Contents lists available at ScienceDirect

Journal of Computational Design and Engineering

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

Chaotic grey wolf optimization algorithm for constrained optimization problems

Mehak Kohli, Sankalap Arora [*]

DAV University, Jalandhar, India

ARTICLE INFO

Article history:
Received 1 October 2016
Received in revised form 17 November 2016
Accepted 15 February 2017
Available online 7 March 2017

Keywords:
Chaotic grey wolf optimization
Firefly algorithm
Flower pollination algorithm
Particle swarm optimization algorithm

ABSTRACT

The Grey Wolf Optimizer (GWO) algorithm is a novel meta-heuristic, inspired from the social hunting behavior of grey wolves. This paper introduces the chaos theory into the GWO algorithm with the aim of accelerating its global convergence speed. Firstly, detailed studies are carried out on thirteen standard constrained benchmark problems with ten different chaotic maps to find out the most efficient one. Then, the chaotic GWO is compared with the traditional GWO and some other popular meta-heuristics viz. Firefly Algorithm, Flower Pollination Algorithm and Particle Swarm Optimization algorithm. The performance of the CGWO algorithm is also validated using five constrained engineering design problems. The results showed that with an appropriate chaotic map, CGWO can clearly outperform standard GWO, with very good performance in comparison with other algorithms and in application to constrained optimization problems.

1. Introduction

Constraints represent a feasible region which is nonempty and is filled with some restrictions or constraints to be followed by the solutions to solve a specific optimization problem (Karaboga and Akay, 2011). In general terms, constraints can be classified into equality constraints and inequality constraints which are represented in the form of mathematical equality and inequality equations respectively. Both types of constraints need to be satisfied by the problem's decision variables. Earlier, some deterministic methods like feasible direction approach and generalized gradient descent method were developed for solving constraint problems (Herskovits, 1986). However, due to their limited applicability and complexity of constraints, these were not effective for real world applications like structural optimization problems, economical optimization, location problems and engineering design problems like spring design, welded beam design, truss design, speed reducer design which involve many difficult equality and inequality constraints to be satisfied (Cagnina et al., 2008; Coello, 2000; Gandomi et al., 2013; Gao et al., 2010; Lee and Geem, 2004; Parsopoulos and Vrahatis, 2002). More and more meta-heuristic algorithms have been proposed to tackle these tough constrained optimization problems. These algorithms aim for tolerable velocity of convergence, a better precision, robustness, and performance. Some of the recent meta-heuristic algorithms proposed are Firefly Algorithm (FA) which is inspired by the flashing and attraction behavior of fireflies (Arora and Singh, 2013; Wang et al., 2014), Flower Pollination Algorithm (FPA) which is based on the characteristics of flowering of plants (Yang, 2012), Particle Swarm Optimization (PSO) which is inspired by the swarm behavior such as fish and bird schooling in nature (Shi and Eberhart, 1998), Bird Swarm Algorithm (BSA) which is based on the unique social interactions of bird swarms (Meng et al., 2015), ebb-tide-fish-inspired (ETFI) algorithm which is a simulation of fascinating characteristic of fish's perception of flow, sound and vibrations of tides in water (Meng et al., 2016), Jaya algorithm in which the main concept is to move the solution found so far towards the best solution and away from the worst solution (Rao, 2016), Grey Wolf Optimization (GWO) algorithm which is based on the social hunting behavior of grey wolves, Animal Migration Optimization (AMO) algorithm whose optimization process is mainly divided into two process viz. migration process and updating process with respect to animals (Luo et al., 2016), Butterfly Optimization Algorithm (BOA) which is inspired by the food foraging behavior of butterflies (Arora and Singh, 2015), Brain Storm Optimization (BSO) algorithm which is based on the simulation of brain storming process in humans (Shi, 2015), Whale Optimization Algorithm (WOA) which is inspired from the social interaction of humpback whales (Mirjalili and Lewis, 2016), Crow Search Algorithm (CSA) which mimics the clever characteristic of crows (Askarzadeh, 2016). Such meta-heuristics are being used extensively to solve complex problems like optimal wind generator design problem (Gao et al., 2010),

Peer review under responsibility of Society for Computational Design and Engineering.

[*] Corresponding author.

E-mail addresses: mehakkohli2525@gmail.com (M. Kohli), sankalap.arora@gmail.com (S. Arora).

http://dx.doi.org/10.1016/j.jcde.2017.02.005
2288-4300/© 2017 Society for Computational Design and Engineering. Publishing Services by Elsevier.

formulation of soil classification (Alavi et al., 2010), prediction of ground soil parameters (Alavi and Gandomi, 2011). Some of the prominent metaheuristics of the literature which have already been used to tackle constrained problems are: Deb introduced a method to handle constraints using GA (Deb, 2000), Montes employed Differential Evolution (DE) Algorithm on constraint handling problems (Mezura-Montes and Coello, 2005), PSO was used by Cagnina to solve constrained optimization problems (Cagnina et al., 2008) and Karaboga used Artificial Bee Colony (ABC) algorithm to handle constraint mechanism (Karaboga and Basturk, 2007).

GWO algorithm in fact, is a new meta-heuristic algorithm, inspired by the leadership behavior and unique mechanism of hunting of grey wolves. This population based meta-heuristic has the ability to avoid local optima stagnation to some extent (Yang et al., 2012). It also has good convergence ability towards the optima. In general, GWO advances itself strongly to exploitation. However, it cannot always implement global search well. Thus, in some cases, GWO fails to find global optimal solution. The search strategy used in basic GWO is mainly based on random walks. Thus, it cannot always deal with the problem successfully.

With the development of the nonlinear dynamics, chaos theory has been widely used in several applications (Pecora and Carroll, 1990). In this context, one of the most famous applications is the introduction of chaos theory into the optimization methods (Yang et al., 2007). Up to now, the chaos theory has been successfully combined with several meta-heuristic optimization methods (Gandomi et al., 2013). Some major efforts in this area includes PSO (Gandomi et al., 2013), FA (Gandomi et al., 2013), BOA (Arora and Singh, 2017), GA (Han and Chang, 2013), hybridizing chaotic sequences with memetic differential evolution algorithm (Jia et al., 2011), imperialist competitive algorithm (Talatahari et al., 2012) and gravitational search algorithm (Han and Chang, 2012), Krill Herd (KH) algorithm (Wang et al., 2014), and Accelerated Particle Swarm Optimization (APSO) (Gandomi et al., 2013).

In the present study, chaotic GWO (CGWO) algorithm is presented for the purpose of accelerating the convergence of GWO. Various one-dimensional chaotic maps are employed in place of the critical parameters used in GWO. Moreover, in order to examine the efficiency of the proposed CGWO in the room of constraint handling mechanism, it has been applied on some constrained benchmark functions and various classical engineering design problems viz. spring design problem, gear train design problem, welded beam design problem, pressure vessel design problem and closed coil helical spring design problem. The results of proposed CGWO on all the constrained benchmark functions have been compared with those obtained by GWO (Mirjalili et al., 2014), Firefly Algorithm (FA) (Yang and algorithm, 2010), Flower Pollination Algorithm (FPA) (Yang, 2012) and Particle Swarm Optimization (PSO) (Kennedy, 2011). On the other hand, the simulation results of all the classical engineering design problems have been compared with other state-of-art meta-heuristics discussed in the respective section.

Organization of the remaining paper is as follows: Section 2: a brief introduction of GWO algorithm is given. Section 3: Description of the proposed CGWO algorithm is given in detail. Section 4: Validation of CGWO algorithm on thirteen constrained benchmark functions is performed. Section 5: Experimental study and discussion on results is done. Section 6: CGWO on various classical engineering design problems is described. Section 7: Conclusion of work along with its future scope is given.

2. Overview of grey wolf optimization algorithm

Initially, Grey Wolf Optimizer (GWO) was introduced by S. Mirjalili in year 2014 (Mirjalili et al., 2014). This algorithm is a simulation of unique hunting and searching prey characteristics of grey wolves. GWO has assumed the four level social hierarchy of grey wolves involving α at first, β at second, δ at third and ω wolves at last level. α wolves are the leader wolves managing and conducting the whole pack of grey wolves. It is also responsible for controlling the whole hunting process, taking all types of decisions like hunting, maintaining discipline, sleeping and waking time for whole pack. β wolf which is the best candidate to be the α, takes feedback from other wolves and give it to the α leader. The third level of grey wolves, i.e. δ wolves, dominate the wolves of forth and the last level called the ω wolves which are responsible for maintaining the safety and integrity in the wolf pack (Mirjalili et al., 2014).

The distances from α, β and δ wolves i.e. D_α, D_β and D_δ to each of the remaining wolf $(\vec{X})$ are calculated using Eq. (1) using which the effect of α, β and δ wolves on the prey viz. $\overrightarrow{X_1}$, $\overrightarrow{X_2}$ and $\overrightarrow{X_3}$ can be calculated as represented in Eq. (2).

$$\overrightarrow{D_\alpha} = \left|\overrightarrow{C_1} \cdot \overrightarrow{X_\alpha} - \vec{X}\right|, \quad \overrightarrow{D_\beta} = \left|\overrightarrow{C_2} \cdot \overrightarrow{X_\beta} - \vec{X}\right|, \quad \overrightarrow{D_\delta} = \left|\overrightarrow{C_3} \cdot \overrightarrow{X_\delta} - \vec{X}\right| \tag{1}$$

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_1} \cdot \overrightarrow{D_\alpha}, \quad \overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2} \cdot \overrightarrow{D_\beta}, \quad \overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_3} \cdot \overrightarrow{D_\delta} \tag{2}$$

$$\vec{A} = 2\vec{a} \cdot \vec{r_1} - \vec{a}, \quad \vec{C} = 2 \cdot \vec{r_2} \tag{3}$$

$$\vec{X}(t+1) = \left(\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}\right)/3 \tag{4}$$

The values of controlling parameters of the algorithm which are a, A and C are calculated using Eq. (3). Here, $\vec{r_1}$ and $\vec{r_2}$ are the random vectors in the range of $[0,1]$. These vectors make wolves able to reach at any point between the prey and the wolf. Vector $\vec{a}$ is involved in controlling activity of the GWO algorithm and used in calculating $\vec{A}$. The component values of $\vec{a}$ vector decreases linearly from 2 to 0 over the courses of iterations (Mirjalili et al., 2014). $\vec{C}$ helps in putting some extra weight on the prey to make it difficult for the wolves to find it. Thus at last, all other wolves update their positions $\vec{X}(t+1)$ using Eq. (4).

In spite of being new comer, GWO is being used in many real world applications such as a modified version of GWO algorithm was proposed and applied successfully for training q-gaussian radial basis functional link nets (Muangkote et al., 2014), A modified GWO algorithm named multi-verse optimizer (MVO) for solving various optimization problems was proposed (Mirjalili et al., 2016), The binary version of GWO algorithm was proposed to be used for feature selection which was one of the important and crucial modification of GWO algorithm (Emary et al., 2016), A multi-objective GWO was modeled to minimize the gases emission level of CO_2 by the capacitor in which 30-bus system was used for the evaluation of the proposed method (Mohamed et al., 2015), GWO algorithm was used to optimize the controlling parameters of DC motor (Madadi and Motlagh, 2014), The flowshop scheduling problem of stage 2 was solved along with the optimization of its release time by using GWO algorithm (Komaki and Kayvanfar, 2015).

3. Chaotic grey wolf optimization algorithm

In spite of having good convergence rate, GWO still cannot always perform that well in finding global optima which affect the convergence rate of the algorithm. So, to reduce this affect and improve its efficiency, CGWO algorithm is developed by introducing chaos in GWO algorithm itself. In general terms, chaos is a deterministic, random- like method found in non-linear, dynamical system, which is non-period, non-converging and bounded. Mathematically, chaos is randomness of a simple deterministic dynamical system and chaotic system may be considered as sources of randomness. In order to introduce chaos in optimization algorithms, different chaotic maps having different mathematical equations are used. Since last decade, chaotic maps have been widely appreciated in the field of optimization due to their dynamic behavior which help optimization algorithms in exploring the search space more dynamically and globally. At a recent time, in accordance with different human's realm a wide variety of chaotic maps designed by physicians, researchers and mathematicians are available in the optimization field (He et al., 2001). Out of all these available chaotic maps, bulk of them has been mostly applied to algorithms to apply it further on real world applications. Thus, as reviewed from the gathered literature ten most relevant uni-dimensional chaotic maps to tackle CGWO have been used in the present work (Gandomi and Yang, 2014) details of which are given in Table 1.

In these chaotic maps, any number in the range $[0,1]$ (or according to the range of chaotic map) can be chosen as the initial value. However, it should be noted that the initial value may have significant impacts on the fluctuation pattern of some of the chaotic maps. This set of chaotic maps has been chosen with different behaviors, while the initial value is 0.7 for all (Saremi et al., 2014). Chaotic maps affect the convergence rate of GWO algorithm positively as these maps induce chaos in the feasible region which is predictable only for very short initial time and is stochastic for longer period of time. Pseudo code of the proposed CGWO algorithm for solving optimization problems is portrayed in Fig. 2.

The optimization procedure of the proposed CGWO algorithm is also presented in the form of flow chart given in Fig. 1. In this, first step involves the stochastic initialization of population of grey wolves. Then, a chaotic map is chosen to be mapped with the algorithm along with the initialization of its first chaotic number and a variable (Gandomi and Yang, 2014). Sequentially, the parameters of the CGWO algorithm involved in conducting the exploration - exploitation mechanism viz. a, A and C are initialized which are same as in GWO. Fitness of all grey wolves initialized in the search space are evaluated using various standard benchmark functions and are sorted according to their fitness. The first wolf got after sorting is assumed to be α wolf and accordingly second and third wolves are assumed as β and δ wolf respectively. Sequentially, the fitter wolf will keep updating its position using Eq. (4) and may get the position of α wolf as optimal solution. The parameters' values are also updated along with the course of iterations using Eq. (3). At the end of the last iteration, fitness of α wolf will be considered as the most optimal solution of the problem found by the CGWO algorithm.

Table 1
Details of chaotic maps applied on CGWO.

S. no.	Map name	Map equation
1	Bernoulli map	$x_{k+1} = \left\{ \begin{array}{ll} \frac{x_k}{1-a} & 0 \leqslant x_k \leqslant a \\ \frac{x_k-(1-a)}{a} & (1-a) \leqslant x_k \leqslant 1 \end{array} \right\}$
2	Logistic map	$x_{k+1} = a.x_k(1-x_k)$
3	Chebyshev map	$x_{k+1} = \cos(a \cdot \cos^{-1} x_k)$
4	Circle map[a]	$x_{k+1} = x_k + b - \left(\frac{a}{2\pi}\right) \sin(2\pi x_k) mod(1)$
5	Cubic map	$x_{k+1} = \rho(1-x_k^2), x_k \in (0,1)$
6	Iterative chaotic map with infinite collapses (ICMIC) map	$x_{k+1} = abs\left(\sin\left(\frac{a}{x_k}\right)\right), \ a \in (0,1)$
7	Piecewise map	$x_{k+1} = \left\{ \begin{array}{ll} \frac{x_k}{a} & 0 \leqslant x_k \leqslant a \\ \frac{x_k-a}{0.5-a} & a \leqslant x_k \leqslant 0.5 \\ \frac{1-a-x_k}{0.5-a} & 0.5 \leqslant x_k \leqslant 1-a \\ \frac{1-x_k}{a} & 1-a \leqslant x_k \leqslant 1 \end{array} \right\}$
8	Singer map	$x_{k+1} = a(7.86x_k - 23.31x_k^2 + 28.75x_k^3 - 13.302875x_k^4)$
9	Sinusoidal map	$x_{k+1} = a.x_k^2 \sin(\pi x_k)$
10	Tent map	$x_{k+1} = \left\{ \begin{array}{ll} x_k/0.7 & x_k < 0.7 \\ 10/3(1-x_k) & x_k \geqslant 0.7 \end{array} \right\}$

[a] Using $a = 0.5$ and $b = 0.2$, it generates chaotic sequence in $(0,1)$, a = control parameter, x_k = chaotic number at iteration 'k'.

4. CGWO for constrained benchmark functions

All the constrained problems are formulated in the form of two functions i.e. objective function and constraint violation function (Powell, 1978). Objective function is the function whose main aim is to find the optimal solution say 'x' in the specified search space. It can be represented as Eq. (5).

$$minimize\ f(x), x = (x_1, x_2, x_3, \ldots, x_n) \in R^n \tag{5}$$

where n is the number of dimensions contained in a solution. $x \in F \in S$ where F is the feasible region in the search space S which defines a n-dimensional rectangle R (Karaboga and Basturk, 2007). This rectangle R has domains size in the form of lower bound (lb) and upper bound (ub) as represented in Eq. (6).

$$lb(i) \leqslant x(i) \leqslant ub(i), \quad 1 \leqslant i \leqslant n \tag{6}$$

and the number of constraints say '$m(m > 0)$' are defined in the F space is the form of Eq. (7).

$$\begin{array}{ll} g_j(x) \leqslant 0, & \text{for } j = 1, \ldots, q, \\ h_j(x) = 0, & \text{for } j = q+1, \ldots, m \end{array} \tag{7}$$

Here, $g_j(x)$ and $h_j(x)$ are called as inequality and equality constraints respectively. If any solution say 'x' satisfies the constraint g_k or h_k in F space, then g_k is considered to be an active constraint at x.

5. Experimental study and discussion

5.1. Parameter settings

Among all the complex methods to calculate the penalty of constraints like iterative method, methods based on feasibility of solutions, simple penalty function method is used in all the constrained optimization problems implemented and discussed in this paper (Joines and Houck, 1994). The population size of grey wolves is taken 30 and 100 iterations are performed for the results of all the constrained benchmark functions. 30 Monte Carlo runs are executed on each of the constrained benchmark functions. For effective validation of the proposed CGWO algorithm in case of constrained benchmark functions, it has been compared with some other optimization algorithms which are GWO (Mirjalili et al., 2014), FA (Yang and algorithm, 2010), FPA (Yang, 2012) and PSO (Kennedy, 2011). Additionally, parameter settings for all these algorithms need to be done for impartial comparison which is one of the difficult task to perform during the execution. The parameter settings done in this work is like for PSO, global learning (no local neighborhoods), an inertial constant = 0.3, a cognitive constant = 1 and a social constant for swarm interaction = 1 is used. For FPA, $\lambda = 1.5$ for Levy distribution function and proximity probability $p = 0.8$ is used. For FA, randomization parameter $\alpha = 0.6$, attractiveness $\beta_0 = 1$ and absorption coefficient $\gamma = 1.0$ is used. For GWO, two random vectors $\vec{r_1}$ and $\vec{r_2}$ are taken in the range of $(0,1)$ and the controlling parameter $\vec{a}$ has linearly

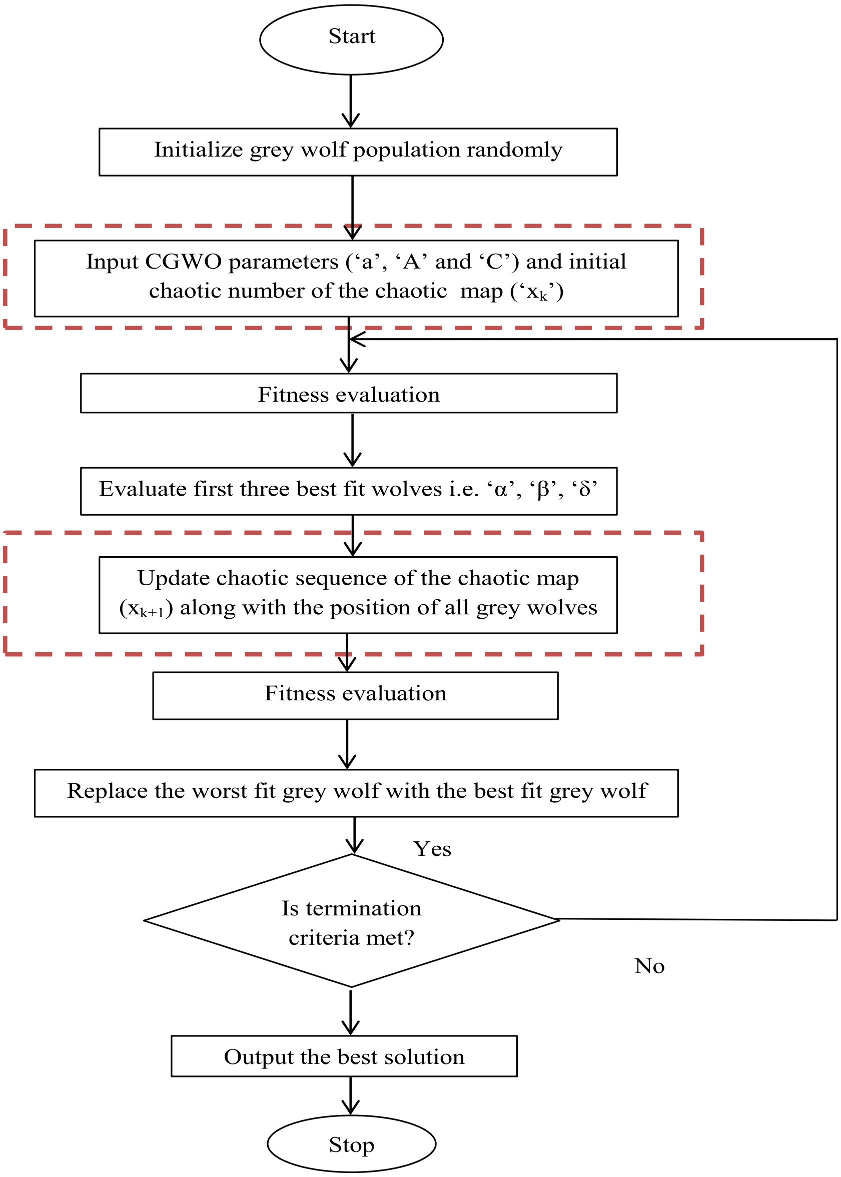
Start
Initialize grey wolf population randomly
Input CGWO parameters ('a', 'A' and 'C') and initial chaotic number of the chaotic map ('xk')
Fitness evaluation
Evaluate first three best fit wolves i.e. 'α', 'β', 'δ'
Update chaotic sequence of the chaotic map (xk+1) along with the position of all grey wolves
Fitness evaluation
Replace the worst fit grey wolf with the best fit grey wolf
Yes
Is termination criteria met?
No
Output the best solution
Stop

Fig. 1. Flowchart of optimization procedure of CGWO.

decreasing values from 2 to 0 over the course of iterations. For CGWO, values of two vectors $\vec{r}_1$ and $\vec{r}_2$ are taken randomly in the range of $(0, 1)$ and the controlling parameter $\vec{a}$ has linearly decreasing values from 2 to 0 over the course of iterations, chaotic function variables $a = 0.5$ and $b = 0.2$ is used. Additionally, the best map performer for constrained optimization problems as per results, i.e., chebyshev map has been used on CGWO constraint handling mechanism. CGWO is implemented in C++ and compiled using Qt Creator 2.4.1 (MinGW) under Microsoft Windows 7 operating system. All simulations are carried out on a computer with an Intel(R) Core(TM) $i5 - 3210@2.50$ Ghz CPU.

5.2. Results and discussion

In order to evaluate the capability of proposed CGWO for handling constrained problems, a set of thirteen widely used constrained benchmark functions have been used (Homaifar et al., 1994) and applied on all the maps mentioned in Table 1. All the problems consist in the set have various linear, non-linear and quadratic equations in the form of equality and inequality constraints which are represented in Table 2. To choose the best possible map for all the constrained optimization problems, all the selected maps are applied on all the constrained benchmark functions whose results are provided in Table 3. According to the results, chebyshev map showed promising performance by outperforming seven out of thirteen constrained benchmark functions and thus is chosen for further investigation of CGWO on constrained optimization problems.

The results of all the constrained benchmark functions applied on CGWO and other algorithms are given in Table 4. It can be easily seen from the results that CGWO has handle seven out of thirteen constrained benchmark functions very efficiently and thus has outperformed all other algorithms for these seven benchmark functions. PSO handled three constrained functions well among all. GWO performed superior to other for two constrained functions. FA outperformed for only one constrained function. The reason was that the chaotic maps with a unimode centered around the middle of maps, tend to produce better results, and chebyshev

```
Initialize the generation counter t and randomly initialize the population $X_i$
of grey wolves where $X_i(i=1, 2,..., n)$
Initialize the value of the chaotic map $x_0$ randomly.
Initialize parameters a, A and C
Calculate fitness of each Wolf
$X_\alpha$ = The best wolf
$X_\beta$ = The second best wolf
$X_\delta$ =The third best wolf
while (t <  Max_iterations)
Sort the population of grey wolves according to their fitness
Update the chaotic number using chaotic map equation
  for each search agent
        Update the position of current wolf using Eq. (4),
  end for
Update parameters a, A, C
Calculate the fitness of all wolves.
Update $X_\alpha, X_\beta, X_\delta$
Replace the worst fit wolf with the best fit wolf
  t=t+1
end while
return $X_\alpha$
```

Fig. 2. Pseudocode of proposed CGWO algorithm.

map fall into this category and it is indeed very effective. The results have also revealed the significant improvement of the proposed CGWO algorithm with the application of deterministic chaotic signals in place of constant value.

5.3. Graphical analysis

For further effective evaluation of performance of all the algorithms, graphical analysis has also been done. The line graphs of convergence of various constrained benchmark functions using CGWO algorithm and other algorithms viz. GWO, FA, FPA and PSO have been shown from Figs. 3–6 which help to analyze the convergence rate of each algorithm more effectively. All these graphs have been taken on 100 iterations to clearly notice and analyze the convergence of all the algorithms.

Fig. 3 shows the line graphs of convergence of all the five optimization algorithms applied on $G1$ test constrained benchmark function. From the graph, it can be seen that CGWO has the best performance for this benchmark function. It is showing superior performance of CGWO by reaching the optima for this test function within 10 iterations only. Further, it can be concluded from this graph that the GWO and FPA performed well when compared with other algorithms. FA demonstrates poor convergence in most of the optimization process, however it eventually ends the value of PSO.

Fig. 4 demonstrates the line graphs of convergence of CGWO along with all other algorithms for $G2$ constrained benchmark function in which it is easily remarkable that CGWO is fastest of all in context of convergence towards the optima than that of FA, FPA, GWO and PSO. PSO has shown a competitive performance to CGWO for this problem and exhibits significant performance in terms of convergence speed. GWO, FA and FPA show a faster convergence rate initially, however they seem to be trapped into suboptimal values as the optimization procedure proceeds.

Fig. 5 illustrates the convergence rate on $G9$ test constrained benchmark function in which CGWO is demonstrating the high rate of convergence as compared to FPA, FA, PSO and GWO. The convergence line graph of FPA and PSO is showing slow convergence by giving constant fitness values for many iterations in between the 100 iterations as they seem to be trapped into suboptimal values as the procedure proceeds, especially FPA. This demonstrates how CGWO is capable of balancing exploration and exploitation to find the global optimum rapidly and effectively.

Fig. 6 is presenting the graphical view of convergence of all algorithms on $G13$ test constrained benchmark function in which it can be clearly seen that CGWO algorithm is nearest to the global optima of this constrained problem among algorithms viz. FPA, FA, GWO and PSO and it also shows fastest convergence of all. GWO and FPA illustrate poorer convergence than the other algorithms in the initial iterations. However, the search process is progressively accelerated during iterations for these algorithms. This indicates that the performance of CGWO can be boosted by the chaotic maps in terms of not only exploration but also exploitation.

5.4. Statistical testing

Statistical testing is a process of making quantitative decisions about a problem in which statistical data set is evaluated and taken which is then compared hypothetically (Wilcoxon et al., 1970). The statistical testing of the constrained benchmark functions applied on all algorithms involved in this paper has been done using a widely used non parametric test named Wilcoxon signed rank-test discussed in Section 5.4.1.

5.4.1. Wilcoxon signed rank-test

Wilcoxon signed rank-test is a statistical method which is solely based on the order of the sample's observations (Wilcoxon et al., 1970). The one with lowest rank will be considered as the best among all and vice versa. The results of this statistical rank test done on all the algorithms have been demonstrated in Table 5 and their rank summary is provided in Table 6. Results depict that CGWO possessed lowest rank among all other optimization algorithms used in comparison for most of the benchmark functions which proves the superior performance of CGWO among other in comparison. However, PSO and GWO competed with CGWO closely and ranked second and third respectively. The superior performance of CGWO doesn't mean that it is superior than all other

Table 2
Details of constrained benchmark functions.

Problem	Type	Objective function	Constraints	Bounds	Optima	No. of variables
G1	Min	$f(x) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i$	$g_1(x) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leqslant 0,$ $g_2(x) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leqslant 0,$ $g_3(x) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leqslant 0,$ $g_4(x) = -8x_1 + x_{10} \leqslant 0,$ $g_5(x) = -8x_2 + x_{11} \leqslant 0,$ $g_6(x) = -8x_3 + x_{12} \leqslant 0,$ $g_7(x) = -2x_4 - x_5 + x_{10} \leqslant 0,$ $g_8(x) = -2x_6 - x_7 + x_{11} \leqslant 0,$ $g_9(x) = -2x_8 - x_9 + x_{12} \leqslant 0$	$L = (0, 0, \ldots, 0)$, $U = (1, 1, 1, 1, 1, 1,$ $1, 1, 1, 100, 100, 100, 1)$	−15	13
G2	Min	$f(x) = \left\lvert \frac{\sum_{i=1}^{n} cos^4(x_i) - 2\prod_{i=1}^{n} cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} ix_i^2}} \right\rvert$	$g_1(x) = -\prod_{i=1}^{n} x_i + 0.75 \leqslant 0,$ $g_2(x) = \sum_{i=1}^{n} x_i - 7.5n \leqslant 0$	$L = 0$, $U = 10$	−0.803619	20
G3	Max	$f(x) = \left(\sqrt{n}\right)^n \prod_{i=1}^{n} x_i$	$g_1(x) = \sum_{i=1}^{n} x_i^2 - 1$	$U = 1$	−1	20
G4	Min	$f(x) = 5.3578547x_3^2 + 0.8356891x_1x_5$ $+37.293239x_1 - 40792.141$	$g_1(x) = u(x) - 92 \leqslant 0,$ $g_2(x) = u(x) \leqslant 0,$ $g_3(x) = v(x) - 110 \leqslant 0,$ $g_4(x) = -v(x) + 90 \leqslant 0,$ $g_5(x) = w(x) - 25 \leqslant 0,$ $g_6(x) = -w(x) + 20 \leqslant 0$ where $u(x) = 85.334407 + 0.0056858x_2x_5$ $+0.0006262x_1x_5 + 0.0022053x_3x_5,$ $v(x) = 80.51249 + 0.0071317x_2x_5$ $+0.0029955x_1x_2 + 0.002181x_3^2,$ $w(x) = 9.300961 + 0.0047026x_3x_5$ $+0.0012547x_1x_3 + 0.0019085x_3x_4$	$L = (78, 33, 27, 27, 27),$ $U = (102, 45, 45, 45, 45)$	−30665.539	5
G5	Min	$f(x) = 3x_1 + 10^{-6}x_1^3 + 2x_2 + \frac{2}{3} \times 10^{-6}x_2^3$	$g_1(x) = x_3 - x_4 - 0.55 \leqslant 0,$ $g_2(x) = x_4 - x_3 - 0.55 \leqslant 0,$ $h_1(x) = 1000[\sin(-x_3 - 0.25)$ $+\sin(-x_4 - 0.25)] + 894.8 - x_1 = 0,$ $h_2(x) = 1000[\sin(x_3 - 0.25)$ $+\sin(x_3 - x_4 - 0.25)] + 894.8 - x_2 = 0$ $h_3(x) = 1000[\sin(x_4 - 0.25)$ $+\sin(x_4 - x_3 - 0.25)] + 1294.8 = 0$	$L = (0, 0, -0.55, -0.55),$ $U = (1200, 1200, 0.55, 0.55)$	5126.4981	4
G6	Min	$f(x) = (x_1 - 10)^3 + (x_2 - 20)^3$	$g_1(x) = (x_1 - 5)^2 + (x_2 - 5)^2 + 100 \leqslant 0,$ $g_2(x) = (x_1 - 5)^2 + (x_2 - 5)^2 - 82.81 \leqslant 0$	$L = (13, 0)$ $U = (100, 100)$	−6961.81388	2
G7	Min	$f(x) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2$ $+4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2$ $+7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$	$g_1(x) = 4x_1 + 5x_2 - 3x_7 + 9x_8 - 105 \leqslant 0,$ $g_2(x) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \leqslant 0,$ $g_{315}(x) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leqslant 0,$ $g_4(x) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \leqslant 0,$ $g_5(x) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leqslant 0.$ $g_6(x) = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \leqslant 0,$ $g_7(x) = x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leqslant 0,$ $g_8(x) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leqslant 0.$	$L = (-10, \ldots, -10), U = (10, \ldots, 10)$	24.306209	10
G8	Max	$f(x) = \frac{sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1 + x_2)}$	$g_1(x) = x_1^2 - x_2 + 1 \leqslant 0,$ $g_2(x) = 1 - x_1 + (x_2 - 4)^2 \leqslant 0$	$L = (0, 0)$ $U = (10, 10)$	−0.095825	2
G9	Min	$f(x) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4$ $+3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$	$g_1(x) = 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 - 127 \leqslant 0,$ $g_2(x) = 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 - 282 \leqslant 0,$ $g_3(x) = 23x_1 + x_2^2 + 6x_6^2 - 8x_7 - 196 \leqslant 0,$	$L = (-10, \ldots, -10)$ $U = (10, \ldots, 10)$	680.63005	7

(*continued on next page*)

Table 2 (*continued*)

Problem	Type	Objective function	Constraints	Bounds	Optima	No. of variables
			$g_4(x) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leqslant 0$			
G10	Min	$f(x) = x_1 + x_2 + x_3$	$g_1(x) = -1 + 0.0025(x_4 + x_6) \leqslant 0,$ $g_2(x) = -1 + 0.0025(-x_4 + x_5 + x_7) \leqslant 0,$ $g_3(x) = -1 + 0.01(-x_5 + x_8) \leqslant 0,$ $g_4(x) = 100x_1 - x_1x_6 + 833.33252x_4$ $-83333.333 \leqslant 0,$ $g_5(x) = x_2x_4 - x_2x_7 - 1250x_4 + 1250x_5 \leqslant 0,$ $g_6(x) = x_3x_5 - x_3x_8 - 2500x_5 + 1,250,000 \leqslant 0$	$L = (100, 1000, 1000, 10, 10, 10, 10, 10)$ $U = (10,000,\ 10,000,$ $10,000,\ 1000, 1000, 1000, 1000, 1000)$	7049.3307	8
G11	Min	$f(x) = x_1^2 + (x_2 - 1)^2$	$h_1(x) = x_2 - x_1^2 = 0$	$L = (-1, -1)$ $U = (1, 1)$	0.75	2
G12	Min	$f(x) = 1 - 0.01[(x_1 - 5)^2$ $+(x_2 - 5)^2 + (x_3 - 5)^2]$	$g_{i,j,k}(x) = (x_1 - i)^2 + (x_2 - j)^2 + (x_3 - k)^2$ $-0.0625 \leqslant i, j, k = 1, 2, \ldots 9$	$L = (0, 0, 0)$ $U = (10, 10, 10)$	−1	3
G13	Min	$f(x) = e^{x_1x_2x_3x_4x_5}$	$h_1(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0,$ $h_2(x) = x_2x_3 - 5x_4x_5 = 0,$ $h_3(x) = x_1^3 + x_2^3 + 1 = 0$	$L = -2.3$ $U = 2.3$	0.0539498	5

optimization algorithms present in the literature which will also lead to the violation of ‘free lunch theorem’ (Ho and Pepyne, 2002). Its performance simply signifies that it is better than other algorithms taken in this work only.

6. CGWO for classical engineering design problems

Engineering design is a process of satisfying the needs involved in building a product. It is a decision making process which consists of a complex objective function and a large number of decision variables such as weight, strength and wear (Askarzadeh, 2016). Meta-heuristic methods come into being as an alternative to the traditional optimization methods. With their merits of finding acceptable solutions in an affordable time and being tolerant of non-convex and non-differentiable, meta-heuristic algorithms have attracted great research interest during the recent years. In real design problems the number of design variables can be very large, and their influence on the objective function to be optimized can be very complicated, with a nonlinear character. Therefore, in this paper, design considerations of five classical engineering design problems viz. spring design problem, gear train design problem, welded beam design problem, pressure vessel design problem and closed coil helical spring design problem have been done in Sections 6.1,6.2,6.3,6.4,6.5. These problems contain various local optima, whereas only global optimum is required. These problems cannot be handled by traditional methods which focus on local optima only. Hence, there is a need for effective and efficient optimization methods for these engineering design problems. In this section, various experiments on these benchmark problems are implemented to verify the performance of the proposed meta-heuristic CGWO method. In order to get an unbiased comparison of CPU times, all the experiments are performed over 30 independent runs for 500 iterations.

6.1. Tension/Compression spring design problem

The main goal of this engineering design problem is to minimize the weight of the spring involving three decision variables which are wire diameter (d), mean coil diameter (D) and number of active coils (N). This problem is subjected to three inequality constraints and an objective function given in Eq. (8).

$$
\begin{aligned}
&\text{Consider} && \vec{x} = [x_1x_2x_3] = [dDN],\\
&\text{Minimize} && f(\vec{x}) = (x_3 + 2)x_2x_1^2,\\
&\text{Subject to} && g_1(\vec{x}) = 1 - \frac{x_2^3x_3}{717{,}854x_1^4} \leqslant 0,\\
& && g_2(\vec{x}) = \frac{4x_2^2 - x_1x_2}{12{,}566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} \leqslant 0,\\
& && g_3(\vec{x}) = 1 - \frac{140.45x_1}{x_2^2x_3} \leqslant 0,\\
& && g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \leqslant 0,
\end{aligned}
\tag{8}
$$

$$
\begin{aligned}
&\text{Variable range} && 0.05 \leqslant x_1 \leqslant 2.00,\\
& && 0.25 \leqslant x_2 \leqslant 1.30,\\
& && 2.00 \leqslant x_3 \leqslant 15.00
\end{aligned}
$$

Table 7 is showing the most optimal solution and the optimal values of decision variables found by CGWO algorithm. Table 8 is showing the comparison of all the simulation results for this problem applied on CGWO algorithm with conventional GWO algorithm and with those found by other optimization algorithms. It can be said from results that CGWO outperforms GWO (Mirjalili et al., 2014), CSA (Askarzadeh, 2016), GA3 (Coello, 2000), GA4 (Coello and Montes, 2002), CPSO (He and Wang, 2007), QPSO (dos Santos Coelho, 2010), PSO (dos Santos Coelho, 2010), SC (Ray and Liew, 2003), UPSO (Parsopoulos and Vrahatis, 2005) and

Table 3
Results of 10 chaotic maps on all constrained benchmark functions on CGWO.

Problem	Bernoulli	Logistic	Chebyshev	Circle	Cubic	Icmic	Peicewise	Singer	Sinusoidal	Tent
G1	−13.1952	−12.9301	−**14.8008**	−10.1684	−15.9102	−14.4705	−12.1389	−13.2742	−14.2749	−14.3466
G2	−0.42058	−0.567739	−**0.79434**	−0.24189	−0.51969	−0.475268	−0.460454	−0.33466	−0.38927	−0.46595
G3	−0.83865	−0.508866	−**0.9681**	−0.18356	−0.57743	−0.114344	−0.78943	−0.89999	−0.89258	−0.79427
G4	−33250.4	−32906.7	−32675.2	−32212.6	−31462.7	−33479.2	−32375.1	−**30902.2**	−31044.9	−31691.2
G5	53772.1	40974.8	54914.1	197252	39197.2	**23919.7**	58004.5	65223.1	45457.1	26868.2
G6	−6289.29	−6349.81	−**6493.18**	−6301.84	−6582.48	−6447.12	−6349.86	−6349.35	−6229.18	−6379.28
G7	130.1639	629.2649	60.2278	60.3222	**36.1793**	38.367	689.2759	228.2740	649.2649	928.1649
G8	−0.04561	−0.075206	−503.093	−0.05272	−0.07348	−0.09212	−0.09485	−0.06385	−0.02462	−**0.09566**
G9	602.173	612.460	**676.670**	612.370	628.40	665.643	607.135	614.274	629.153	657.12
G10	6994.23	6045.14	**7046.13**	6027.24	7034.43	7060.12	7024.24	7029.26	7013.17	7010.43
G11	0.6260	0.6250	0.6610	0.6400	0.6390	**0.6680**	0.6420	0.6250	0.6280	0.6390
G12	−73.2839	−70.1368	−**48.2570**	−82.3629	−193.363	−163.368	−273.368	−468.478	−2698.378	−2738.36
G13	0.4832	0.923643	0.5759	0.8933	**0.45489**	0.8935	1.3638	0.93538	1.353829	0.4678

Table 4
Comparison results of all constrained benchmark functions.

Problem	CGWO	GWO	FPA	FA	PSO
G1	−**14.8008**	−14.3159	−12.4265	−67.6314	−14.0273
G2	−**0.79434**	−0.31375	−0.30612	−0.517728	−0.65436
G3	−**0.9681**	−0.83910	−0.82839	−1.99369	−0.78568
G4	−32675.2	−33141.1	−33350.1	−**30446.7**	−32212.1
G5	54914.1	**43924.93**	58282.2	97119.4	79388.5
G6	−**6493.18**	−6265.65	−6346.38	−6349.86	−6248.57
G7	60.2278	42.1324	39.0470	27.6540	**24.1480**
G8	−503.093	−672.078	−10.7468	−11.0266	−**0.03440**
G9	676.670	603.816	813.734	680.438	**680.617**
G10	**7046.13**	6653.97	2821.31	6091.50	4691.59
G11	0.66100	**0.693021**	0.62507	0.62500	0.62508
G12	−**48.2570**	−47.3590	−72.9248	−53.2563	−1378.90
G13	**0.57590**	1.09478	0.67968	0.856731	0.82005

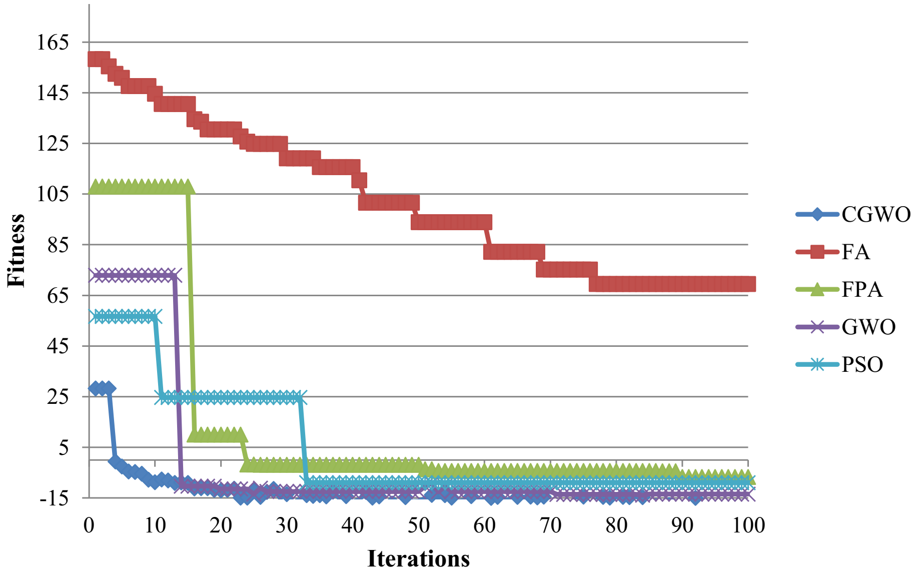

Fig. 3. Comparison of five optimization algorithms for the $G1$ constrained benchmark function in 100 iterations.

$(\mu+\lambda)-ES$ (Mezura-Montes and Coello, 2005). Also, the mean obtained by CGWO for this problem is better than those obtained by all other algorithms in comparison. Diagrammatical representation of the spring design problem is presented in Fig. 7.

6.2. Gear train design problem

The goal of this engineering design problem is to minimize the cost of gear ratio of the gear train whose schematic diagram is shown in Fig. 8. This problem has no equality or inequality constraints except a boundary constraint. It consists of four decision variables represented as $n_A(x_1)$, $n_B(x_2)$, $n_D(x_3)$, $n_F(x_4)$ using which the gear ratio can be formulated as $n_B n_D / n_F n_A$. Mathematical formulation of the objective function of gear train design problem along with its boundary constraint is given in Eq. (9).

$$\begin{aligned} &\text{Min.} \quad f(x) = ((1/6.931) - (x_3 x_2 / x_1 x_4))^2 \\ &\text{S.t.} \quad 12 \leqslant x_i \leqslant 60 \end{aligned} \tag{9}$$

Table 9 is showing the most optimal solution and the optimal values of decision variables found by CGWO algorithm. Table 10 is showing the comparison of all the simulation results for this problem applied on CGWO algorithm with conventional GWO

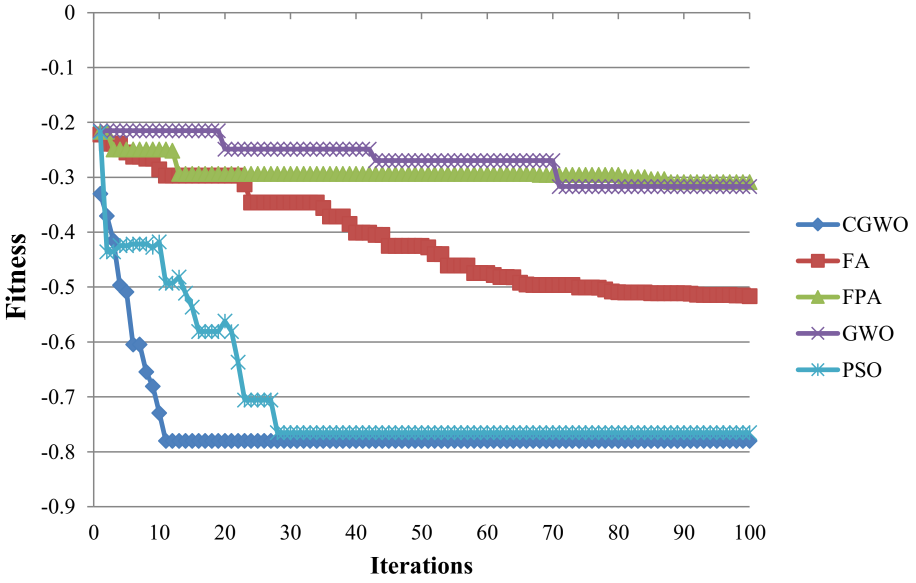

Fig. 4. Comparison of five optimization algorithms for the *G*2 constrained benchmark function in 100 iterations.

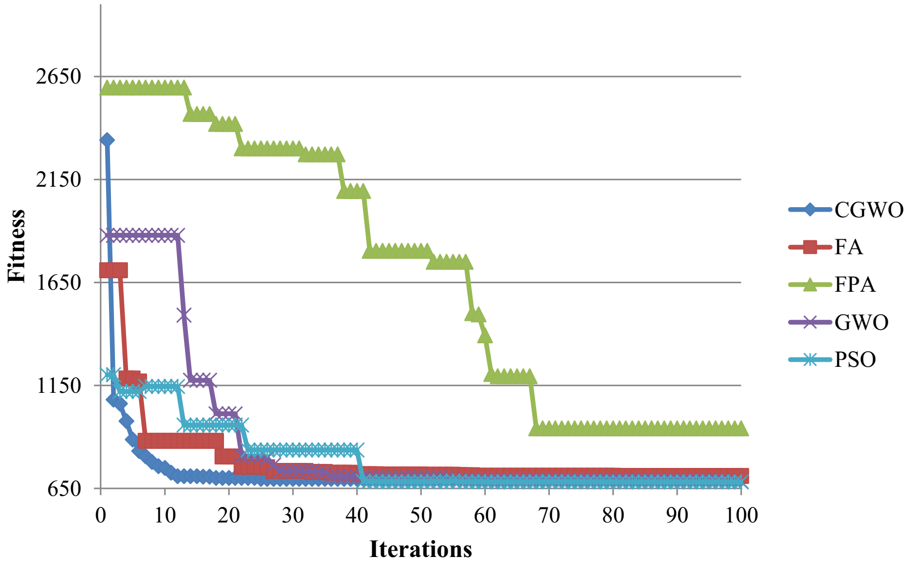

Fig. 5. Comparison of five optimization algorithms for the *G*9 constrained benchmark function in 100 iterations.

algorithm and with those found by other optimization algorithms. In terms of optimal results, CGWO outperforms GWO (Mirjalili et al., 2014), CSA (Askarzadeh, 2016), UPSO (Parsopoulos and Vrahatis, 2005), ABC (Akay and Karaboga, 2012) and MBA (Sadollah et al., 2013). In terms of mean, CGWO gives better value than those obtained by all other algorithms in comparison.

6.3. Welded beam design problem

Welded beam design problem which is a minimization problem has four variables namely weld thickness (h), length of bar attached to the weld (l), bar's height (t), bar's thickness (b) as shown in Fig. 9. The constraints included in this problem are bending stress (θ), beam deflection (δ), shear stress (τ), buckling load (P_c) and other side constraints. The mathematical formulas related to this problem are represented in Eq. (10).

$$
\begin{aligned}
&\text{Consider} && \vec{x} = [x_1 x_2 x_3 x_4] = [hltb],\\
&\text{Minimize} && f(\vec{x}) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2),\\
&\text{Subject to} && g_1(\vec{x}) = \tau(\vec{x}) - \tau_{max} \leqslant 0,\\
& && g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{max} \leqslant 0,\\
& && g_3(\vec{x}) = \delta(\vec{x}) - \delta_{max} \leqslant 0,\\
& && g_4(\vec{x}) = x_1 - x_4 \leqslant 0,\\
& && g_5(\vec{x}) = P - P_c(\vec{x}) \leqslant 0,\\
& && g_6(\vec{x}) = 0.125 - x_1 \leqslant 0,\\
& && g_7(\vec{x}) = 0.10471 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \leqslant 0
\end{aligned}
\tag{10}
$$

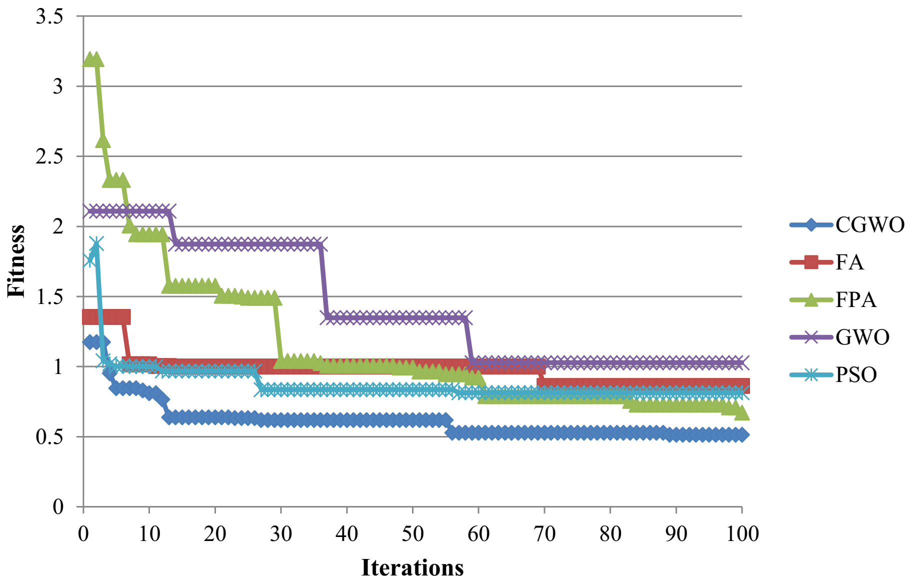

Fig. 6. Comparison of five optimization algorithms for the *G*13 constrained benchmark function in 100 iterations.

Table 5
Pair-wise Wilcoxon signed rank test results.

Function	Wilcoxon signed rank test order
G1	**CGWO** < GWO < PSO < FPA < FA
G2	**CGWO** < PSO < FA < GWO < FPA
G3	**CGWO** < GWO < FPA < PSO < FA
G4	FA < PSO < **CGWO** < GWO < FPA
G5	GWO < **CGWO** < FPA < PSO < FA
G6	**CGWO** < FA < FPA < GWO < PSO
G7	PSO < FA < FPA < GWO < **CGWO**
G8	PSO < FPA < FA < **CGWO** < GWO
G9	PSO < FA < **CGWO** < GWO < FPA
G10	**CGWO** < GWO < FA < PSO < FPA
G11	GWO < **CGWO** < PSO < FPA < FA
G12	**CGWO** < GWO < FA < FPA < PSO
G13	**CGWO** < FPA < PSO < FA < GWO

Variable range

$$0.1 \leqslant x_1 \leqslant 2,$$
$$0.1 \leqslant x_2 \leqslant 10,$$
$$0.1 \leqslant x_3 \leqslant 10,$$
$$0.1 \leqslant x_4 \leqslant 2$$

where

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

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

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

$$M = P(L + \frac{x_2}{2})$$

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

$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4}\left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

$$\sigma(\vec{x}) = \frac{6PL}{x_4x_3^2}$$

Table 6
Rank summary of statistical assessment results.

Function	CGWO	GWO	FPA	FA	PSO
G1	**1**	2	4	5	3
G2	**1**	4	5	3	2
G3	**1**	2	3	5	4
G4	3	4	5	**1**	2
G5	2	**1**	3	5	4
G6	**1**	4	3	2	5
G7	5	4	3	2	**1**
G8	4	5	2	3	**1**
G9	3	4	5	2	**1**
G10	**1**	2	5	3	4
G11	2	**1**	4	5	3
G12	**1**	2	4	3	5
G13	**1**	5	2	4	3
Total	**26**	40	48	43	38

Table 7
Optimal solution of spring design problem by CGWO algorithm.

Tension/Compression spring design problem				
Parameter	x_1	x_2	x_3	$f(x)$
Value	0.052796	0.804380	2.000000	0.0119598

Table 8
Comparison results of spring design problem.

Algorithm	Worst	Mean	Best	Std.
CGWO	**0.0121791**	**0.0121749**	**0.0119598**	**1.039E−05**
GWO	0.0122515	0.0121836	0.0126660	1.085E−05
CSA	0.0126701	0.0127690	0.0126652	1.357E−06
GA3	0.0128220	0.0127690	0.0127048	3.940E−05
GA4	0.0129730	0.0127420	0.0126810	5.90E−05
CPSO	0.0129240	0.0127330	0.0126747	5.20E−04
HPSO	0.0127190	0.0127072	0.0126652	1.58E−05
G-QPSO	0.0177590	0.0135240	0.0126650	0.001268
QPSO	0.0181270	0.0138540	0.0126690	0.001341
PSO	0.0718020	0.0195550	0.0128570	0.011662
DSS-MDE	0.0127382	0.0126693	0.0126652	1.25E−05
PSO-DE	0.0126653	0.0126652	0.0126652	1.2E−08
SC	0.0167172	0.0129226	0.0126692	5.9E−04
UPSO	N.A.	0.0229400	0.0131200	7.2E−03
$(\mu + \lambda) - ES$	N.A.	0.0131650	0.0126890	3.9E−04
ABC	N.A.	0.0127090	0.0126650	0.01281
TLBO	N.A.	0.0126657	0.0126650	N.A.
MBA	0.0129000	0.0127130	0.0126650	6.3E−05

N.A. – Not Available.

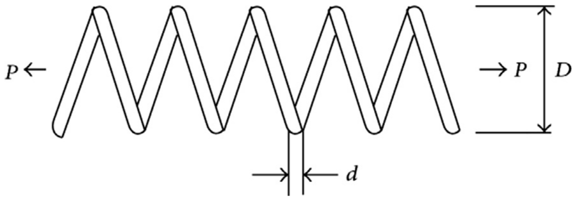

Fig. 7. Structure of tension/spring design (Rao et al., 2011).

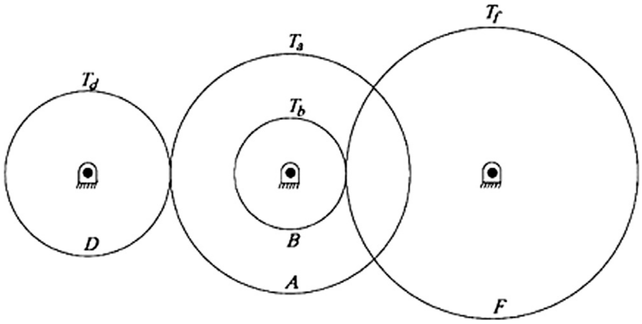

Fig. 8. Structure of gear train design (Rao et al., 2011).

$$\delta(\vec{x}) = \frac{4PL^3}{Ex_3^2x + x_4}$$

$$P_c(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$$P = 6000\text{ lb}, \quad L = 14\text{ in.}, \quad \delta_{max} = 0.25\text{ in.}, \quad E = 30 \times 10^6\text{ psi},$$

$$G = 12 \times 10^6\text{ psi}, \quad \tau_{max} = 13,600\text{ psi}, \quad \sigma_{max} = 30,000\text{ psi}$$

Table 11 is showing the most optimal solution and the optimal values of decision variables found by CGWO algorithm. Table 12 is showing the comparison of all the simulation results for this problem applied on CGWO algorithm, conventional GWO algorithm and with those found by other optimization algorithms. In terms of best result, CGWO outperforms GWO (Mirjalili et al., 2014), GA3 (Coello, 2000), GA4 (Coello and Montes, 2002), CPSO (He and Wang, 2007), SC (Ray and Liew, 2003), UPSO (Parsopoulos and Vrahatis, 2005) and CDE (He and Wang, 2007). Also, the mean obtained by CGWO for this problem is better than those obtained by all other algorithms in comparison.

6.4. Pressure vessel design problem

Pressure vessel design problem is a classical engineering design problem whose main goal is to minimize the welding, manufacturing and material cost of the pressure vessel. There are a total of four decision variables involved in this problem which are thickness of shell (T_s), thickness of head (T_h) which are discrete decision variables, inner radius (R) and length of cylindrical section of the vessel (L) which are continuous decision variables. The diagrammatical representation of pressure vessel design problem is given in Fig. 10 showing variables of pressure vessel. The mathematical equations of the nonlinear objective function and constraints is represented in Eq. (11). The mentioned problem has four inequality constraints.

Table 9
Optimal solution of gear train design problem by CGWO algorithm.

Gear train design problem					
Parameter	x_1	x_2	x_3	x_4	$f(x)$
Value	45.1903	21.2025	14.6466	50.2213	2.833970E−13

Table 10
Comparison results of gear train design problem.

Algorithm	Worst	Mean	Best	Std.
CGWO	**2.71358E−10**	**7.09107E−11**	**2.833970E−13**	**1.02462E−10**
GWO	5.03136E−09	1.62918E−09	1.568642E−11	1.76011E−09
CSA	3.18473E−08	2.05932E−09	2.700857E−12	5.059779E−9
UPSO	N.A.	3.80562E−08	2.700857E−12	1.09000E−09
ABC	N.A.	3.64133E−10	2.700857E−12	5.52000E−09
MBA	2.06290E−08	2.47163E−09	2.700857E−12	3.94000E−09

N.A. – Not Available.

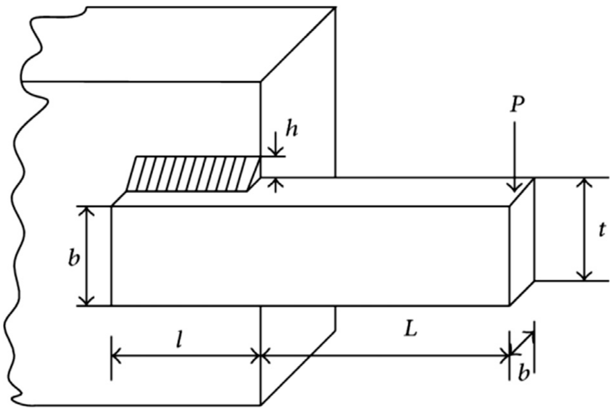

Fig. 9. Structure of welded beam design (Rao et al., 2011).

$$
\begin{array}{ll}
\text{Consider} & \vec{x} = [x_1x_2x_3x_4] = [T_sT_hRL], \\
\text{Minimize} & f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 \\
 & \quad +3.1661x_1^2x_4 + 19.84x_1^2x_3, \\
\text{Subject to} & g_1(\vec{x}) = -x_1 + 0.0193x_3 \leqslant 0, \\
 & g_2(\vec{x}) = -x_2 + 0.00954x_3 \leqslant 0, \\
 & g_3(\vec{x}) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1,296,000 \leqslant 0, \\
 & g_4(\vec{x}) = x_4 - 240 \leqslant 0 \\
\text{Variable range} & \\
 & 0 \leqslant x_1 \leqslant 100, \\
 & 0 \leqslant x_2 \leqslant 100, \\
 & 10 \leqslant x_3 \leqslant 200, \\
 & 10 \leqslant x_4 \leqslant 200
\end{array}
\tag{11}
$$

Table 12
Comparison results of welded beam design problem.

Algorithm	Worst	Mean	Best	Std.
CGWO	**2.435700**	**2.428900**	**1.725450**	**1.35780**
CPSO	1.782143	1.748831	1.728024	1.29E−2
GA4	1.993408	1.792654	1.728226	7.47E−2
GA3	1.785835	1.771973	1.748309	1.12E−2
CDE	N.A.	1.768150	1.733460	N.A.
UPSO	N.A.	2.837210	1.921990	0.68300
GWO	2.913600	2.859400	1.942100	2.69080
SC	6.399678	3.002588	2.385434	9.60E−1

N.A. – Not Available.

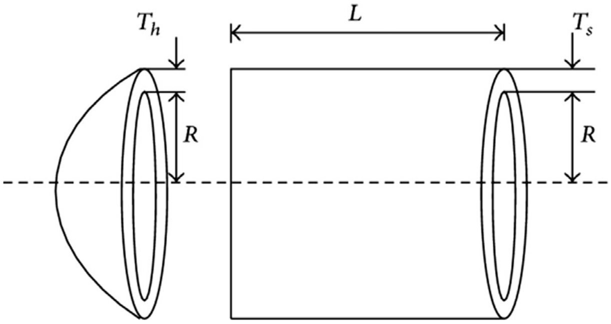

Fig. 10. Structure of pressure vessel design problem (Rao et al., 2011).

Table 13 is showing the most optimal solution and the optimal values of decision variables found by CGWO algorithm. Table 14 is showing the comparison of all the simulation results for this problem applied on CGWO algorithm, conventional GWO algorithm and with those found by other optimization algorithms. It can be said from results that CGWO outperforms GWO (Mirjalili et al., 2014), $(\mu + \lambda) - ES$ (Mezura-Montes and Coello, 2005), CSA (Askarzadeh, 2016),HPSO (He and Wang, 2007), PSO-DE (Liu et al., 2010), ABC (Akay and Karaboga, 2012), TLBO (Rao et al., 2011), G-QPSO (dos Santos Coelho, 2010), QPSO (dos Santos Coelho, 2010), CDE (He and Wang, 2007), GA4 (Coello and Montes, 2002), CPSO (He and Wang, 2007), UPSO (Parsopoulos and Vrahatis, 2005), GA3 (Coello, 2000) and PSO (dos Santos Coelho, 2010). Also, the mean obtained by CGWO for this problem is better than those obtained by all other algorithms in comparison.

6.5. Closed coil helical spring design problem

The main goal of this mechanical engineering design constrained problem is to minimize the volume of closed coil helical spring. Helical spring is made up of closed coiled wire having the shape of a helix and is intended for tensile and compressive load (Savsani et al., 2010). From Fig. 11, it can be seen that the coils of spring are so closed that the plane is at nearly right angles to the helix's axis and coil is subjected to torsion. This problem has a total of two decision variables namely coil diameter (D) and wire diameter (d) whose range is given in Eq. (13). The volume of the helical spring (U) can be minimized using the objective function given in Eq. (12).

Table 11
Optimal solution of welded beam design problem by CGWO algorithm.

Welded beam design problem					
Parameter	x_1	x_2	x_3	x_4	$f(x)$
Value	0.343891	1.883570	9.03133	0.212121	1.72545

Table 13
Optimal solution of pressure vessel design problem by CGWO algorithm.

Pressure vessel design problem					
Parameter	x_1	x_2	x_3	x_4	$f(x)$
Value	1.187150	0.600000	69.707500	7.7984400	5034.1800

Table 14
Comparison results of pressure vessel design problem.

Algorithm	Worst	Mean	Best	Std.
CGWO	**6188.110**	**5783.582**	**5034.180**	**254.505**
GWO	6395.360	6159.320	6051.563	379.674
$(\mu+\lambda)-ES$	N.A.	6379.938	6059.701	210.000
CSA	7332.841	6342.499	6059.714	384.945
HPSO	6288.677	6099.932	6059.714	86.200
PSO-DE	N.A.	6059.714	6059.714	N.A.
ABC	N.A.	6245.308	6059.714	205.000
TLBO	N.A.	6059.714	6059.714	N.A.
G-QPSO	7544.492	6440.378	6059.720	448.471
QPSO	8017.281	6440.378	6059.720	479.267
CDE	6371.045	6085.230	6059.734	43.0130
GA4	6469.322	6177.253	6059.946	130.929
CPSO	6363.804	6147.133	6061.077	86.4500
UPSO	9387.770	8016.370	6154.700	745.869
GA3	6308.497	6293.843	6288.744	7.41330
PSO	14076.324	8756.680	6693.721	1492.56

N.A. – Not Available.

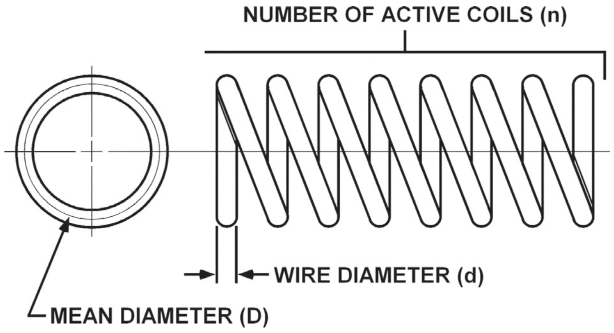

Fig. 11. Structure of closed coil helical spring design problem (Savsani et al., 2010).

$$U = \frac{\pi^2}{4}(N_c + 2)Dd^2 \tag{12}$$

where

$$\begin{aligned} &0.508 \leqslant d \leqslant 1.016, \\ &1.270 \leqslant D \leqslant 7.620, \\ &15 \leqslant N_c \leqslant 25 \end{aligned} \tag{13}$$

The constrained problem of helical spring is subjected to eight constraints out of which first is stress constraint represented in Eq. (14) and second is configuration constraint given in Eq. (15).

$$\begin{aligned} &S - 8C_f F_{max}\frac{D}{\pi d^3} \geqslant 0, \\ &C_f = \frac{4C-1}{4C-4} + \frac{0.615}{C}, \\ &C = \frac{D}{d} \end{aligned} \tag{14}$$

Here, F_{max} which is the maximum load and S, the shear stress allowed on the spring are set to 453.6 kg and 13288.02 kg f/cm^2 respectively.

$$K = \frac{Gd^4}{8N_c D^3} \tag{15}$$

where G is set to 808543.6 kg f/cm^2 and K is the spring constant. Next constraint is the length constraint expressed as given in Eq. (18) in which the maximum length l_{max} is equal to 35.56 cm. l_f is the free length which can be calculated using Eq. (17). The deflection (δ_l) made in the spring due to maximum work load is also involved in calculating free length as given in Eq. (16).

$$\delta_l = \frac{F_{max}}{K} \tag{16}$$

$$l_f = \delta_l + 1.05(N_c + 2)d \tag{17}$$

$$l_{max} - l_f \geqslant 0 \tag{18}$$

The wire diameter should also follow the constraint represented in Eq. (19) where d_{min} is set to 0.508 cm.

$$d - d_{min} \geqslant 0 \tag{19}$$

The outer diameter of the coil (D) spring should also be less the maximum diameter specified (D_{max}) which is 7.62 cm. Mathematically, it is expressed in Eq. (20).

$$D_{max} - (D + d) \geqslant 0 \tag{20}$$

The mean coil diameter (C) must also be at least three times the diameter of the wire as represented mathematically in Eq. (21).

$$C - 3 \geqslant 0 \tag{21}$$

Next, the deflection occurs under the preload δ_p must also be less than its specified value δ_{pm} which is 15.24 cm as represented in Eq. (22). The preload deflection can be calculated using Eq. (23).

$$\delta_{pm} - \delta_p \geqslant 0 \tag{22}$$

Table 15
Optimal solution of closed coil helical spring design problem by CGWO algorithm.

Closed coil helical spring design problem			
Parameter	d	D	$f(x)$
Value	0.599394	1.92367	42.0990

Table 16
Comparison results of closed coil helical spring design problem.

Algorithm	Worst	Mean	Best	Std.
CGWO	**42.9625**	**41.9815**	**42.0990**	**2.7502**
GWO	44.5842	43.6468	43.6524	1.7684
DTLBO	46.4322	46.3192	46.3012	N.A.
TLBO	46.5214	46.4998	46.3221	N.A.
Conventional	N.A.	N.A.	46.4392	N.A.
PSO	46.6752	46.6254	46.5212	N.A.
ABS	46.6241	46.6033	46.5115	N.A.
GA	46.3932	46.6821	46.6653	N.A.

N.A. – Not Available.

$$\delta_p = \frac{F_p}{K} \qquad (23)$$

Here, F_p is set to 136.08 kg. The combined deflection constraint is given in Eq. (24) which makes the deflection of the coil consistent with its length.

$$l_f - \delta_p = \frac{F_{max} - F_p}{K} - 1.05(N_c + 2)d \geqslant 0 \qquad (24)$$

The next and the last constraint is subjected to the preload deflection of the spring which defines that it must be equal to its specified value (δ_ω) which is equal to 3.175 cm. It is expressed in Eq. (25).

$$\frac{F_{max} - F_p}{K} - \delta_\omega \leqslant 0 \qquad (25)$$

Table 15 is showing the most optimal solution and the optimal values of decision variables found by CGWO algorithm. Table 16 is showing the comparison of all the simulation results for this problem applied on CGWO algorithm, GWO algorithm and with those found by other optimization algorithms. It can be said from results that CGWO outperforms GWO (Mirjalili et al., 2014), DTLBO (Thamaraikannan and Thirunavukkarasu, 2014), TLBO (Rao et al., 2011), Conventional method (Hinze, 2005), PSO (He et al., 2004), ABS (Thamaraikannan and Thirunavukkarasu, 2014) and GA (Das and Pratihar, 2002). Also, the mean obtained by CGWO for this problem is better than those obtained by all other algorithms in comparison.

7. Conclusion and future scope

The chaos theory and Grey Wolf Optimizer (GWO) are hybridized in order to design an improved meta-heuristic Chaotic Grey Wolf Optimization (CGWO) algorithm for constrained optimization problems. Various chaotic maps are used to regulate the key parameter, a, of GWO. The proposed CGWO is validated on thirteen constrained benchmark functions and five constrained engineering design problems. The chebyshev map is selected as its a through comparing various chaotic GWO variants to form the best CGWO. The simulations showed that the usage of deterministic chaotic signals instead of linearly decreasing values is an important modification of the GWO algorithm. Statistical results and success rates of the CGWO suggest that the tuned GWO clearly improves the reliability of the global optimality and they also enhanced the quality of the results. In comparison with other algorithms viz. FA, FPA, GWO and PSO, it seems the CGWO performed significantly well. The results of CGWO on constrained engineering problems showed its applicability for the complex real-world problems. The main reason of the superior performance of CGWO lies behind the chaos induced by the chaotic maps in the search space. This chaos helps the controlling parameter to find the optimal solution more quickly and thus refine the convergence rate of the algorithm. So, it can be easily concluded here that proposed CGWO can handle constrained problems effectively and efficiently. Further investigation on convergence analysis may prove fruitful. In addition, further topics of studies can also focus on the extension of the CGWO to solve mixed-type problems and discrete optimization problems.

References

Akay, B., & Karaboga, D. (2012). Artificial bee colony algorithm for large-scale problems and engineering design optimization. *Journal of Intelligent Manufacturing, 23*(4), 1001–1014.

Alavi, A. H., & Gandomi, A. H. (2011). Prediction of principal ground-motion parameters using a hybrid method coupling artificial neural networks and simulated annealing. *Computers & Structures, 89*(23), 2176–2194.

Alavi, A. H., Gandomi, A. H., Sahab, M. G., & Gandomi, M. (2010). Multi expression programming: A new approach to formulation of soil classification. *Engineering with Computers, 26*(2), 111–118.

Arora, S., & Singh, S. (2013). A conceptual comparison of firefly algorithm, bat algorithm and cuckoo search. In *International conference on control computing communication & materials (ICCCCM), 2013* (pp. 1–4). IEEE.

Arora, S., & Singh, S. (2015). Butterfly algorithm with levy flights for global optimization. In *2015 International conference on signal processing, computing and control (ISPCC)* (pp. 220–224). IEEE.

Arora, S., & Singh, S. (2017). An improved butterfly optimization algorithm with chaos. *Journal of Intelligent & Fuzzy Systems, 32*(1), 1079–1088.

Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Computers & Structures, 169*, 1–12.

Cagnina, L. C., Esquivel, S. C., & Coello, C. A. C. (2008). Solving engineering optimization problems with the simple constrained particle swarm optimizer. *Informatica, 32*(3).

Coello, C. A. C. (2000). Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry, 41*(2), 113–127.

Coello, C. A. C., & Montes, E. M. (2002). Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics, 16*(3), 193–203.

Das, A., & Pratihar, D. (2002). Optimal design of machine elements using a genetic algorithm. *Journal of the Institution of Engineers (India), Part MC, Mechanical Engineering Division, 83*(3), 97–104.

Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering, 186*(2), 311–338.

dos Santos Coelho, L. (2010). Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. *Expert Systems with Applications, 37*(2), 1676–1683.

Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016). Binary grey wolf optimization approaches for feature selection. *Neurocomputing, 172*, 371–381.

Gandomi, A. H., & Yang, X.-S. (2014). Chaotic bat algorithm. *Journal of Computational Science, 5*(2), 224–232.

Gandomi, A., Yang, X.-S., Talatahari, S., & Alavi, A. (2013). Firefly algorithm with chaos. *Communications in Nonlinear Science and Numerical Simulation, 18*(1), 89–98.

Gandomi, A. H., Yun, G. J., Yang, X.-S., & Talatahari, S. (2013). Chaos-enhanced accelerated particle swarm optimization. *Communications in Nonlinear Science and Numerical Simulation, 18*(2), 327–340.

Gandomi, A. H., Yang, X.-S., Alavi, A. H., & Talatahari, S. (2013). Bat algorithm for constrained optimization tasks. *Neural Computing and Applications, 22*(6), 1239–1255.

Gao, X.-Z., Jokinen, T., Wang, X., Ovaska, S. J., & Arkkio, A. (2010). A new harmony search method in optimal wind generator design. In *2010 XIX international conference on electrical machines (ICEM)* (pp. 1–6). IEEE.

Gao, X.-Z., Wang, X., Ovaska, S. J., & Xu, H. (2010). A modified harmony search method in constrained optimization. *International Journal of Innovative Computing, Information and Control, 6*(9), 4235–4247.

Han, X., & Chang, X. (2012). A chaotic digital secure communication based on a modified gravitational search algorithm filter. *Information Sciences, 208*, 14–27.

Han, X., & Chang, X. (2013). An intelligent noise reduction method for chaotic signals based on genetic algorithms and lifting wavelet transforms. *Information Sciences, 218*, 103–118.

He, Q., & Wang, L. (2007). An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence, 20*(1), 89–99.

He, Q., & Wang, L. (2007). A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Applied Mathematics and Computation, 186*(2), 1407–1422.

He, D., He, C., Jiang, L.-G., Zhu, H.-W., & Hu, G.-R. (2001). Chaotic characteristics of a one-dimensional iterative map with infinite collapses. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 48*(7), 900–906.

He, S., Prempain, E., & Wu, Q. (2004). An improved particle swarm optimizer for mechanical design optimization problems. *Engineering Optimization, 36*(5), 585–605.

Herskovits, J. (1986). A two-stage feasible directions algorithm for nonlinear constrained optimization. *Mathematical Programming, 36*(1), 19–38.

Hinze, M. (2005). A variational discretization concept in control constrained optimization: The linear-quadratic case. *Computational Optimization and Applications, 30*(1), 45–61.

Ho, Y.-C., & Pepyne, D. L. (2002). Simple explanation of the no free lunch theorem of optimization. *Cybernetics and Systems Analysis, 38*(2), 292–298.

Homaifar, A., Qi, C. X., & Lai, S. H. (1994). Constrained optimization via genetic algorithms. *Simulation, 62*(4), 242–253.

Jia, D., Zheng, G., & Khan, M. K. (2011). An effective memetic differential evolution algorithm based on chaotic local search. *Information Sciences, 181*(15), 3175–3187.

Joines, J. A., & Houck, C. R. (1994). On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA's. In *Proceedings of the first IEEE conference on evolutionary computation, 1994. IEEE world congress on computational intelligence* (pp. 579–584). IEEE.

Karaboga, D., & Akay, B. (2011). A modified artificial bee colony (abc) algorithm for constrained optimization problems. *Applied Soft Computing, 11*(3), 3021–3031.

Karaboga, D., & Basturk, B. (2007). Artificial bee colony (abc) optimization algorithm for solving constrained optimization problems. In *Foundations of fuzzy logic and soft computing* (pp. 789–798). Springer.

Kennedy, J. (2011). Particle swarm optimization. In *Encyclopedia of machine learning* (pp. 760–766). Springer.

Komaki, G. M., & Kayvanfar, V. (2015). Grey wolf optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time. *Journal of Computational Science, 8*, 109–120.

Lee, K. S., & Geem, Z. W. (2004). A new structural optimization method based on the harmony search algorithm. *Computers & Structures, 82*(9), 781–798.

Liu, H., Cai, Z., & Wang, Y. (2010). Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Applied Soft Computing, 10*(2), 629–640.

Luo, Q., Ma, M., & Zhou, Y. (2016). A novel animal migration algorithm for global numerical optimization. *Computer Science and Information Systems, 13*(1), 259–285.

Madadi, A., & Motlagh, M. M. (2014). Optimal control of dc motor using grey wolf optimizer algorithm. *TJEAS Journal, 4*(4), 373–379.

Meng, X.-B., Gao, X., Lu, L., Liu, Y., & Zhang, H. (2015). A new bio-inspired optimisation algorithm: Bird swarm algorithm. *Journal of Experimental & Theoretical Artificial Intelligence*, 1–15.

Meng, Z., Pan, J.-S., & Alelaiwi, A. (2016). A new meta-heuristic ebb-tide-fish-inspired algorithm for traffic navigation. *Telecommunication Systems, 62*(2), 403–415.

Mezura-Montes, E., & Coello, C. A. C. (2005). A simple multimembered evolution strategy to solve constrained optimization problems. *IEEE Transactions on Evolutionary Computation, 9*(1), 1–17.

Mezura-Montes, E., & Coello, C. A. C. (2005). Useful infeasible solutions in engineering optimization with evolutionary algorithms. In *MICAI 2005: Advances in artificial intelligence* (pp. 652–662). Springer.

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95*, 51–67.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46–61.

Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Computing and Applications, 27*(2), 495–513.

Mohamed, A.-A. A., El-Gaafary, A. A., Mohamed, Y. S., & Hemeida, A. M. (2015). Energy management with capacitor placement for economics low carbon emissions using modified multi-objective grey wolf optimizer.

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)* (pp. 209–214). IEEE.

Parsopoulos, K. E., & Vrahatis, M. N. (2002). Particle swarm optimization method for constrained optimization problems. *Intelligent Technologies–Theory and Application: New Trends in Intelligent Technologies, 76*, 214–220.

Parsopoulos, K. E., & Vrahatis, M. N. (2005). Unified particle swarm optimization for solving constrained engineering optimization problems. In *Advances in natural computation* (pp. 582–591). Springer.

Pecora, L. M., & Carroll, T. L. (1990). Synchronization in chaotic systems. *Physical Review Letters, 64*(8), 821.

Powell, M. J. (1978). A fast algorithm for nonlinearly constrained optimization calculations. In *Numerical analysis* (pp. 144–157). Springer.

Rao, R. (2016). Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International Journal of Industrial Engineering Computations, 7*(1), 19–34.

Rao, R. V., Savsani, V. J., & Vakharia, D. (2011). Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design, 43*(3), 303–315.

Ray, T., & Liew, K. M. (2003). Society and civilization: An optimization algorithm based on the simulation of social behavior. *IEEE Transactions on Evolutionary Computation, 7*(4), 386–396.

Sadollah, A., Bahreininejad, A., Eskandar, H., & Hamdi, M. (2013). Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Applied Soft Computing, 13*(5), 2592–2612.

Saremi, S., Mirjalili, S., & Lewis, A. (2014). Biogeography-based optimisation with chaos. *Neural Computing and Applications, 25*(5), 1077–1097.

Savsani, V., Rao, R., & Vakharia, D. (2010). Optimal weight design of a gear train using particle swarm optimization and simulated annealing algorithms. *Mechanism and Machine Theory, 45*(3), 531–541.

Shi, Y. (2015). An optimization algorithm based on brainstorming process. *Emerging Research on Swarm Intelligence and Algorithm Optimization*, 1–35.

Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. In *The 1998 IEEE international conference on evolutionary computation proceedings, 1998. IEEE world congress on computational intelligence* (pp. 69–73). IEEE.

Talatahari, S., Azar, B. F., Sheikholeslami, R., & Gandomi, A. (2012). Imperialist competitive algorithm combined with chaos for global optimization. *Communications in Nonlinear Science and Numerical Simulation, 17*(3), 1312–1319.

Thamaraikannan, B., & Thirunavukkarasu, V. (2014). Design optimization of mechanical components using an enhanced teaching-learning based optimization algorithm with differential operator. *Mathematical Problems in Engineering, 2014*. 10 pages.

Wang, G.-G., Guo, L., Gandomi, A. H., Hao, G.-S., & Wang, H. (2014). Chaotic krill herd algorithm. *Information Sciences, 274*, 17–34.

Wang, G.-G., Guo, L., Duan, H., & Wang, H. (2014). A new improved firefly algorithm for global numerical optimization. *Journal of Computational and Theoretical Nanoscience, 11*(2), 477–485.

Wilcoxon, F., Katti, S., & Wilcox, R. A. (1970). Critical values and probability levels for the Wilcoxon rank sum test and the Wilcoxon signed rank test. *Selected Tables in Mathematical Statistics, 1*, 171–259.

Yang, X.-S. (2012). Flower pollination algorithm for global optimization. In *Unconventional computation and natural computation* (pp. 240–249). Springer.

Yang, X.-S., & algorithm, Firefly (2010). Levy flights and global optimization. In *Research and development in intelligent systems XXVI* (pp. 209–218). Springer.

Yang, D., Li, G., & Cheng, G. (2007). On the efficiency of chaos optimization algorithms for global optimization. *Chaos, Solitons & Fractals, 34*(4), 1366–1375.

Yang, X.-S., Gandomi, A. H., Talatahari, S., & Alavi, A. H. (2012). *Metaheuristics in water, geotechnical and transport engineering*. Newnes.