified grey wolf optimizer for global engineering optimization. Appl Comput Intell Soft Comput 2016;2016.

[38] Alomoush AA, Alsewari AA, Alamri HS, Aloufi K, Zamli KZ. Hybrid harmony search algorithm with grey wolf optimizer and modified opposition-based learning. IEEE Access 2019;7:68764–85.

[39] Sanjay R, Jayabarathi T, Raghunathan T, Ramesh V, Mithulananthan N. Optimal allocation of distributed generation using hybrid grey wolf optimizer. Ieee Access 2017;5:14807–18.

[40] Al-Tashi Q, Kadir SJA, Rais HM, Mirjalili S, Alhussian H. Binary optimization using hybrid grey wolf optimization for feature selection. IEEE Access 2019;7:39496–508.

[41] Zhu A, Xu C, Li Z, Wu J, Liu Z. Hybridizing grey wolf optimization with differential evolution for global optimization and test scheduling for 3D stacked SoC. J Syst Eng Electron 2015;26:317–28.

[42] Jitkongchuen D. A hybrid differential evolution with grey wolf optimizer for continuous global optimization. 2015 7th Int. Conf. Inf. Technol. Electr. Eng., IEEE; 2015, p. 51–4.

[43] Tawhid MA, Ali AF. A hybrid grey wolf optimizer and genetic algorithm for minimizing potential energy function. Memetic Comput 2017;9:347–59.

[44] Gaidhane PJ, Nigam MJ. A hybrid grey wolf optimizer and artificial bee colony algorithm for enhancing the performance of complex systems. J Comput Sci 2018;27:284–302.

[45] Arora S, Singh H, Sharma M, Sharma S, Anand P. A new hybrid algorithm based on grey wolf optimization and crow search algorithm for unconstrained function optimization and feature selection. Ieee Access 2019;7:26343–61.

[46] Tuba E, Dolicanin-Djekic D, Jovanovic R, Simian D, Tuba M. Combined elephant herding optimization algorithm with K-means for data clustering. Inf. Commun. Technol. Intell. Syst., Springer; 2019, p. 665–73.

[47] Tuba E, Capor-Hrosik R, Alihodzic A, Jovanovic R, Tuba M. Chaotic elephant herding optimization algorithm. 2018 IEEE 16th World Symp. Appl. Mach. Intell. Informatics, IEEE; 2018, p. 213–6.

[48] ElShaarawy IA, Houssein EH, Ismail FH, Hassanien AE. An exploration-enhanced elephant herding optimization. Eng Comput 2019.

[49] Li J, Lei H, Alavi AH, Wang G-G. Elephant herding optimization: variants, hybrids, and applications. Mathematics 2020;8:1415.

[50] Talbi E-G. Metaheuristics: from design to implementation. vol. 74. John Wiley & Sons; 2009.

[51] Mafarja MM, Mirjalili S. Hybrid whale optimization algorithm with simulated annealing for feature selection. Neurocomputing 2017;260:302–12.

[52] Li J, Guo L, Li Y, Liu C. Enhancing Elephant Herding Optimization with Novel Individual Updating Strategies for Large-Scale Optimization Problems 2019.

[53] Yao X, Liu Y, Lin G. Evolutionary programming made faster. IEEE Trans Evol Comput 1999;3:82–102.

[54] Ghasemi MR, Varaee H. Modified Ideal Gas Molecular Movement Algorithm Based on Quantum Behavior. In: Schumacher A, Vietor T, Fiebig S, Bletzinger K-U, Maute K, editors. Adv. Struct. Multidiscip. Optim., Cham: Springer International Publishing; 2018, p. 1997–2010. https://doi.org/10.1007/978-3-319-67988-4_148.

[55] Suganthan PN, Hansen N, Liang JJ, Deb K, Chen Y-P, Auger A, et al. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. KanGAL Rep 2005;2005005:2005.

[56] Xie SQ, Gan J, Wang GG, Vn C. Optimal process planning for compound laser cutting and punch using Genetic Algorithms. Int J Mechatronics Manuf Syst 2009;2:20–38. https://doi.org/10.1504/IJMMS.2009.024346.

[57] Storn R, Price K. Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. vol. 3. ICSI Berkeley; 1995.

[58] Ren Z, Fang X, Wang S, Qiu J, Zhu JG, Guo Y, et al. Design optimization of an interior-type permanent magnet BLDC motor using PSO and improved MEC. 2007 Int. Conf. Electr. Mach. Syst., IEEE; 2007, p. 1350–3.

[59] Derrac J, García S, Molina D, Herrera F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 2011;1:3–18. https://doi.org/10.1016/j.swevo.2011.02.002.

[60] Arora J. Introduction to optimum design. Academic Press; 2004.

[61] Belegundu AD, Arora JS. A study of mathematical programmingmethods for structural optimization. Part II: Numerical results. Int J Numer Methods Eng 1985;21:1601–23. https://doi.org/10.1002/nme.1620210905.

[62] Coello CAC, Montes EM. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. Adv Eng Informatics 2002;16:193–203.

[63] He Q, Wang L. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng Appl Artif Intell 2007;20:89–99.

[64] Mahdavi M, Fesanghary M, Damangir E. An improved harmony search algorithm for solving optimization problems. Appl Math Comput 2007;188:1567–79. https://doi.org/10.1016/j.amc.2006.11.033.

[65] Huang F, Wang L, He Q. An effective co-evolutionary differential evolution for constrained optimization. Appl Math Comput 2007;186:340–56.

[66] Coello CAC. Use of a self-adaptive penalty approach for engineering optimization problems. Comput Ind 2000;41:113–27.

[67] Mezura-Montes E, Coello CAC. An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. Int J Gen Syst 2008;37:443–73.

[68] Yang X, Press L. Nature-Inspired Metaheuristic Algorithms Second Edition. n.d.

[69] Coello CAC. Constraint-handling using an evolutionary multiobjective optimization technique. Civ Eng Syst 2000;17:319–46.

[70] Deb K. Optimal design of a welded beam via genetic algorithms. AIAA J 1991;29:2013–5. https://doi.org/10.2514/3.10834.

[71] Deb K. An efficient constraint handling method for genetic algorithms. Comput Methods Appl Mech Eng 2000;186:311–38. https://doi.org/10.1016/S0045-7825(99)00389-8.

[72] Lee KS, Geem ZW. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput Methods Appl Mech Eng 2005;194:3902–33. https://doi.org/10.1016/j.cma.2004.09.007.

[73] Ragsdell KM, Phillips DT. Optimal design of a class of welded structures using geometric programming. J Eng Ind 1976;98:1021–5.

[74] Deb K. Geneas: A robust optimal design technique for mechanical component design. Evol. algorithms Eng. Appl., Springer; 1997, p. 497–514.

[75] Kaveh A, Talatahari S. An improved ant colony optimization for constrained engineering design problems. Eng Comput 2010;27:155–82. https://doi.org/10.1108/02644401011008577.

[76] Sandgren E. Nonlinear integer and discrete programming in mechanical design. Proc. ASME Des. Technol. Conf., 1988, p. 95–105.

[77] Kannan BK, Kramer SN. An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. J Mech Des 1994;116:405–11.

[78] Ray T, Liew KM. Society and civilization: An optimization algorithm based on the simulation of social behavior. Evol Comput IEEE Trans 2003;7:386–96. https://doi.org/10.1109/TEVC.2003.814902.

[79] Ray T, Saini P. Engineering design optimization using a swarm with an intelligent information sharing among individuals. Eng Optim 2001;33:735–48. https://doi.org/10.1080/03052150108940941.

[80] Zhang M, Luo W, Wang X. Differential evolution with dynamic stochastic selection for constrained optimization. Inf Sci (Ny) 2008;178:3043–74.

[81] Raj KH, Sharma RS, Mishra GS, Dua A, Patvardhan C. An evolutionary computational technique for constrained optimisation in engineering design. J Inst Eng Mech Eng Div 2005;86:121–8.

[82] Garg H. A hybrid GSA-GA algorithm for constrained optimization problems. Inf Sci (Ny) 2019;478:499–523. https://doi.org/10.1016/j.ins.2018.11.041.

[83] Tsai JFA. Global optimization of nonlinear fractional programming problems in engineering design. Eng Optim 2005;37:399–409. https://doi.org/10.1080/03052150500066737.

[84] Serrano-rubio JP, Hernández-aguirre A, Herrera-guzmán R. An evolutionary algorithm using spherical inversions. Soft Comput 2017. https://doi.org/10.1007/s00500-016-2461-y.

[85] Tsai J-F, Li H-L, Hu N-Z. Global optimization for signomial discrete programming problems in engineering design. Eng Optim 2002;34:613–22.

[86] Zhang L, Tang Y, Hua C, Guan X. A new particle swarm optimization algorithm with adaptive inertia weight based on Bayesian techniques. Appl Soft Comput 2015;28:138–49. https://doi.org/10.1016/j.asoc.2014.11.018.

[87] Ku KJ, Rao SS, Chen L. Taguchi-aided search method for design optimization of engineering systems. Eng Optim 1998;30:1–23. https://doi.org/10.1080/03052159808941235.

[88] Akhtar S, Tai K, Ray T. A socio-behavioural simulation model for engineering design optimization. Eng Optim 2002;34:341–54. https://doi.org/10.1080/03052150212723.

[89] Cagnina LC, Esquivel SC, Nacional U, Luis DS, Luis S, Coello CAC. Solving engineering optimization problems with the simple constrained particle swarm optimizer. Informatica 2008;32:319–26.

[90] Mezura-Montes E, Coello CACC, Velazquez-Reyes J, Munoz-Davila L. Multiple trial vectors in differential evolution for engineering design. Eng Optim 2007;39:567–89. https://doi.org/10.1080/03052150701364022.

[91] Mezura-Montes E, Coello CAC, Landa-Becerra R. Engineering optimization using simple evolutionary algorithm. Tools with Artif. Intell. 2003. Proceedings. 15th IEEE Int. Conf., 2003, p. 149–56. https://doi.org/10.1109/TAI.2003.1250183.

[92] Rao SS. Engineering Optimization: Theory and Practice. John Wiley & Sons; 2009. https://doi.org/10.1002/9780470549124.

[93] Hsu Y-LL, Liu T-CC. Developing a fuzzy proportional-derivative controller optimization engine for engineering design optimization problems. Eng Optim 2007;39:679–700. https://doi.org/10.1080/03052150701252664.