Swarm and Evolutionary Computation 44 (2019) 101–112

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

A novel Random Walk Grey Wolf Optimizer

Shubham Gupta *, Kusum Deep

Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand 247667, India

ARTICLE INFO

Keywords:
Optimization
Swarm intelligence
Grey Wolf Optimizer
Random walk

ABSTRACT

Grey Wolf Optimizer (GWO) algorithm is a relatively new algorithm in the field of swarm intelligence for solving continuous optimization problems as well as real world optimization problems. The Grey Wolf Optimizer is the only algorithm in the category of swam intelligence which is based on leadership hierarchy. This paper has three important aspects- Firstly, for improving the search ability by grey wolf a modified algorithm RW-GWO based on random walk has been proposed. Secondly, its performance is exhibited in comparison with GWO and state of art algorithms GSA, CS, BBO and SOS on IEEE CEC 2014 benchmark problems. A non-parametric test Wilcoxon and Performance Index Analysis has been performed to observe the impact of improving the leaders in the proposed algorithm. The results presented in this paper demonstrate that the proposed algorithm provide a better leadership to search a prey by grey wolves. The third aspect of the paper is to use the proposed algorithm and GWO on real life application problems. It is concluded from this article that RW-GWO algorithm is an efficient and reliable algorithm for solving not only continuous optimization problems but also for real life optimization problems.

1. Introduction

Swarm Intelligence is one of the most promising area for solving real world optimization problems. The collaboration among social creatures for searching the food and designing an intelligent framework is known as Swarm intelligence. To deal with non-linear non-convex, discontinuous and discrete optimization problems many algorithms have been developed by simulating the swarming behavior of various intelligent creatures like ant, wolf, honey bees, birds and whales etc. Particle Swarm Optimization (PSO) [1], Ant Colony Optimization (ACO) [2], Firefly Algorithm [3], Artificial Bee Colony (ABC) [4], Spider Monkey Optimization (SMO) [5], Whale Optimization [6], Grey Wolf Optimizer (GWO) [7] and many other algorithms which are based on swarm intelligence have proven that these algorithms have a great potential to deal with real world optimization problems.

One of the interesting developments in the area of numerical optimization was the publication of the No Free Lunch (NFL) theorem [8]. This theorem states that the performance of all optimization (search) algorithms, amortized over the set of all possible functions, is equivalent. That is, if one problem can be solved by an algorithm effectively, then it is not necessary that it will be solved effectively by another algorithm. This theorem forms the basis of proposing many nature inspired optimization algorithm now and then.

The Grey Wolf Optimizer (GWO) is based on leadership hierarchy theory introduced by Mirjalili et al. [7]. GWO algorithm is relatively a new contribution to the family of swarm intelligence based meta-heuristics. In the family of Swarm intelligence based algorithms, GWO is the only algorithm which is based on leadership hierarchy. The GWO algorithm is a simple population based algorithm which simulates the leadership and social behavior of the grey wolf for hunting prey.

From recent years a substantial growth in the study of GWO can be observed for solving different real life application problems. Ina Ref. [9] GWO has been used for adjustment of PID controller parameters in DC motors. For training q-Gaussian radial basis functional link-nets neural network Muangkote et al. [10] have proposed the improved version of GWO. In Song et al. [11] GWO is used for surface waves analysis. In Ref. [12] GWO and DE have been used for single and multi-objective optimal flow problem. In training of Multi-Layer perceptron (MLP) Mirjalili [13] used GWO to analyse its performance. Shakarami and Davoudkhani [14] used a strategy based on GWO for wide area power system stabilizer (WAPSS). In Ref. [15] GWO is used for solving load frequency control (LFC) problem in an interconnected power system. For the solution of non-convex economic load dispatch problem Grey Wolf Optimizer has been used in Ref. [16]. Zhang et al. [17] have used GWO for path planning tasks. In Ref. [18] GWO has been hybridized with crossover and mutation to solve four problems of economic dispatch. In Ref. [19] grouped GWO has been proposed for maximum power point tracking of doubly-fed induction generator.

For a dynamic welding scheduling problem Lu et al. [20] have used hybrid multi-objective GWO. In Ref. [21] Hybrid version of GWO named

* Corresponding author.
E-mail addresses: g.shubh93@gmail.com (S. Gupta), kusumfma@iitr.ac.in (K. Deep).

https://doi.org/10.1016/j.swevo.2018.01.001
Received 29 June 2017; Received in revised form 7 November 2017; Accepted 4 January 2018
Available online 6 January 2018
2210-6502/

Hybrid Grey Wolf Optimizer and Genetic Algorithm (HGWOGA) has been designed for minimizing the potential energy of molecules. Emary et al. [22] proposed the binary version of GWO for feature selection.

These applications have shown the ability of Grey Wolf Optimizer in terms of exploration strength compared to other metaheuristic algorithms.

In Ref. [23] fuzzy hierarchical operators are used in GWO to present five variants of GWO and to analyse their performance. In Heidari and Pahlavani [24], a modified GWO with Levy flight has been proposed. In Ref. [25] Mirjalili developed the GWO for multi-objective optimization problems.

In this paper an improved variant of GWO, Random Walk Grey Wolf Optimizer (RW-GWO) is proposed to increase the potential of GWO algorithm. The proposed RW-GWO algorithm is obtained by improving the leaders on the basis of some drawbacks that leading wolves in GWO have. The performance of both these algorithms is evaluated on CEC 2014 benchmark functions.

The rest of the paper is organized as follows Section 2 provides a motivation and brief introduction of GWO. In Section 3 motivation and improved version of GWO named RW-GWO have been proposed. Numerical results on CEC 2014 benchmark functions have been presented in Section 4. Section 5 shows the statistical validity of the results presented in Section 4 for the comparison between GWO and proposed version of GWO. In Section 6 results on some real life applications are presented. Finally, in Section 7 we conclude the paper and suggest some future ideas.

2. Grey Wolf Optimizer (GWO)

This section briefly introduces the inspiration behind the Grey Wolf Optimizer, physical behavior of wolf pack and working of the algorithm.

2.1. Inspiration

GWO algorithm is inspired from the leadership hierarchy and hunting strategy of grey wolves. Grey wolves, also known as timber or western wolves always live in a pack of approximately 5–11 wolves. Social hierarchy is the main feature of their pack. For hunting the prey and to maintain the discipline within pack they categorize their group into four types of wolves. The first category consist of leader wolf called alpha wolf (α), which can be male or female and is responsible for the vital decisions of the pack. The wolves in the second category are the subservient wolves that are liable to deliver the messages of alpha wolf to the other wolves and helps the alpha wolf in making decisions like hunting and for selecting place for residence etc. Wolves in this category are called beta wolves (β). In the last category of the pack, the wolves that have permission for eating food in the end are included. These wolves are known as omega wolves (ω). These wolves are very important part of the pack because in the absence of omega wolves, pack may face some internal problems of fighting. The remaining wolves are categorized as delta wolves (δ). In this category the caretakers, hunters and sentinels are included. Caretaker wolves takes care of wounded wolves in the pack. Hunters are contributing their role in hunting process of prey and sentinel wolves work as defender of the pack from external enemies.

The group hunting is another major and important feature of the pack. According to Muro et al. [26] their hunting process involves three steps – *i*) Chasing and approaching the prey. *ii*) Encircling the prey and *iii*) attacking the prey.

2.2. Mathematical modal

After analyzing the social behavior of grey wolves, Mirjalili et al. [7] modelled the social behavior and hunting process to design GWO. A step wise brief description is given below

2.2.1. Social behavior

By analyzing the social behavior of wolf pack the candidate having best fitness is considered as alpha wolf or α solution. Candidates having second best and third best fitness are called as Beta wolf or β solution and delta wolf or δ solution and remaining solutions are considered as Omega wolves or ω solutions. The ω solutions are iteratively improved by following other leading wolves.

2.2.2. Encircling prey

The encircling strategy by the wolves around the prey is mathematically modelled by proposing the following equations as

$$X_{t+1} = X_{p,t} - \mu \cdot d \tag{1}$$

$$d = \left| c \cdot X_{p,t} - X_t \right| \tag{2}$$

$$\mu = 2 \cdot b \cdot r_1 - b \tag{3}$$

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

where X_{t+1} is the position of the wolf at $(t+1)^{th}$ iteration X_t is the position of the wolf at t^{th} iteration, $X_{p,t}$ is the position of the prey at t^{th} iteration. d is difference vector expressed by (2), μ and c are coefficient vectors and b is a linearly decreasing vector from 2 to 0 over iterations, expressed as

$$b = 2 - 2 \cdot \left(\frac{t}{maximum\ \ no.\, of\ \ iterations} \right) \tag{5}$$

and r_1, r_2 are the uniformly distributed random vectors whose component lie between 0 and 1.

2.2.3. Hunting

Hunting strategy of the grey wolves can be mathematically modelled by approximating the prey position with the help of α, β and δ solutions (wolves). Therefore by following this approximation each wolf can update their positions by

$$X_1' = X_\alpha - \mu_\alpha \cdot d_\alpha \tag{6}$$

$$X_2' = X_\beta - \mu_\beta \cdot d_\beta \tag{7}$$

$$X_3' = X_\delta - \mu_\delta \cdot d_\delta \tag{8}$$

$$X_{t+1} = \frac{(X_1' + X_2' + X_3')}{3} \tag{9}$$

where X_α, X_β, X_δ are the positions approximated by α, β and δ solutions (wolves) with the help of eq. (1).

2.2.4. Exploration and exploitation in attacking prey

It is obvious that when the prey stops moving, wolf will kill the prey and in this way they complete their hunting process. In hunting process when or $c > 1$ wolf will explore the whole search space for finding the prey (optima), and wolves will explo $|\mu| > 1$ it the search space when $|\mu| < 1$ or $c < 1$. When t → maximum number of iterations, then by eq. (5) $b \to 0 \Rightarrow \mu \to 0$ therefore in this situation coefficient c is responsible for exploration.

In this way grey wolves complete their process of hunting by repeating the encircling and hunting steps as described above. The pseudo code of original GWO algorithm is presented in Fig. 1.

3. Proposed RW-GWO algorithm

3.1. Reasons behind improving the search ability by the leaders

As defined in the literature of the GWO in previous section, grey wolf

```
Initialize the Grey Wolf population x_i (i = 1,2,...,n)
Initialize parameters b, μ and c as defined in subsection 2.2.2
Initialize l = 1, the iteration number
Evaluate the fitness of each wolf
Select  α = fittest wolf of the pack
        β = second best wolf
        δ = third best wolf
 while l < maximum number of iterations
     for each wolf
         update the position by equation (9)
     end
   update b, μ and c
   update α, β and δ wolves
   l = l + 1
 end
```

Fig. 1. Pseudo code of GWO.

pack successfully complete their process of hunting by the guidance of the leaders of the pack namely α, β and δ wolves of the pack.

Mathematically speaking, in GWO, each wolf update its position with the help of these leading wolves. Thus α, β and δ wolves are the leading responsible search agents in updating the state of each wolf and provides an optimum direction towards the prey. Therefore it is very important that in each iteration these leading wolves should be the best (in terms of fitness), so that each wolf will get an optimum guidance to approach a prey.

Since in GWO algorithm it is described that all the wolves of the pack will update their positions according to leading wolves of the pack with the help of eq. (9). But then the natural question arises that which leading wolf will help in updating the position of alpha wolf as it is the dominant wolf of the pack, and why alpha wolf will take the guidance of low fitted (inferior) wolves beta and delta of pack to update its position? Similarly why beta wolf will update its position with the help of low fitted (inferior) wolf delta of the pack? This is the main drawback in GWO algorithm, and probably this is the reason why the pack does not converge to global optima. So during an iteration to update the position of each wolf of the pack selection of the leading wolves is very important as each wolf is updated by leading wolves. Therefore an improvisation is needed in the leaders of the pack to avoid the problem of premature convergence due to the stagnation in local optima and to main a social behavior within the pack.

To accomplish this, in the present paper random walk based Grey Wolf Optimizer has been proposed in which leaders explore the search space by random walk and then omega wolves update their position by following them.

3.2. Random walk

Random walk [27] is a random process that consists of consecutive random steps. In mathematical terms a random walk

$$W_N = \sum_{i=1}^{N} s_i$$

where s_i is a random step that can be taken from any random distribution. The relationship between any two consecutive random walks is defined as

$$W_N = \sum_{i=1}^{N} s_i = \sum_{i=1}^{N-1} s_i + X_N = W_{N-1} + s_N$$

This relationship shows that the next state W_N is only dependent on current state W_{N-1} and the step taken from current state to the next state. Step size s_i can be fixed or can vary. So for a wolf starting with a point x_0 and suppose its final location is x_N, then a random walk can also be defined as

$$x_n = x_0 + \alpha_1 s_1 + \alpha_2 s_2 + \cdots + \alpha_N s_N = x_0 + \sum_{i=1}^{N} \alpha_i s_i$$

where $\alpha_i > 0$ is a parameter that controls the step size s_i in each iteration.

3.3. Design of proposed random walk based GWO: RW-GWO

As the GWO starts with an initial population of wolves say x_i, $i = 1, 2, \ldots, n$, therefore iteratively in each iteration a random walk has been incorporated for only the leaders α, β and δ of the population in which parameter α_i has been taken as a vector that is linearly decreasing from 2 to 0 as the iterations proceed. In this proposed algorithm random walk is incorporated in which step size is drawn from a Cauchy distribution.

The reason of considering Cauchy distributed random step size is that as the variance of a Cauchy distribution is infinity, so it may take a longer jump sometime that is very effective at that the time of stagnation and thus very helpful for the leading wolves to explore the search space for finding prey and provide a great guidance for other wolves. Here it is noted that any extra objective function evaluation efforts are not incorporated in the algorithm. So the function evaluation remains same in both the algorithm.

The pseudo code of the proposed algorithm RW-GWO is described in Fig. 2 and the flow chart of the algorithm RW-GWO is presented in Fig. 3.

4. Numerical experiments

In this section the performance of GWO and proposed RW-GWO is studied on the basis of the criteria of IEEE CEC 2014 benchmark functions [28]. This benchmark set consists of 30 unconstrained optimization problems of unimodal, multimodal, hybrid and composite type with varying difficulty levels. All these numerical experiments have been performed on MATLAB 2010a.

4.1. Benchmark functions

As per the guidelines provided by IEEE CEC 2014 [28], for each of the test function 51 runs are performed for the dimensions 10 and 30 to observe the performance of both the algorithms. The range of the search space for each variable is [-100, 100]. Initial population is uniformly generated using random number generator. The termination criteria is set to be the maximum function evaluation ($10^D \times dimension$).

4.2. Analysis of the results

All the results presented in this paper are in the format provided by the IEEE CEC 2014 [28]. Table 1 shows the results obtained by performing GWO and RW-GWO on CEC 2014 test functions for 10 dimension. Table 2 provides the results of the same experiment conducted for 30 dimension. In these tables maximum, minimum, mean, median and standard deviation of the absolute error value of the objective function value for the test functions are reported. In Tables 1 and 2 the better results are highlighted with bold letters. As the unimodal problems are suitable for the evaluation of exploitation strength of any search algorithm, the functions F1-F3 are unimodal in CEC 2014 benchmark problem set. From Tables 1 and 2 the superior performance of RW-GWO than GWO can be observed in all the statistics. Thus in terms of exploiting the search regions around the explored search regions RW-GWO is better than GWO.

The multimodal test problem usually examine the exploration strength and local optima avoidance ability of the search algorithm. In CEC 2014 test problems, functions F4-F16 are multimodal. For the dimension 10, average error in RW-GWO is less than GWO in all the functions except F11 and for the dimension 30, in all the problems RW-

```
Initialize the Grey wolf population x_i (i = 1,2,...,n)
Initialize parameters b, μ and c as defined in subsection 2.2.2
Initialize l = 1, the iteration number
Evaluate the fitness of each wolf
Select  α = fittest wolf of the pack
        β = second best wolf
        δ = third best wolf
while l < maximum number of iterations
 Evaluate the fitness of each wolf
        for each leader wolf
             find new position y_i of the leaders x_i by random walk
              if f(y_i) < f(x_i)
              update the leaders
        end
        for each ω wolf
          update the position by equation (9) and apply greedy approach between current and
        updated positions
        end
     update b, μ and c
     update α, β and δ wolves
     l = l + 1
end
```

Fig. 2. Pseudo code of RW-GWO.

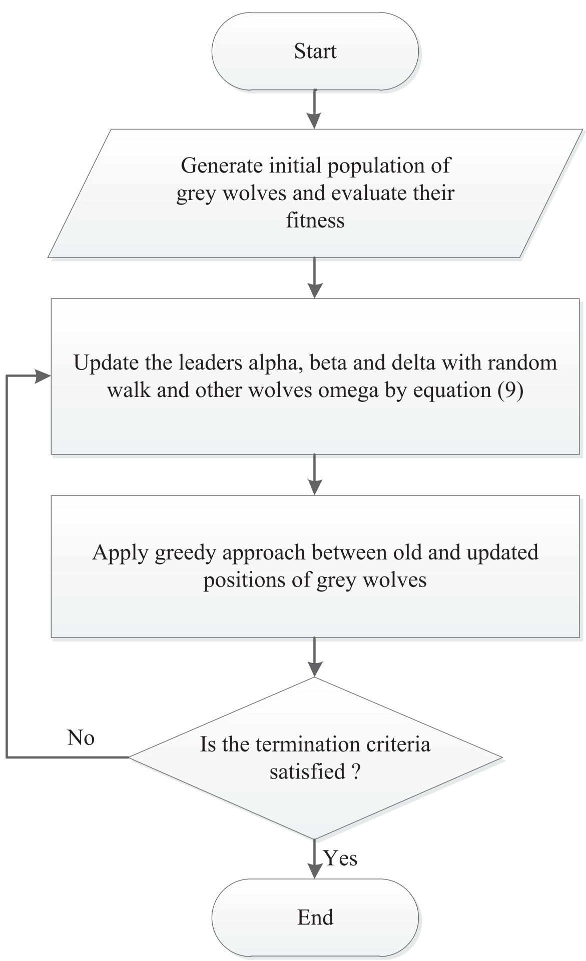

Fig. 3. Flow chart of proposed RW-GWO algorithm.

GWO is better than GWO in terms of average error. In terms of minimum error RW-GWO is better than GWO in all problems except the problem F7, F12 for 10 dimension and except F11, F14 and F15 for 30 dimension. Thus in exploring the promising regions of the search space RW-GWO is better than GWO in most of the problems.

The hybrid and composite problems are used to evaluate the strength of avoiding stagnation problem in local optima due to massive number of local optima and to evaluate the ability of balancing exploration and exploitation in metaheuristic algorithm. The problems F17-F22 are hybrid functions and F23-F30 problems are composite functions. From the table it can be observed than for the dimension 10, in the problems except F18 and for the dimension 30 in all the problems RW-GWO shows its superior performance than GWO.

Overall in terms of exploring, exploiting and escaping from the stagnation in local optima RW-GWO algorithm outperforms GWO.

4.3. Comparison with other metaheuristic algorithms

Although the structure of the Grey Wolf Optimizer algorithm is different from other metaheuristic algorithm, so the comparison of GWO with other search algorithms is not significant, but for solving the particular real life optimization problem with inference based on the their performance on standard test problems comparison can be done. Therefore in this section, performance of RW-GWO has been compared with the performance of state of art algorithms GSA [24,29], CS [24,30], LX-BBO [31], B-BBO [31] and SOS [24,32] on 30 dimensional CEC 2014 problems based on the average error in objective function value. The maximum number of function evaluations (termination criteria) is taken same ($10^4 \times dimension$) for all the algorithms for fair comparison. The comparison is presented in Table 3. From the analysis of the results reported in Table 3, it can be concluded that the proposed algorithm RW-GWO provides a very competitive results compared to other metaheuristic algorithms.

4.4. Algorithm complexity

As per the guidelines of IEEE CEC 2014, the complexity of an algorithm is calculated on CEC 2014 benchmark functions. The parameters T0, T1 and T2 are same as defined in IEEE CEC 2014. T0 is the computing time of the specified test program given in CEC 2014. T1 is the computing time for 2×10^5 function evaluations of function F18 only and T2 is computing time taken by an algorithm to run 2×10^5 function evaluations of F18 only.

The algorithm complexities are shown in Table 4. For better view of complexity Fig. 4 has been plotted. This shows that GWO has slightly higher complexity than RW-GWO. Therefore RW-GWO is better algorithm compared to GWO in terms of time complexity.

Table 1
Average, Standard Deviation, Median, Minimum, Maximum error value obtained by GWO and RW-GWO for 10-dimentional IEEE CEC2014 benchmark problems. Better results are highlighted in bold.

Function	Algorithm	Mean	Std Dev	Median	Minimum	Maximum
1	GWO	1.81E+06	1.80E+06	1.42E+06	1.59E+05	9.37E+06
	RW-GWO	**1.52E+05**	**6.45E+04**	**1.72E+05**	**9.98E+02**	**2.71E+05**
2	GWO	1.03E+07	5.17E+07	**1.13E+03**	1.91E+02	2.65E+08
	RW-GWO	**2.19E+03**	**2.36E+03**	1.24E+03	**7.83E+01**	**9.28E+03**
3	GWO	4.55E+03	3.50E+03	4.41E+03	9.91E+01	1.30E+04
	RW-GWO	**2.64E+01**	**6.46E+01**	**8.17E+00**	**1.60E-01**	**4.35E+02**
4	GWO	3.27E+01	8.88E+00	3.52E+01	6.05E+00	3.80E+01
	RW-GWO	**6.55E+00**	**1.21E+00**	**6.77E+00**	**2.55E+00**	**8.89E+00**
5	GWO	2.03E+01	**2.41E-01**	2.04E+01	1.87E+01	2.05E+01
	RW-GWO	**1.96E+01**	2.80E+00	**2.00E+01**	**5.00E-02**	**2.00E+01**
6	GWO	1.92E+00	**1.10E+00**	1.79E+00	2.14E-01	5.54E+00
	RW-GWO	**1.49E+00**	1.11E+00	**1.71E+00**	**9.34E-02**	**3.99E+00**
7	GWO	1.11E+00	8.29E-01	8.41E-01	**6.78E-02**	3.66E+00
	RW-GWO	**1.61E-01**	**5.25E-02**	**1.61E-01**	8.06E-02	**2.97E-01**
8	GWO	1.05E+01	4.69E+00	8.96E+00	3.98E+00	2.59E+01
	RW-GWO	**4.51E+00**	**1.47E+00**	**3.98E+00**	**1.99E+00**	**8.96E+00**
9	GWO	1.47E+01	7.13E+00	1.32E+01	6.16E+00	3.40E+01
	RW-GWO	**1.07E+01**	**4.70E+00**	**9.95E+00**	**3.00E+00**	**2.49E+01**
10	GWO	3.95E+02	2.11E+02	3.78E+02	1.42E+02	1.00E+03
	RW-GWO	**1.34E+02**	**6.98E+01**	**1.49E+02**	**1.53E+01**	**2.69E+02**
11	GWO	**4.70E+02**	2.18E+02	**4.37E+02**	1.31E+02	1.16E+03
	RW-GWO	5.55E+02	**1.95E+02**	5.59E+02	**1.25E+02**	**1.10E+03**
12	GWO	6.08E-01	5.22E-01	4.58E-01	**1.28E-02**	1.58E+00
	RW-GWO	**8.93E-02**	**3.92E-02**	**7.64E-02**	2.35E-02	**1.84E-01**
13	GWO	1.69E-01	5.81E-02	1.71E-01	7.77E-02	3.16E-01
	RW-GWO	**1.23E-01**	**3.10E-02**	**1.24E-01**	**7.47E-02**	**1.82E-01**
14	GWO	3.37E-01	2.18E-01	2.18E-01	5.15E-02	7.09E-01
	RW-GWO	**1.37E-01**	**9.51E-02**	**1.26E-01**	**2.91E-02**	**5.79E-01**
15	GWO	1.57E+00	7.89E-01	1.54E+00	4.47E-01	3.90E+00
	RW-GWO	**7.41E-01**	**1.98E-01**	**6.98E-01**	**2.96E-01**	**1.24E+00**
16	GWO	2.31E+00	**5.51E-01**	2.33E+00	9.82E-01	**3.40E+00**
	RW-GWO	**2.09E+00**	5.67E-01	**2.13E+00**	**5.13E-01**	3.51E+00
17	GWO	3.85E+03	3.20E+03	2.57E+03	8.23E+02	1.69E+04
	RW-GWO	**1.97E+03**	**1.86E+03**	**1.43E+03**	**1.19E+02**	**7.78E+03**
18	GWO	**3.59E+03**	**4.02E+03**	**1.31E+03**	1.01E+02	**1.52E+04**
	RW-GWO	7.09E+03	6.35E+03	6.67E+03	**1.39E+01**	2.24E+04
19	GWO	2.37E+00	9.09E-01	2.03E+00	1.34E+00	5.80E+00
	RW-GWO	**1.75E+00**	**6.15E-01**	**1.62E+00**	**5.55E-01**	**3.39E+00**
20	GWO	1.45E+03	2.38E+03	1.16E+02	3.83E+01	8.18E+03
	RW-GWO	**1.47E+01**	**9.47E+00**	**1.24E+01**	**2.95E+00**	**5.18E+01**
21	GWO	1.79E+03	1.69E+03	9.38E+02	1.13E+02	6.18E+03
	RW-GWO	**4.45E+02**	**5.43E+02**	**2.82E+02**	**3.34E+01**	**3.03E+03**
22	GWO	1.16E+02	**5.70E+01**	1.46E+02	2.66E+01	1.71E+02
	RW-GWO	**6.59E+01**	5.71E+01	**3.73E+01**	**1.37E+00**	**1.64E+02**
23	GWO	3.35E+02	4.54E+00	3.35E+02	3.29E+02	3.45E+02
	RW-GWO	**3.29E+02**	**8.39E-05**	**3.29E+02**	**3.29E+02**	**3.29E+02**
24	GWO	1.33E+02	2.34E+01	1.27E+02	1.11E+02	2.03E+02
	RW-GWO	**1.19E+02**	**5.97E+00**	**1.19E+02**	**1.07E+02**	**1.35E+02**
25	GWO	1.92E+02	**1.83E+01**	2.00E+02	1.37E+02	**2.02E+02**
	RW-GWO	**1.85E+02**	2.30E+01	**1.98E+02**	**1.32E+02**	2.03E+02
26	GWO	1.00E+02	4.08E-02	1.00E+02	1.00E+02	1.00E+02
	RW-GWO	**1.00E+02**	**2.96E-02**	**1.00E+02**	**1.00E+02**	**1.00E+02**
27	GWO	3.35E+02	**7.15E+01**	3.46E+02	4.42E+00	**4.08E+02**
	RW-GWO	**3.25E+02**	8.60E+01	**3.40E+02**	**1.21E+00**	4.23E+02
28	GWO	4.03E+02	7.01E+01	3.71E+02	**2.39E+02**	6.91E+02
	RW-GWO	**3.06E+02**	**9.33E-02**	**3.06E+02**	3.06E+02	**3.07E+02**
29	GWO	4.76E+04	3.35E+05	6.36E+02	3.54E+02	2.39E+06
	RW-GWO	**2.05E+02**	**1.64E+00**	**2.05E+02**	**2.02E+02**	**2.11E+02**
30	GWO	1.01E+03	3.40E+02	8.99E+02	5.97E+02	2.08E+03
	RW-GWO	**3.15E+02**	**9.30E+01**	**2.82E+02**	**2.24E+02**	**5.90E+02**

4.5. Convergence analysis

As the fittest solution in each iteration is known as alpha solution (wolf). Therefore to analyze the convergence of the algorithms average of alpha solution of 51 runs for different iterations of some selected CEC 2014 functions have been plotted for 30 dimension in Figs. 5 and 6. In these figures horizontal axis indicates the number of iterations and the vertical axis represents the objective function value of functions. From the figures it can be concluded that convergence in RW-GWO is better than GWO i.e. in RW-GWO leaders are converging to the optima faster and avoids stagnation in most of the cases.

5. Statistical analysis

In this section two different statistical tools are used to analyze the performance of GWO and RW-GWO.

I. Wilcoxon test
II. Performance Index (PI)

Table 2
Average, Standard Deviation, Median, Minimum, Maximum error value obtained by GWO and RW-GWO for 30- dimensional IEEE CEC2014 benchmark problems. Better results are highlighted in bold.

Function	Algorithm	Mean	Std Dev	Median	Minimum	Maximum
1	GWO	3.32E+07	2.02E+07	3.04E+07	8.18E+06	9.79E+07
	RW-GWO	**8.02E+06**	**3.31E+06**	**7.66E+06**	**2.22E+06**	**1.94E+07**
2	GWO	1.01E+09	1.15E+09	6.10E+08	1.42E+06	5.72E+09
	RW-GWO	**2.23E+05**	**5.51E+05**	**9.28E+04**	**2.83E+04**	**3.35E+06**
3	GWO	2.85E+04	6.80E+03	2.81E+04	1.58E+04	4.58E+04
	RW-GWO	**3.16E+02**	**4.34E+02**	**5.57E+01**	**1.23E+01**	**1.34E+03**
4	GWO	1.94E+02	5.82E+01	1.77E+02	1.00E+02	3.83E+02
	RW-GWO	**3.41E+01**	**1.80E+01**	**2.81E+01**	**1.87E+01**	**8.29E+01**
5	GWO	2.10E+01	**4.83E-02**	2.10E+01	2.08E+01	2.10E+01
	RW-GWO	**2.05E+01**	7.46E-02	**2.05E+01**	**2.03E+01**	**2.07E+01**
6	GWO	1.16E+01	**2.64E+00**	1.17E+01	6.24E+00	1.82E+01
	RW-GWO	**9.84E+00**	3.49E+00	**1.03E+01**	**3.21E+00**	**1.80E+01**
7	GWO	7.67E+00	4.64E+00	5.31E+00	2.27E+00	1.86E+01
	RW-GWO	**2.53E-01**	**1.43E-01**	**2.21E-01**	**8.68E-02**	**8.85E-01**
8	GWO	6.50E+01	1.45E+01	6.25E+01	3.42E+01	1.22E+02
	RW-GWO	**4.38E+01**	**8.48E+00**	**4.34E+01**	**2.49E+01**	**6.64E+01**
9	GWO	8.54E+01	3.30E+01	8.05E+01	3.88E+01	2.42E+02
	RW-GWO	**6.33E+01**	**1.30E+01**	**6.37E+01**	**3.41E+01**	**9.42E+01**
10	GWO	1.80E+03	4.93E+02	1.74E+03	6.99E+02	3.08E+03
	RW-GWO	**9.61E+02**	**2.72E+02**	**9.47E+02**	**5.23E+02**	**1.60E+03**
11	GWO	2.90E+03	7.24E+02	2.81E+03	**1.47E+03**	6.45E+03
	RW-GWO	**2.68E+03**	**3.68E+02**	**2.62E+03**	1.79E+03	**3.49E+03**
12	GWO	2.12E+00	9.58E-01	2.44E+00	**8.20E-02**	3.13E+00
	RW-GWO	**5.45E-01**	**1.66E-01**	**5.17E-01**	2.57E-01	**1.12E+00**
13	GWO	3.74E-01	8.88E-02	3.77E-01	2.19E-01	6.92E-01
	RW-GWO	**2.80E-01**	**6.30E-02**	**2.66E-01**	**1.85E-01**	**4.60E-01**
14	GWO	7.49E-01	1.40E+00	7.08E-01	**1.24E-01**	1.04E+01
	RW-GWO	**4.23E-01**	**2.15E-01**	**3.01E-01**	1.85E-01	**7.72E-01**
15	GWO	2.06E+01	2.20E+01	1.46E+01	**3.96E+00**	1.39E+02
	RW-GWO	**8.81E+00**	**1.51E+00**	**8.79E+00**	5.08E+00	**1.26E+01**
16	GWO	1.09E+01	**5.80E-01**	1.10E+01	9.45E+00	1.20E+01
	RW-GWO	**1.03E+01**	6.11E-01	**1.02E+01**	**8.98E+00**	**1.15E+01**
17	GWO	6.28E+05	6.11E+05	**4.46E+05**	**4.61E+04**	3.59E+06
	RW-GWO	**5.71E+05**	**4.10E+05**	4.52E+05	5.68E+04	**2.06E+06**
18	GWO	5.27E+06	1.34E+07	2.11E+04	2.12E+03	6.41E+07
	RW-GWO	**6.52E+03**	**4.62E+03**	**6.23E+03**	**4.89E+02**	**1.83E+04**
19	GWO	2.56E+01	1.77E+01	2.07E+01	7.50E+00	8.35E+01
	RW-GWO	**1.14E+01**	**2.03E+00**	**1.11E+01**	**7.40E+00**	**1.61E+01**
20	GWO	1.31E+04	5.26E+03	1.19E+04	4.00E+03	2.90E+04
	RW-GWO	**6.27E+02**	**1.12E+03**	**2.66E+02**	**1.02E+02**	**6.00E+03**
21	GWO	4.97E+05	1.05E+06	**1.60E+05**	6.12E+04	4.74E+06
	RW-GWO	**2.58E+05**	**1.76E+05**	2.42E+05	**2.60E+04**	**6.22E+05**
22	GWO	2.50E+02	**1.16E+02**	1.90E+02	5.13E+01	6.32E+02
	RW-GWO	**2.08E+02**	1.29E+02	**1.62E+02**	**3.32E+01**	**5.43E+02**
23	GWO	3.28E+02	4.16E+00	3.27E+02	3.17E+02	3.38E+02
	RW-GWO	**3.15E+02**	**2.77E-01**	**3.15E+02**	**3.14E+02**	**3.15E+02**
24	GWO	2.00E+02	**7.27E-04**	2.00E+02	2.00E+02	2.00E+02
	RW-GWO	**2.00E+02**	3.04E-03	**2.00E+02**	**2.00E+02**	**2.00E+02**
25	GWO	2.11E+02	2.04E+00	2.11E+02	2.07E+02	2.15E+02
	RW-GWO	**2.04E+02**	**1.18E+00**	**2.05E+02**	**2.02E+02**	**2.07E+02**
26	GWO	1.00E+02	9.62E-02	1.00E+02	1.00E+02	1.01E+02
	RW-GWO	**1.00E+02**	**7.36E-02**	**1.00E+02**	**1.00E+02**	**1.00E+02**
27	GWO	4.33E+02	1.82E+01	4.30E+02	4.03E+02	4.86E+02
	RW-GWO	**4.09E+02**	**6.09E+00**	**4.08E+02**	**4.03E+02**	**4.40E+02**
28	GWO	9.14E+02	6.63E+01	9.07E+02	7.93E+02	1.12E+03
	RW-GWO	**4.34E+02**	**8.45E+00**	**4.35E+02**	**4.16E+02**	**4.53E+02**
29	GWO	2.90E+05	1.57E+06	3.28E+04	4.98E+03	1.12E+07
	RW-GWO	**2.14E+02**	**2.37E+00**	**2.14E+02**	**2.08E+02**	**2.19E+02**
30	GWO	2.98E+04	1.57E+04	2.71E+04	8.09E+03	6.80E+04
	RW-GWO	**6.69E+02**	**2.14E+02**	**6.62E+02**	**2.76E+02**	**1.13E+03**

5.1. *Wilcoxon test*

In order to test the statistical validity of the results obtained from RW-GWO, non-parametric pair wise Wilcoxon test has been applied. The test has been conducted for 5% level of significance. The results are presented in Table 5. The following criteria has been applied for concluding the results

1. The observed difference is highly significant if p-value ≤ 0.01.
2. The observed difference is significant if p-value < 0.05.
3. Both the algorithms are statistically same if p-value $= 0.05$.
4. The observed difference is marginally significant if p-value ≤ 0.10.
5. The observed difference is not significant if p-value > 0.10.

On the basis of this criteria of analysis the comparative results between GWO and RW-GWO are shown in table as A+, B+, C+, D+ when RW-GWO is better with high significant than, better with significant than, marginally better than, or not significantly better than GWO respectively, and A, B, C and D when GWO is better with high significant than, better with significant than, marginally better than, or not significantly better than RW-GWO respectively. E shows that both the algorithms are statistically same. Here marginally better means that if

Table 3
Comparison of average error in objective function value for 30 dimension of CEC 2014 test problems. Better results are highlighted in bold.

Problem	GSA	CS	LX-BBO	B-BBO	SOS	RW-GWO
1	1.57E+07	**2.29E+05**	1.01E+07	6.50E+06	1.95E+07	8.02E+06
2	8.89E+03	**1.39E+02**	5.34E+04	2.36E+04	3.61E+09	2.23E+05
3	7.32E+04	1.05E+04	1.64E+04	6.04E+03	2.38E+04	**3.16E+02**
4	3.68E+02	7.07E+01	9.99E+01	1.03E+02	4.14E+02	**3.41E+01**
5	2.00E+01	2.02E+01	**3.06E+00**	3.74E+00	2.07E+01	2.05E+01
6	2.88E+01	2.48E+01	1.69E+01	2.00E+01	2.48E+01	**9.84E+00**
7	**1.99E-04**	7.45E-02	1.76E-01	7.82E-02	3.38E+01	2.53E-01
8	1.40E+02	7.12E+01	5.53E+01	**4.71E-01**	8.10E+01	4.38E+01
9	1.65E+02	1.78E+02	7.66E+01	9.11E+01	1.73E+02	**6.33E+01**
10	3.35E+03	2.02E+03	1.26E+04	6.69E+03	2.40E+03	**9.61E+02**
11	4.06E+03	4.49E+03	1.23E+04	6.72E+03	4.48E+03	**2.68E+03**
12	9.10E-02	8.11E-01	**1.11E-02**	**1.11E-02**	7.37E-01	5.45E-01
13	4.03E-01	4.17E-01	6.55E-01	6.75E-01	6.85E-01	**2.80E-01**
14	**2.30E-01**	5.18E-01	6.20E-01	3.93E-01	8.39E+00	4.23E-01
15	1.26E+01	1.32E+01	1.55E+01	1.88E+01	2.56E+02	**8.81E+00**
16	1.48E+01	1.23E+01	1.08E+01	1.07E+01	1.20E+01	**1.03E+01**
17	7.29E+05	**1.24E+05**	1.46E+06	1.28E+06	5.59E+06	5.71E+05
18	**3.86E+02**	1.42E+03	2.90E+03	8.22E+02	4.86E+05	6.52E+03
19	1.57E+02	1.18E+01	5.19E+03	7.81E+03	4.07E+01	**1.14E+01**
20	8.24E+04	**1.36E+02**	2.61E+04	1.63E+04	1.59E+04	6.27E+02
21	1.79E+05	**1.67E+03**	1.11E+06	1.23E+06	7.86E+05	2.58E+05
22	9.51E+02	3.11E+02	1.88E+03	**1.68E+02**	5.45E+02	2.08E+02
23	**2.00E+02**	3.44E+02	4.11E+02	3.43E+02	3.42E+02	3.15E+02
24	2.01E+02	2.21E+02	1.48E+04	3.41E+04	2.36E+02	**2.00E+02**
25	**2.00E+02**	2.09E+02	5.29E+02	6.54E+02	2.15E+02	2.04E+02
26	1.69E+02	1.01E+02	2.13E+00	3.64E+01	1.01E+02	**1.00E+02**
27	7.69E+02	4.18E+02	**1.96E+02**	3.05E+02	4.96E+02	4.09E+02
28	7.65E+02	9.13E+02	1.94E+03	2.12E+03	1.32E+03	**4.34E+02**
29	**2.00E+02**	2.01E+02	1.98E+07	3.09E+07	2.16E+04	2.14E+02
30	2.31E+04	3.59E+03	6.96E+06	1.38E+07	3.73E+04	**6.69E+02**

Table 4
Algorithm complexity (in seconds).

		D = 10	D = 30
T0		0.14	0.14
T1		0.78	1.38
GWO	$\widehat{T2}$	2.27	3.60
	$(\widehat{T2}-T1)/T0$	10.83	16.13
RW-GWO	$\widehat{T2}$	2.15	3.52
	$(\widehat{T2}-T1)/T0$	9.95	15.55

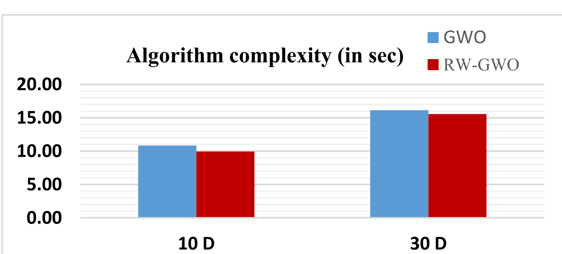

Fig. 4. Algorithm complexity of GWO and RW-GWO for dimension 10 and 30.

between algorithm A and algorithm B observed difference is marginally significant, then better algorithm is denoted as marginally significant algorithm. From Table 5 it can be seen that for the dimension 10, in 22 out of 30 functions and for the dimension 30, in 26 out of 30 functions RW-GWO is better significantly compared to GWO.

5.2. Performance index analysis

In order to investigate the performance of RW-GWO and GWO the criteria of Performance Index as given in Deep and Thakur [33] is used in this paper. The PI can be used to test the relative performance of different algorithms successfully. The PI of any algorithm can be calculated by the following formula

$$PI_k = \frac{1}{n_p}\sum_{j=1}^{n_p}\left(k_1\,\alpha_1^j + k_2\,\alpha_2^j\right)$$

where $\alpha_1^j = \frac{Mf^j}{Af^j}$ and $\alpha_2^j = \frac{Mt^j}{At^j}$ for $j = 1, 2, \ldots, n_p$.

Mf^j = Minimum of average error value obtained by all the algorithm in obtaining the solution for the j^{th} function.
Af^j = Average error value obtained by an algorithm in obtaining the solution for j^{th} function.
Mt^j = Minimum of average time used by all the algorithm in obtaining the solution for the j^{th} function.
At^j = Average time used by an algorithm in obtaining the solution for the j^{th} function.
n_p = Total number of problems.

k_1 and k_2 ($k_1 + k_2 = 1$ and $0 \le k_1, k_2 \le 1$) are the weights assigned to the error value and computational time respectively. From the definition of PI it can be seen that PI is a function of k_1 and k_2. If we consider $k_1 = w$ then $k_2 = 1 - w$. PI is evaluated for $w = 0, 0.2, 0.4, 0.6, 0.8$ and 1. The obtained results are plotted in Fig. 7, which clearly shows that when $k_1 = 0$, then PI of RW-GWO is lower than GWO. PI of RW-GWO is better than GWO for $k_1 \ge 0.2$. Hence it can be concluded from analysis of PI that GWO has a better PI than RW-GWO when user is focused on computational time only and when user wants an algorithm best in sense of error and computational time also then in that case RW-GWO is recommended over GWO.

5.3. Computational complexity

Any metaheuristic algorithm should have less computational complexity, so that the real life optimization problems can be solved in less computational efforts. Therefore analyzing the computational complexity of any search algorithm is very crucial to observe its effectiveness on optimization problems.

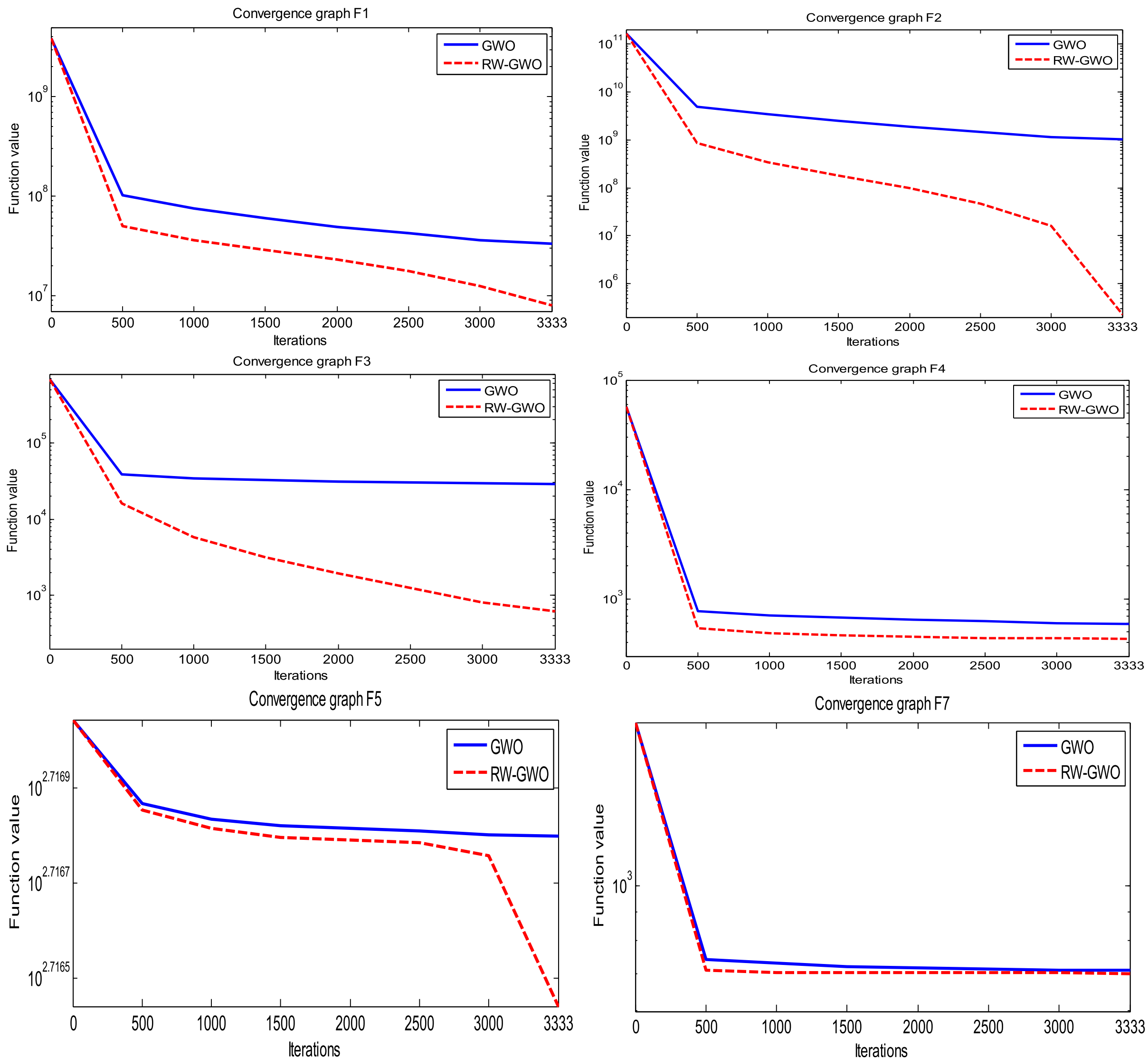

Fig. 5. Convergence curves for 30D of certain CEC 2014 functions.

The step wise computational complexity in GWO and RW-GWO in terms of worst case of computational time can be calculated as follows-

In the first step at the time of initializing the population of wolves both the algorithms have complexity of $O(N)$ where N is the size of the wolf population. Both the algorithms have complexity $O(1)$ for initialization of parameters b, μ and c. The leaders in both the algorithms can be selected in $O(N)$ computational efforts as for finding the leaders linear search is performed.

The main computational steps for both the algorithms starts after the while loop. In original GWO position is updated for each wolf, so the computational complexity is of $O(N)$ for this step. As this process of updating the wolf runs until the maximum number of iterations are not reached, so the computational complexity is multiplied by maximum number of iterations say T. In the last step of the algorithm, alpha wolf from the population of wolves can be selected in $O(N)$ computational efforts. After summing up all the complexities as discussed above, the total computation complexity in GWO is obtained as $O(T{\cdot}N)$.

In RW-GWO inside while loop updating the leaders by using random walk takes $O(1)$ computational time and remaining omega wolves can be updated in $O(N)$ computational time and in the end of the algorithm alpha wolf from the population of wolves can be selected in $O(N)$ computational efforts. After summing up all the complexities as discussed above, the total computation complexity in RW-GWO is obtained as $O(T{\cdot}N)$.

Therefore from the worst complexity point of view both the algorithms are same and the complexity for both the algorithms is found to be $O(T{\cdot}N)$. where T is the maximum number of iterations (termination criteria in the algorithm GWO and RW-GWO) and N is the size of wolf pack.

6. Applications

In order to analyze the performance of GWO and RW-GWO on real life application problems, we implemented these algorithms on very familiar problems- Parameter estimation for frequency-modulated (FM) problem (f_1) which is unconstrained optimization problem, optimal capacity of gas production facilities (f_2) and pressure vessel design (f_3).. Let the most general optimization problem

$$\begin{aligned}&\boldsymbol{Min} \quad f(x)\\ &\boldsymbol{s.t.} \quad g_i(x) \leq 0, \quad i = 1, 2, \ldots, m\\ &h_j(x) = 0, \quad j = m+1, m+2, \ldots, p.\end{aligned}$$

where g_i are inequality constraints and $h_j(x)$ are equality constraints for the problem.

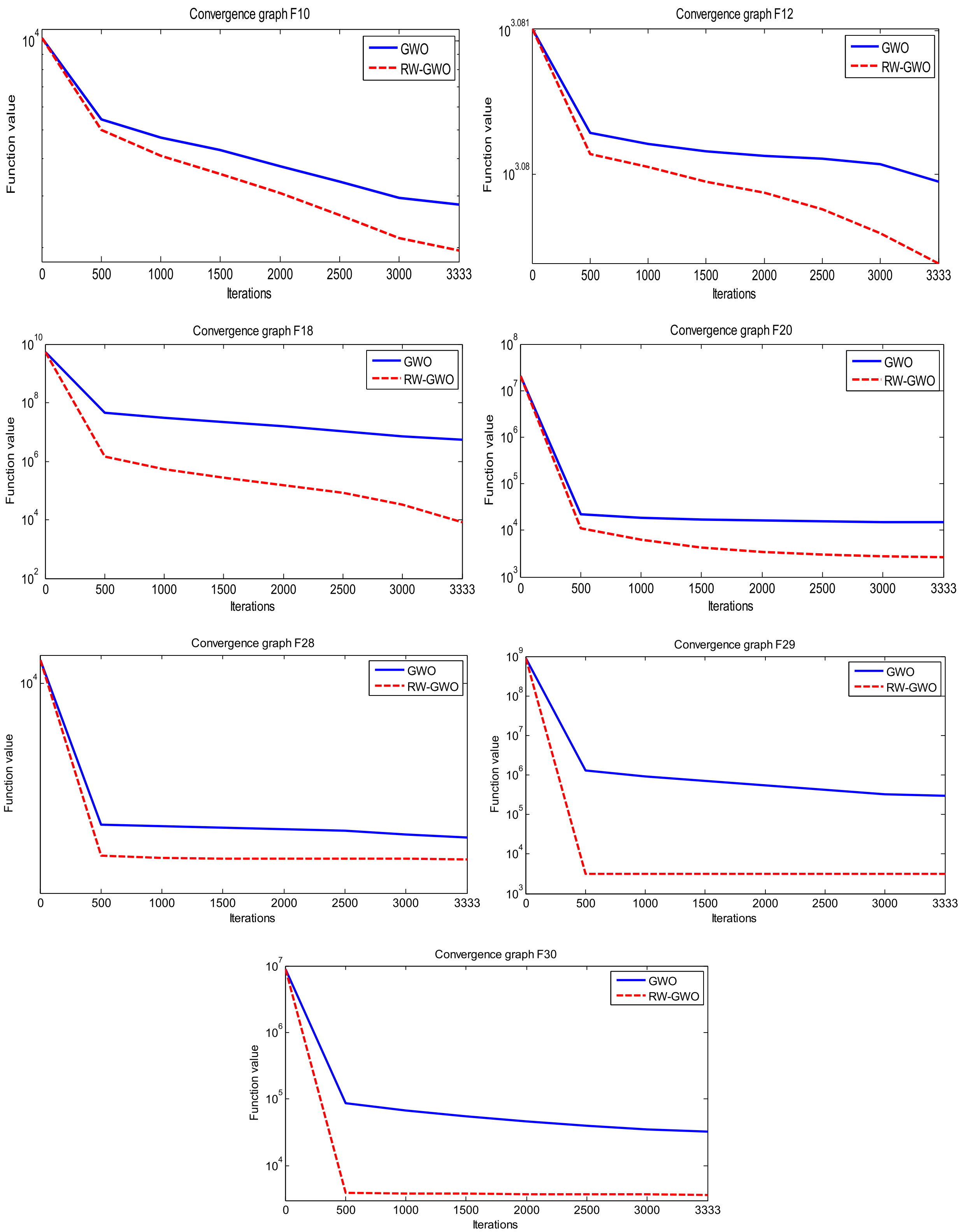

Fig. 6. Convergence curves for 30D of certain CEC 2014 functions.

To deal with constraints we applied Deb's technique [34] in GWO and RW-GWO between current and updated positions, which is one of most popular technique for dealing with constraints. This technique follows three feasibility rules

i. Feasible solution is preferred over infeasible solution.
ii. If both solutions are feasible then one having better fitness is preferred.
iii. If both the solutions are infeasible then one having less constraint violation is preferred.

Table 5
Results of Wilcoxon signed rank test on CEC 2014 functions.

Dimension = 10			Dimension = 30		
Function	p-value	conclusion	Function	p-value	conclusion
F1	1.40E-09	A+	F1	6.15E-10	A+
F2	4.20E-01	D	F2	5.15E-10	A+
F3	5.15E-10	A+	F3	5.15E-10	A+
F4	7.35E-10	A+	F4	5.15E-10	A+
F5	9.14E-09	A+	F5	3.60E-10	A+
F6	6.76E-02	C+	F6	1.28E-02	B+
F7	1.49E-09	A+	F7	5.15E-10	A+
F8	3.14E-09	A+	F8	2.36E-09	A+
F9	8.90E-03	A+	F9	2.22E-05	A+
F10	4.94E-09	A+	F10	1.87E-09	A+
F11	5.96E-02	C	F11	5.46E-02	C+
F12	4.81E-07	A+	F12	6.92E-09	A+
F13	1.36E-04	A+	F13	3.24E-07	A+
F14	1.75E-06	A+	F14	3.70E-03	A+
F15	7.74E-09	A+	F15	8.49E-06	A+
F16	6.62E-02	D+	F16	9.24E-05	A+
F17	2.47E-04	A+	F17	7.57E-01	D
F18	4.80E-03	A	F18	6.03E-08	A+
F19	3.82E-04	A+	F19	3.73E-09	A+
F20	5.15E-10	A+	F20	5.15E-10	A+
F21	8.69E-08	A+	F21	2.09E-01	D
F22	1.69E-05	A+	F22	4.05E-02	B+
F23	5.15E-10	A+	F23	4.89E-10	A+
F24	3.57E-05	A+	F24	2.27E-11	A
F25	2.09E-01	D+	F25	4.76E-10	A+
F26	1.20E-01	D+	F26	5.82E-06	A+
F27	2.90E-01	D+	F27	7.13E-09	A+
F28	2.50E-09	A+	F28	5.14E-10	A+
F29	5.15E-10	A+	F29	5.15E-10	A+
F30	5.15E-10	A+	F30	5.15E-10	A+

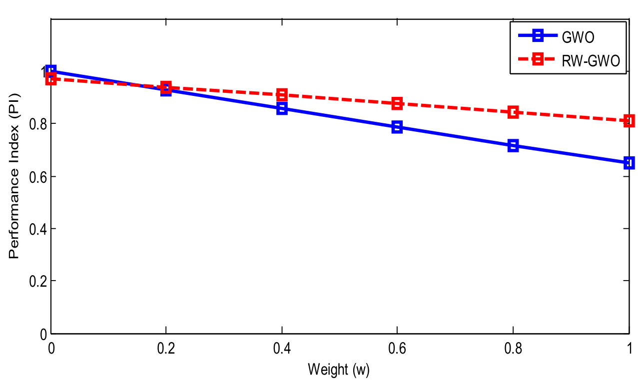

Fig. 7. Performance Index of graph of RW-GWO and GWO.

In the third rule, the constraint violation for any solution x can be calculated as

$$viol(x) = \sum_{i=1}^{m} G_i(x) + \sum_{j=m+1}^{p} H_j(x)$$

where

$$G_i(x) = \begin{cases} g_i(x) & if \quad g_i(x) > 0 \\ 0 & otherwise \end{cases}$$

$$H_j(x) = \begin{cases} |h_j(x)| & if \ |h_j(x)| - \in > 0 \\ 0 & otherwise \end{cases}$$

and $\in$ is predefined tolerance parameter.

6.1. Unconstrained real life problems

6.1.1. Gear train design

The objective of this problem is to find the optimal number of tooth for four gears of a train to minimize the gear ratio [35]. As all the decision variables should be integer so we round off these variables to the nearest integer.

The mathematical form of the problem is as follows -

$$Min \quad f_1(x) = \left(\frac{1}{6.931} - \frac{x_1 x_3}{x_2 x_4}\right)^2, \qquad x = (x_1, x_2, x_3, x_4) = (\eta_A, \eta_B, \eta_C, \eta_D)$$

$$s.t. \qquad 12 \le x_1, x_2, x_3, x_4 \le 60.$$

The best solution obtained by RW-GWO and other algorithms are presented in Table 6.

The table shows that the algorithm RW-GWO solves the problem with better objective function value compared to the original GWO algorithm. Although the obtained solution with RW-GWO is same as algorithm CS [36], MBA [37] but with more function evaluation cost. Also this problem has been solved by Artificial bee colony (ABC) [38], Genetic algorithm (GA) [39] and ALM [40]. The results from the table verified the better performance of RW-GWO compared to the other reported algorithms.

6.1.2. Parameter estimation for frequency-modulated (FM) [35].

FM sound wave synthesis has an important role in some modern music systems. The parameter optimization of an FM synthesizer is a six dimensional optimization problem where the vector to be optimized is $x = (a_1, \omega_1, a_2, \omega_2, a_3, \omega_3)$ of sound wave given by eq. (17). The problem is to generate a sound (1) similar to target (2). The problem is highly complex multimodal one having strong epistasis with minimum value $f(x) = 0$. The expression for estimated and target sound is as follows

$$y(t) = a_1 \sin(\omega_1 t\theta + a_2 \sin(\omega_2 t\theta + a_3 \sin(\omega_3 t\theta))) \tag{17}$$

$$y_0(t) = (1.0)\sin((5.0)t\theta - (1.5)\sin((4.8)\ t\theta + (2.0)\sin((4.9)t\theta))) \tag{18}$$

where $\theta = 2\pi/100$ and the parameter range is [-6.4 6.35]. The fitness function is defined as

$$\boldsymbol{Min\, f_2} = \sum_{t=0}^{100} (y(t) - y_0(t))^2$$

The obtained results in terms of minimum, maximum, average and standard deviation of the error from GWO, RW-GWO and some other algorithms are reported in Table 7. For all the algorithms maximum number of function evaluations is set to be 30,000 and 30 runs of each algorithm is performed on the problem. From the Table it can be observed that the proposed algorithm RW-GWO performs better for FM problem compare to other reported algorithms.

Table 6
Results on gear train design problem. Better result is highlighted in bold.

Algorithm	Optimal solution				f_{min}	Max function evaluations
	x_1	x_2	x_3	x_4		
RW-GWO	19	43	16	49	$\mathbf{2.7009 \times 10^{-12}}$	**200**
GWO	17	45	21	55	1.3616×10^{-9}	200
CS	19	43	16	49	2.7009×10^{-12}	5000
MBA	19	43	16	49	2.7009×10^{-12}	1120
GA	14	33	17	50	1.3620×10^{-9}	N/A
ABC	19	44	16	49	1.0742×10^{-5}	40,000
ALM	15	33	13	41	2.407×10^{-8}	N/A

Table 7
Comparison of results on frequency-modulated problem. Better results are highlighted in bold.

Algorithm	Error in objective function			
	Minimum	Maximum	Average	STD
RW-GWO	**0.006001**	**8.87117**	18.7917	6.948362
GWO	1.9311	20.0383	25.1633	5.91743
CLPSO [41,42]	0.007	14.08	**3.82**	23.53
CPSOH [42,43]	3.45	42.52	27.08	60.61
TRIBES-D [42,44]	2.22	22.24	14.68	**4.57**
G-CMA-ES [42,45]	3.326	55.09	38.75	16.77

6.2. Constrained real life problems

6.2.1. A tension/compression string design problem

The objective this problem is to minimize the weight $f_3(x)$ of a tension/compression spring subject to the constraints on shear stress, deflection, and surge frequency, outside diameter and on design variables. The decision variables are – wire diameter x_1, mean coil diameter x_2 and number of active coils x_3. The mathematical formulation of the problem is given by

$$\textbf{Min}\quad f_3(x) = (x_3+2)x_2x_1^2$$
$$\textbf{s. t.}\quad g_1(x) = 1 - \frac{x_2^3x_3}{71785x_1^4} \le 0$$
$$g_2(x) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \le 0$$
$$g_3(x) = 1 - \frac{140.45x_1}{x_2^2x_3} \le 0$$
$$g_4(x) = \frac{x_1+x_2}{1.5} - 1 \le 0$$
$$0.05 \le x_1 \le 2,\ 0.25 \le x_2 \le 1.30,\ 2 \le x_3 \le 15.$$

This problem has been solved by metaheuristics GA [46], GSA [29] and numerical approaches constraint correction [47] and mathematical optimization [48] and the obtained results by these techniques with GWO and RW-GWO are presented in Table 8. For all the algorithms maximum number of function evaluations is set to be 30,000 and 30 runs of each algorithm is performed on the problem. In GWO we got only 47% success in feasibility out of 30 runs while in RW-GWO success is 100% in sense of feasibility. The table verified the better performance of RW-GWO compared to other techniques.

6.2.2. Pressure vessel design [43].

This problem is very popular in literature and this problem consists of four decision variables with four inequality constraints. In this problem the variables R and L are continuous while T_s and T_h are integer multiple of 0.0625. The objective of this problem is to find the least fabrication cost to obtain a design for pressure vessel. The mathematical formulation of the problem is given by

Table 8
Comparision of results on compression string design problem. Better result is highlighted in bold.

Algorithms	Optimal decision vector			Optimum value
	x_1	x_2	x_3	
RW-GWO	0.05167	0.35613	11.33056	**0.012674**
GWO	0.05155	0.35322	11.50405	0.012675
GA (Coello)	0.05148	0.35166	11.63220	0.012705
GSA	0.05028	0.32368	13.52541	0.012702
Constraint correction (Arora)	0.05000	0.31590	14.25000	0.012833
Mathematical Optimization (Belegunda)	0.05340	0.39918	9.18540	0.012730

$$\textbf{Min}\quad f_4(\boldsymbol{x}) = 0.6224\,x_1x_3x_4 + 1.7781\,x_2x_3^2 + 3.1661\,x_1^2x_4 + 19.84\,x_1^2x_3$$
$$x = (x_1,\ x_2, x_3, x_4) = (\boldsymbol{T_s}, \boldsymbol{T_h}, \boldsymbol{R}, \boldsymbol{L})$$
$$\textbf{s.t.}\quad g_1(x) = -x_1 + 0.0193\,x_3 \le 0.$$
$$g_2(x) = -x_2 + 0.00954\,x_3 \le 0$$
$$g_3(x) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0$$
$$g_4(x) = x_4 - 240 \le 0$$
$$0 \le x_1,\ x_2 \le 99$$
$$10 \le x_3,\ x_4 \le 200$$

The structure of pressure vessel is optimized with GWO and RW-GWO and the obtained results are presented in Table 9. This problem has been solved by GA (Coello and Montes) [49] using 2, 00,000 function evaluations. Same functions evaluations are used in GWO and RW-GWO to solve this problem. GSA [29], PSO (He and Wang) [50], GA (Coello) [46], GA (Deb) [51], ES [52], Lagrangian Multiplier [40] and Branch and Bound (Sandgen) [53] algorithms are employed on pressure vessel problem. The obtained results from these algorithms are also reported in Table 9. The results shows that the obtained cost is minimum in RW-GWO compared to other reported algorithms.

7. Conclusion and future scope

This work proposes a modified version of GWO by incorporating Random walk for leading wolves in GWO to optimize the search ability for prey by wolf pack. Set of 30 standard benchmark problems of IEEE CEC 2014 have taken to check the robustness of the proposed RW-GWO algorithm. The performance of proposed algorithm is compared with basic GWO and other metaheuristic algorithms that shows that RW-GWO is very competitive with the other algorithms. All the analysis of the results is done on the criteria laid by CEC 2014. From the analysis of the results done in this article it is recommended that RW-GWO outperforms GWO in terms of error value as well as computational time. Also RW-GWO provides a significantly better results compared to GWO and other state of art algorithms for sample of application problems described in the paper.

This proposed algorithm is giving a new direction toward the improvement of leader's search ability such that real word applications problems can be solved. Similarly other improvement for leading wolves can also be proposed to solve unconstrained optimization problems. Also in future RW-GWO can be developed for solving different types of optimization problems like constrained optimization problems, integer programming problems etc.

Table 9
Comparision of results on pressure vessel design problem. Better result is highlighted in bold.

Algorithms	Optimal decision vector				Optimum value
	x_1	x_2	x_3	x_4	
RW-GWO	0.81250	0.43750	42.09840	176.63784	**6059.736**
GWO	0.87500	0.43750	44.98072	144.10807	6136.660
GA (Coello and Montes)	0.81250	0.43750	42.09740	176.65405	6059.946
GSA	1.12500	0.62500	55.98866	84.45420	8538.836
PSO (He and Wang)	0.81250	0.43750	42.09127	176.74650	6061.078
GA (Coello)	0.81250	0.43450	40.32390	20.00000	6288.745
GA (Deb)	0.93750	0.50000	48.32900	112.67900	6410.381
ES	0.81250	0.43750	42.09809	176.64052	6059.746
Lagrangian Multiplier	1.12500	0.62500	58.29100	43.69000	7198.043
Branch and Bound (Sandgen)	1.12500	0.62500	47.70000	117.70100	8129.104

Acknowledgement

The first author would like to thank Ministry of Human Resources, Government of India for funding this research. Grant No. MHR-02-41-113-429.

References

[1] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proc. Sixth Int. Symp. Micro Mach. Hum. Sci., 1995, pp. 39–43.
[2] M. Dorigo, M. Birattari, T. St, Ant Colony Optimization, 2006.
[3] X. Yang, Firefly Algorithm, Lévy Flights and Global Optimization, 2010.
[4] D. Karaboga, B. Basturk, A Powerful and Efficient Algorithm for Numerical Function Optimization : Artificial Bee Colony (ABC) Algorithm, 2007, pp. 459–471.
[5] J.C. Bansal, H. Sharma, S. Singh, J. Maurice, Spider Monkey Optimization Algorithm for Numerical Optimization, 2014, pp. 31–47.
[6] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Software 95 (2016) 51–67.
[7] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Software 69 (2014) 46–61.
[8] D. Wolpert, No Free Lunch Theorems for Search, Most, 1995, pp. 1–38.
[9] A. Madadi, M.M. Motlagh, Optimal control of DC motor using grey wolf optimizer algorithm, TJEAS J. -2014-4-04/373-379 4 (4) (2014) 373–379.
[10] N. Muangkote, K. Sunat, S. Chiewchanwattana, An improved grey wolf optimizer or training q-Gaussian Radial Basis Functional-link nets, in: Computer Science and Engineering Conference (ICSEC), 2014 International, IEEE, 2014, pp. 209–214.
[11] X. Song, L. Tang, S. Zhao, X. Zhang, L. Li, J. Huang, W. Cai, Grey Wolf Optimizer for parameter estimation in surface waves, Soil Dynam. Earthq. Eng. 75 (2015) 147–157.
[12] A.A. El-Fergany, H.M. Hasanien, Single and multi-objective optimal power flow using grey wolf optimizer and differential evolution algorithms, Elec. Power Compon. Syst. 43 (13) (2015) 1548–1559.
[13] S. Mirjalili, How effective is the Grey Wolf optimizer in training multi-layer perceptrons, Appl. Intell. 43 (1) (2015) 150–161.
[14] M.R. Shakarami, I.F. Davoudkhani, Wide-area power system stabilizer design based on Grey Wolf Optimization algorithm considering the time delay, Elec. Power Syst. Res. 133 (2016) 149–159.
[15] D. Guha, P.K. Roy, S. Banerjee, Load frequency control of interconnected power system using grey wolf optimization, Swarm Evolut. Comput. 27 (2016) 97–115.
[16] V.K. Kamboj, S.K. Bath, J.S. Dhillon, Solution of non-convex economic load dispatch problem using Grey Wolf Optimizer, Neural Comput. Appl. 27 (5) (2016) 1301–1316.
[17] S. Zhang, Y. Zhou, Z. Li, W. Pan, Grey wolf optimizer for unmanned combat aerial vehicle path planning, Adv. Eng. Software 99 (2016) 121–136.
[18] T. Jayabarathi, T. Raghunathan, B.R. Adarsh, P.N. Suganthan, Economic dispatch using hybrid grey wolf optimizer, Energy 111 (2016) 630–641.
[19] B. Yang, X. Zhang, T. Yu, H. Shu, Z. Fang, Grouped grey wolf optimizer for maximum power point tracking of doubly-fed induction generator based wind turbine, Energy Convers. Manag. 133 (2017) 427–443.
[20] C. Lu, L. Gao, X. Li, S. Xiao, A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real-world welding industry, Eng. Appl. Artif. Intell. 57 (2017) 61–79.
[21] M.A. Tawhid, A.F. Ali, A Hybrid grey wolf optimizer and genetic algorithm for minimizing potential energy function, Memet. Comput. (2017) 1–13.
[22] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary grey wolf optimization approaches for feature selection, Neurocomputing 172 (2016) 371–381.
[23] L. Rodríguez, O. Castillo, J. Soria, P. Melin, F. Valdez, C.I. Gonzalez, G.E. Martinez, J. Soto, A fuzzy hierarchical operator in the grey wolf optimizer algorithm, Appl. Soft Comput. 57 (2017) 315–328.
[24] A.A. Heidari, P. Pahlavani, An efficient modified grey wolf optimizer with Lévy flight for optimization tasks, Appl. Soft Comput. 60 (2017) 115–134.
[25] S. Mirjalili, S. Saremi, S.M. Mirjalili, L.S. Coelho, Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization, Expert Syst. Appl. 47 (2016) 106–119.
[26] C. Muro, R. Escobedo, L. Spector, R.P. Coppinger, Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations, Behav. Process. 88 (2011) 192–197.
[27] X.S. Yang, Nature-inspired Metaheuristic Algorithms, Luniver Press, 2010.
[28] J.J. Liang, B.Y. Qu, P.N. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2014. Special Session and Competition on Single Objective RealParameter Numerical Optimization, Technical Report 11, Computational Intelligence Laboratory, Zhengzhou University, Singapore, December 2013. Zhengzhou China and Technical Report, Nanyang Technological University.
[29] E. Rashedi, Esmat, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (13) (2009) 2232–2248.
[30] X.S. Yang, S. Deb, Cuckoo search via Lévy flights, in: Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on, IEEE, 2009, pp. 210–214.
[31] V. Garg, K. Deep, Performance of Laplacian Biogeography-Based Optimization Algorithm on CEC 2014 continuous optimization benchmarks and camera calibration problem, Swarm Evolut. Comput. 27 (2016) 132–144.
[32] M.Y. Cheng, D. Prayogo, Symbiotic organisms search: a new metaheuristic optimization algorithm, Comput. Struct. 139 (2014) 98–112.
[33] K. Deep, M. Thakur, A new mutation operator for real coded genetic algorithms, Appl. Math. Comput. 193 (2007) 211–230.
[34] K. Deb, An efficient constraint handling method for genetic algorithms 186 (2000) 311–338.
[35] S. Das, P.N. Suganthan, Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems, Electronics (2011) 1–42.
[36] A.H. Gandomi, X.S. Yang, A.H. Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, Eng. Comput. 29 (1) (2013) 17–35.
[37] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems, Appl. Soft Comput. 13 (5) (2013) 2592–2612.
[38] T.K. Sharma, M. Pant, V.P. Singh, Improved Local Search in Artificial Bee Colony Using Golden Section Search, arXiv preprint arXiv: 1210.6128, 2012.
[39] K. Deb, M. Goyal, A combined genetic adaptive search (GeneAS) for engineering design, Comput. Sci. Inf. 26 (1996) 30–45.
[40] B.K. Kannan, S.N. Kramer, An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design, J. Mech. Des. 116 (1994) 405–411.
[41] C. Li, S. Yang, T.T. Nguyen, A self-learning particle swarm optimizer for global optimization problems, IEEE Trans. Syst. Man, Cybern., Part B (Cybern.) 42 (3) (2012) 627–646.
[42] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (3) (2006) 281–295.
[43] F.v.d. Bergh, A.P. Engelbrecht, A cooperative approach to particle swarm optimization, IEEE Trans. Evol. Comput. 8 (3) (2004) 225–239.
[44] C. Maurice, 2007. [Online]. Available: http://www.particleswarm.info/Programs.html.
[45] A. Auger, N. Hansen, A restart CMA evolution strategy with increasing population size, in: Evolutionary Computation, 2005. The 2005 IEEE Congress on, vol. 2, IEEE, 2005, pp. 1769–1776.
[46] C.A.C. Coello, Use of a self-adaptive penalty approach for engineering optimization problems, Comput. Ind. 41 (2) (2000) 113–127.
[47] J.S. Arora, Introduction to Optimum Design, Academic Press, 2004.
[48] A.D. Belegundu, J.S. Arora, A study of mathematical programming methods for structural optimization. Part I: Theory, Int. J. Numer. Meth. Eng. 21 (9) (1985) 1583–1599.
[49] C.A.C. Coello, E.M. Montes, Constraint-handling in genetic algorithms through the use of dominance-based tournament selection, Adv. Eng. Inf. 16 (3) (2002) 193–203.
[50] Q. He, L. Wang, An effective co-evolutionary particle swarm optimization for constrained engineering design problems, Eng. Appl. Artif. Intell. 20 (1) (2007) 89–99.
[51] K. Deb, GeneAS: a robust optimal design technique for mechanical component design, in: Evolutionary Algorithms in Engineering Applications, Springer Berlin Heidelberg, 1997, pp. 497–514.
[52] E.M. Montes, C.A.C. Coello, An empirical study about the usefulness of evolution strategies to solve constrained optimization problems, Int. J. Gen. Syst. 37 (4) (2008) 443–473.
[53] E. Sandgren, Nonlinear integer and discrete programming in mechanical design, in: Proceedings of the ASME Design Technology Conference, 1988, pp. 95–105.