Accelerated grey wolf optimiser for continuous optimisation problems

Shubham Gupta* and Kusum Deep

Department of Mathematics,
Indian Institute of Technology Roorkee,
Roorkee – 247667, India
Email: sgupta@ma.iitr.ac.in
Email: kusumfma@iitr.ac.in
*Corresponding author

Seyedali Mirjalili

Institute for Integrated and Intelligent Systems,
Griffith University,
Nathan Campus,
Brisbane, QLD 4111, Australia
Email: mirjalili@griffithuni.edu.au

Abstract: Grey wolf optimiser (GWO) is a relatively simple and efficient nature-inspired optimisation algorithm which has shown its competitive performance compared to other population-based meta-heuristics. This algorithm drives the solutions towards some of the best solutions obtained so far using a unique mathematical model, which is inspired from leadership behaviour of grey wolves in nature. To combat the issue of premature convergence and local optima stagnation, an enhanced version of GWO is proposed in this paper. The proposed algorithm is named accelerated grey wolf optimiser (A-GWO). In A-GWO, novel modified search equations are developed that enhances the exploratory behaviour of wolves at later generations, and the exploitation of search space is also improved in the whole search process. To validate the performance of the proposed algorithm, set of 23 well-known classical benchmark problems are used. The results and comparison through various metrics show the reliability and efficiency of the A-GWO.

Keywords: optimisation; swarm intelligence; grey wolf optimiser; GWO; engineering optimisation test problems.

Reference to this paper should be made as follows: Gupta, S., Deep, K. and Mirjalili, S. (2020) 'Accelerated grey wolf optimiser for continuous optimisation problems', *Int. J. Swarm Intelligence*, Vol. 5, No. 1, pp.22–59.

Biographical notes: Shubham Gupta receives his BSc (Honours) in Mathematics from Banaras Hindu University (Varanasi, India) in 2013 and MSc in Mathematics and Computing from Indian Institute of Technology Guwahati (Assam, India) in 2015. He is currently a Research Scholar in Mathematics Department of Indian Institute of Technology Roorkee (Roorkee, India). His area of research includes nature inspired algorithms specially artificial bee colony, sine cosine algorithm, grey wolf optimiser, particle swarm optimisation, and genetic algorithm, etc.

Kusum Deep is a Professor with the Department of Mathematics, Indian Institute of Technology Roorkee, India. She pursued her BSc and MSc degrees from Centre for Advanced Studies, Panjab University, Chandigarh. She earned her PhD from IIT Roorkee. She carried out research at Loughborough University, UK, under an International Post Doctorate Bursary funded by Commission of European Communities, Brussels. She is a senior member of Operations Research Society of India, IEEE, Computer Society of India, Indian Mathematical Society and Indian Society of Industrial Mathematics. Her research interests include numerical optimisation, evolutionary algorithms, genetic algorithms, particle swarm optimisation, etc.

Seyedali Mirjalili is a Lecturer at Griffith University and internationally recognised for his advances in swarm intelligence (SI) and optimisation, including the first set of SI techniques from a synthetic intelligence standpoint – a radical departure from how natural systems are typically understood – and a systematic design framework to reliably benchmark, evaluate, and propose computationally cheap robust optimisation algorithms. He has published over 80 journal articles, many in high-impact journals with over 7,000 citations in total with an H-index of 29 and G-index of 84. From Google Scholar metrics, he is globally the third most cited researcher in engineering optimisation and robust optimisation. He is serving an Associate Editor of *Advances in Engineering Software* and the *Journal of Algorithms*.

1 Introduction

Optimisation is an effort to find the best solution for an optimisation problem within some given circumstances. To tackle the difficulties of real-world optimisation problems, scientists and practitioners have developed many mathematical models and techniques in the literature. The applications of such techniques can be found in diverse fields. For many reasons such as the restricted domain of the nature of the problem, complexity in terms of solving the problem, the derivative-based mechanism, deterministic techniques are not suitable for those real-world problems. In all those cases probabilistic population-based techniques have shown their potential as they do not require any special characteristic of problem like differentiability, convexity, etc. Some of the most well-regarded algorithms in this class are: genetic algorithm (GA) (Holland, 1992), particle swarm optimisation (PSO) (Eberhart and Kennedy, 1995), differential evolution (DE) (Storn and Price, 1997), gravitational search algorithm (GSA) (Rashedi et al., 2009), evolution strategy (ES) (Diouane et al., 2015), artificial bee colony (ABC) algorithm (Karaboga and Basturk, 2007), biogeography-based optimisation (BBO) (Simon, 2008), harmony search (HS) algorithm (Geem et al., 2001), ant colony optimisation (ACO) (Dorigo and Birattari, 2011), and sine cosine algorithm (SCA) (Mirjalili, 2016), etc. Swarm intelligence is a very popular branch in the field of meta-heuristics where the algorithm are designed from the simulation of the intelligent and collective behaviour of social creatures such as ants, bees, cats, cuckoos, monkeys, wolves, etc. Similar to other stochastic population-based algorithms swarm intelligence-based algorithms start the search process using a random set of solutions (population), and they iteratively improve the quality of those solutions using direct or random search to find the optimal solution for the problem. PSO (Eberhart and Kennedy, 1995), ACO (Dorigo and Birattari, 2011), ABC algorithm (Karaboga and Basturk, 2007), cuckoo search (CS) algorithm (Yang and Deb, 2009), moth-flame optimisation algorithm (MFO) (Mirjalili, 2015), spider monkey optimisation (Bansal et al., 2014), and grey wolf optimiser (GWO) algorithm (Mirjalili et al., 2014) are some of the methods in this category.

Despite the large number of algorithms and improvements in this field, one might ask if there is a need for another improvement. The no free lunch theorem (NFL) (Wolpert and Macready, 1995) is a well-known theorem in the field of meta-heuristic algorithms which states that there is no single optimisation algorithm available to solve all optimisation problems. This means that for a given set of problems one optimisation algorithm may perform better than other search algorithms, while for the other set of problems this particular algorithm may not perform better than other algorithms.

In the present work, we focus on GWO algorithm proposed by Mirjalili et al. (2014) as a simple and powerful optimisation approach to solve continuous optimisation problems. Recently, GWO algorithm has become quite popular and largely applied to problems in diverse fields. The following paragraphs presents some of the applications.

In Zhang et al. (2016), GWO has been used to solve unmanned combat aerial vehicle path planning. In Jayakumar et al. (2016), GWO was used to solve combined heat and power economic dispatch. Combined economic emission dispatch problems are solved in Song et al. (2014) by using GWO algorithm. For sizing of multiple distributed generation in the distribution system classical GWO has been implemented in Sultana et al. (2016). To train the q-Gaussian radial basis an improved version of GWO has been proposed by Muangkote et al. (2014). To solve optimal reactive power dispatch problem, Sulaiman et al. (2015) have used the classical GWO algorithm. In Kamboj et al. (2016), to solve the problem of non-convex economic load dispatch GWO has been used.

Despite the successful application of GWO in different fields, several works criticised that this algorithm suffers from degraded performance when applying to highly multi-modal problems. Several attempts have been made to alleviate this drawbacks as follows: Gupta and Deep (2018a, 2018b) proposed a novel random walk GWO based on Cauchy distribution to solve engineering application problems. In Heidari and Pahlavani (2017), flight strategy was used in classical GWO to enhance the search capability of grey wolves. In Rodríguez et al. (2017), fuzzy hierarchical operators have been used in place of centroid equation of GWO. In Long et al. (2018), the search equation of GWO was modified using the random wolf to enhance the exploration ability of grey wolves and solve the real-world optimisation problems efficiently. In Saremi et al. (2015), Evolutionary population dynamics was integrated in GWO to speed up the convergence and to improve the performance of classical GWO. In Kohli and Arora (2018), a chaotic GWO has been proposed to accelerate the convergence speed. In Mittal et al. (2016), the modified version of GWO was proposed to maintain a proper balance between exploration and exploitation. In Kumar and Kumar (2018), astrophysics-inspired GWO is proposed to solve the engineering optimisation problems.

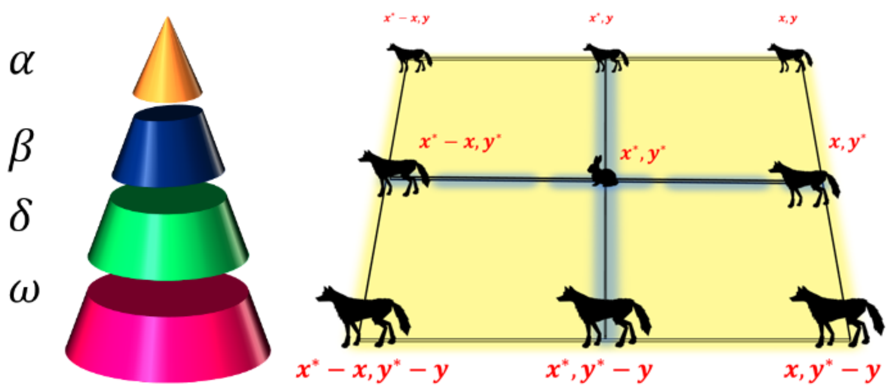
As other meta-heuristic algorithms such as GA, PSO, and ABC, The GWO algorithm also faces the problem of stagnation of solution vectors (wolf pack) in local optima and premature convergence. Therefore, local optima entrapment and premature convergence are two major issues in the GWO algorithm. The goal of this paper is to combat these issues by proposing an accelerated grey wolf optimiser (A-GWO), in which an acceleration factor and a uniform distribution are used to boost both exploration and exploitation.

The remainder of the paper is organised as follows: Section 2 provides a brief description of classical GWO algorithm. In Section 3 an enhanced variant of GWO is proposed. In Section 4 numerical results on standard test problems with varying dimension are presented. In Section 5 five engineering application problems are used to evaluate the performance of the proposed algorithm. Finally, Section 6 concludes the paper and suggests some future ideas.

2 GWO algorithm

The GWO algorithm (Mirjalili et al., 2014) mimics the social behaviour and dominant leadership in a grey wolf pack. Grey wolves are known as apex predators and they lie on the top of food chain. Their pack is about 5–11 wolves and within the pack, to maintain a discipline and leadership behaviour their pack is divided into four type of wolf [as shown in Figure 1(a)] – alpha (the dominant wolf of the pack which is responsible for all the important and major decision regarding the various activities of wolves within pack), beta (these are the subordinate wolf to the alpha and acts as a leading wolf if the alpha wolf passes away), delta (the sentinels, caretakers and elder wolves belongs to this category) and omega (all the remaining wolves of the pack that are allowed to eat in the end).

Figure 1 (a) Leadership hierarchy in a wolf pack (b) Various search positions of grey wolf in 2D (see online version for colours)



(a) (b)

The wolf alpha, beta and delta are known as leading wolves of the pack and the hunting process of prey is totally dependent on these wolves. According to Muro et al. (2011), grey wolves perform their hunting process in three steps as follows:

1 tracking and chasing a prey

2 encircling the prey

3 attacking the prey.

The following subsections presents the mathematical model developed by Mirjalili et al. (2014).

2.1 Mathematical model of GWO

Leadership behaviour

To simulate the leadership characteristic of the grey wolf pack in an optimisation problem, the fittest solution of an optimisation problem is considered as alpha, second and third best solutions (considering a merit, cost, or fitness function) are assumed as beta and delta solutions respectively, and the remaining solutions are considered as omega solutions. To find a prey, all the omega wolves performs their search process with the guidance of alpha, beta and delta wolves.

Encircling prey

When the prey location is identified by the grey wolves, they perform their encircling and hunting strategies around prey. To encircle a prey, the following equations are used in GWO:

$$\overrightarrow{X_{t+1}} = \overrightarrow{X_{p,t}} - \vec{A} \times \vec{D} \tag{1}$$

$$\vec{D} = \left| \vec{C} \cdot \overrightarrow{X_{p,t}} - \overrightarrow{X_t} \right| \tag{2}$$

$$\vec{A} = 2 \times \overrightarrow{b_t} \times \overrightarrow{r_1} - \overrightarrow{b_t} \tag{3}$$

$$\vec{C} = 2 \times \overrightarrow{r_2} \tag{4}$$

where $\overrightarrow{X_t}$ and $\overrightarrow{X_{t+1}}$ represent the states of a grey wolf at iteration t and $t + 1$ respectively. $\overrightarrow{X_{p,t}}$ represents the state of prey at iteration t. $\vec{D}$ represents the difference vector. $\vec{A}$ and $\vec{C}$ are coefficient vectors that maintained the exploration and exploitation during the search process. $\overrightarrow{r_1}$ and $\overrightarrow{r_2}$ represents the random vectors which are uniformly distributed and whose components lies in the interval (0, 1). $\vec{b}$ is a vector that decreases linearly from 2 to 0 as the iteration of algorithm proceeds. Note that each element in these vector represent and is used for one dimension (variable) of the problem. The vector $\vec{b}$ can be formulated mathematically as

$$\vec{b}_t = 2 - 2 \times \left(\frac{t}{T} \right) \tag{5}$$

where T is the maximum number of iteartions, and t indicates the current iteration.

Hunting

Grey wolves have the ability to identify the position of prey and surround them. Usually, the hunting activity is guided by alpha wolf, but beta and delta wolves also participate in hunting process infrequently by assuming that beta and delta wolves have enough useful information about prey. Therefore, these leading hunters are saved in each iteration to perform the search process by other hunting wolves. Since these leaders have enough knowledge regarding the prey, the prey position can be approximated by these leaders with the help of equation (1) in each iteration as

$$\overrightarrow{D_\alpha} = \left|\overrightarrow{C_\alpha} \times \overrightarrow{X_\alpha(t)} - \overrightarrow{X(t)}\right| \tag{6}$$

$$\overrightarrow{D_\beta} = \left|\overrightarrow{C_\beta} \times \overrightarrow{X_\beta(t)} - \overrightarrow{X(t)}\right| \tag{7}$$

$$\overrightarrow{D_\delta} = \left|\overrightarrow{C_\delta} \times \overrightarrow{X_\delta(t)} - \overrightarrow{X(t)}\right| \tag{8}$$

Here $\overrightarrow{X_\alpha}$, $\overrightarrow{X_\beta}$ and $\overrightarrow{X_\delta}$ are the vectors that represent alpha, beta and delta wolves at t^{th} iteration respectively. $\overrightarrow{C_\alpha}$, $\overrightarrow{C_\beta}$ and $\overrightarrow{C_\delta}$ are random vectors as defined in equation (5). After calculating the difference vectors $\overrightarrow{D_\alpha}$, $\overrightarrow{D_\beta}$ and $\overrightarrow{D_\delta}$ the updated states of grey wolves for $(t + 1)^{\text{th}}$ iteration can be calculated as follows:

$$\overrightarrow{x_1} = \overrightarrow{X_\alpha(t)} - \overrightarrow{A_\alpha} \times \overrightarrow{D_\alpha} \tag{9}$$

$$\overrightarrow{x_2} = \overrightarrow{X_\beta(t)} - \overrightarrow{A_\beta} \times \overrightarrow{D_\beta} \tag{10}$$

$$\overrightarrow{x_3} = \overrightarrow{X_\delta(t)} - \overrightarrow{A_\delta} \times \overrightarrow{D_\delta} \tag{11}$$

$$\overrightarrow{X(t+1)} = \left(\overrightarrow{x_1} + \overrightarrow{x_2} + \overrightarrow{x_3}\right)/3 \tag{12}$$

where the values of $\overrightarrow{A_\alpha}$, $\overrightarrow{A_\beta}$ and $\overrightarrow{A_\delta}$ can be calculated using equation (3). As such, the alpha, beta and delta estimate the position of prey and omega wolves update their positions around the prey. Therefore, on repeating the process of encircling and hunting in mathematical form the optima of the problem can be determined. A pictorial representation of grey wolf position and its possible updated positions using GWO search equations is shown in Figure 1(b). In this figure (x^*, y^*) represents the position of prey.

2.2 Exploration (search for prey) and exploitation (attack on prey) in GWO

The exploration and exploitation are two conflicting operators in meta-heuristics. In the phase of exploration, new regions of a search space are discovered, and in exploitation phase, the solutions previously discovered search regions are improved. Therefore, an algorithm should be capable of addressing and balancing these two important operators to estimate the global optima of the problem.

In GWO algorithm the vectors $\vec{A}$ and $\vec{C}$ have been introduced in this regard. When $|\vec{A}| < 1$ and/or $\vec{C} < 1$, search regions are exploited and this situation represents the behaviour of chasing a prey by grey wolves and when $|\vec{A}| > 1$ and/or $\vec{C} > 1$ the new search regions are discovered in order to explore the search space and to prevent from stagnation in local solutions. This simulates the attacking prey behaviour by grey wolves. As in GWO algorithm, after the half of maximum number of iterations, $|\vec{A}| < 1$ [as $\vec{b} < 1$, and $\vec{A} \in (-\vec{b}, \vec{b})$] then, in this case, the exploration of a search space is performed by the vector $\vec{C}$. The balance between the operator's exploration and exploitation is maintained in GWO algorithm with the decreasing nature of the vector $\vec{b}$.

3 Proposed accelerated GWO

In classical GWO algorithm, the search of prey depends on the leaders of the pack i.e., a guided search is used in GWO with the help of alpha, beta and delta wolves. But when these leaders get trapped in a local optimum, then in that case wolf population (wolf pack) converges to local optima and thus the problem of insufficient diversity within the search space occurs during the search process. Therefore, in the situation where the leading wolves fail to guide the other wolves (omega) towards an optimal direction and wolf pack gets trapped in a region of local solutions, a perturbation is needed for the leaders alpha, beta and delta that enhances the exploration ability of wolves and tries to provide an optimum direction towards the optima. To do so, this paper introduces an acceleration factor which provides an explorative behaviour to the grey wolves so that the stagnation in local optima can be avoided and the convergence rate can speed up. This proposed algorithm is named as A-GWO.

As discussed above, when the value of A decreases and reaches at the situation where $|A| < 1$ the exploitation is performed in GWO due to coefficient A, and in this case, only the vector C performs exploration of search space. Our initial investigation showed that the parameter C is insufficient to provide a suitable explorative strength of a search space after half of the maximum number of iterations when $|A| < 1$. To alleviate this drawback, an acceleration factor is introduced after half of the maximum number of iterations.

In the proposed A-GWO algorithm modified proposed encircling equations can be stated as:

$$\overrightarrow{X_{t+1}} = \vec{w} \times \overrightarrow{X_{p,t}} - \vec{A} \cdot \vec{D} \tag{13}$$

$$\vec{D} = \left| \vec{C} \times \overrightarrow{X_{p,t}} - \overrightarrow{X_t} \right| \tag{14}$$

$$\vec{A} = 2 \times \vec{b} \times \vec{r_1} - \vec{b} \tag{15}$$

$$\vec{c} = \overrightarrow{r_2} \tag{16}$$

Here the vector $\vec{w}$ that is introduced to accelerate the convergence rate and to explore the promising regions is called an acceleration factor. From the experimental analysis, it can be observed that smaller value of $\vec{w}$ performs a global search and large value of $\vec{w}$ performs a local search. To move in the directions around the prey, the value of $\vec{w}$ should be negative too. In the present work, a suitable value that maintains a balance between local and global search is chosen in the interval [–0.2, 0.2] as large values may diverge the population from the optima. The value of $\vec{w}$ used in the proposed algorithm is although randomly but uniformly distributed within this interval. Also, the exploitation of GWO algorithm is increased by restricting the vector $\vec{C}$ in a unit circle i.e., in the whole process of search the vector $\vec{C}$ is used to simulate the behaviour of attacking prey by grey wolves. Thus in a complete search process, for the exploration of a search space, only the vector $\vec{A}$ is restricted and the vector $\vec{C}$ is for exploitation. All the other parameters are kept same as in classical GWO algorithm. The new hunting equations proposed in A-GWO are

$$\overrightarrow{X_1} = \overrightarrow{w_\alpha} \times \overrightarrow{X_\alpha(t)} - \overrightarrow{A_\alpha} \times \overrightarrow{D_\alpha} \tag{17}$$

$$\overrightarrow{X_2} = \overrightarrow{w_\beta} \times \overrightarrow{X_\beta(t)} - \overrightarrow{A_\beta} \times \overrightarrow{D_\beta} \tag{18}$$

$$\overrightarrow{X_3} = \overrightarrow{w_\delta} \times \overrightarrow{X_\delta(t)} - \overrightarrow{A_\delta} \times \overrightarrow{D_\delta} \tag{19}$$

$$\overrightarrow{X(t+1)} = \left(\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}\right)/3 \tag{20}$$

Here the values $\overrightarrow{w_\alpha}$, $\overrightarrow{w_\beta}$ and $\overrightarrow{w_\delta}$ are uniformly distributed within the interval [–0.2, 0.2], $\overrightarrow{D_\alpha}$, $\overrightarrow{D_\beta}$ and $\overrightarrow{D_\delta}$ are obtained from equations (6), (7) and (8) and $\overrightarrow{X_\alpha}$, $\overrightarrow{X_\beta}$ and $\overrightarrow{X_\delta}$ are the positions of alpha, beta and delta wolf respectively. In the proposed algorithm A-GWO, the value of $\vec{w}$ can be fixed as 1 upto half of the maximum number of iterations.

When the value of $\vec{b}$ decreased from 1 to 0, which happens at every iteration when $t >$ (maximum number of iterations/2), then the coefficient vector $\vec{A}$ lies in the interval (–1, 1) and in this situation the exploitation phase (attacking prey) of GWO algorithm occurs. In this situation the vector $\vec{C}$ is not able to explore the search space properly. The acceleration factor $\vec{w}$ helps in that situation to enhance the exploration ability of grey wolves and then the promising regions that are never visited earlier can be explored to find the better solution. This acceleration factor also helps to speed up the convergence rate as the diversity is increased after half of the maximum number of iterations with the help of acceleration factor w in A-GWO algorithm. The vector $\vec{C}$ enhances the exploitation strength of wolves in the whole search process and provides a local search.

The pseudo code of the proposed algorithm is presented in Figure 2.

Figure 2 Pseudo code of accelerated GWO

```
Initialize population of wolves
Initialize the parameters b⃗ and maximum number of iteration T
Evaluate the fitness of each grey wolf
Select the leaders as
Xα − the fittest solution
Xα − second best solution
Xα − third best solution
initialize the iteration count t = 0
   while t < T
        If t ≤ T/2
            update each wolf position with the help of equations (1) to (12)
            upadate the leaders and parameter b⃗
        else
            update each wolf position with the help of equations (13) to (20)
            update the leaders and parameter b⃗
        end if
    Evaluate the fitness of each grey wolf
    update the leaders Xα,Xβ and Xδ to proceed with the search process
    t = t + 1
    end while
    return Xα
```

4 Experimental results and analysis

4.1 Benchmark problems and parameter setting

Before applying any meta-heuristic optimisation algorithm on real-world application problem, it should be tested on benchmark test problems to analyse its performance. Therefore, in this subsection, a set of 23 standard well-known benchmark test problems with 30, 50 and 100 dimensions have been taken of different categories (unimodal and multimodal) from various sources (Mirjalili et al., 2014; Long et al., 2018; Gao et al., 2013, 2014) to verify the performance of the proposed improved version of GWO. The test problems with their global optima have been reported in Table 1. 2-D shapes of some test problems are shown in Figure 3.

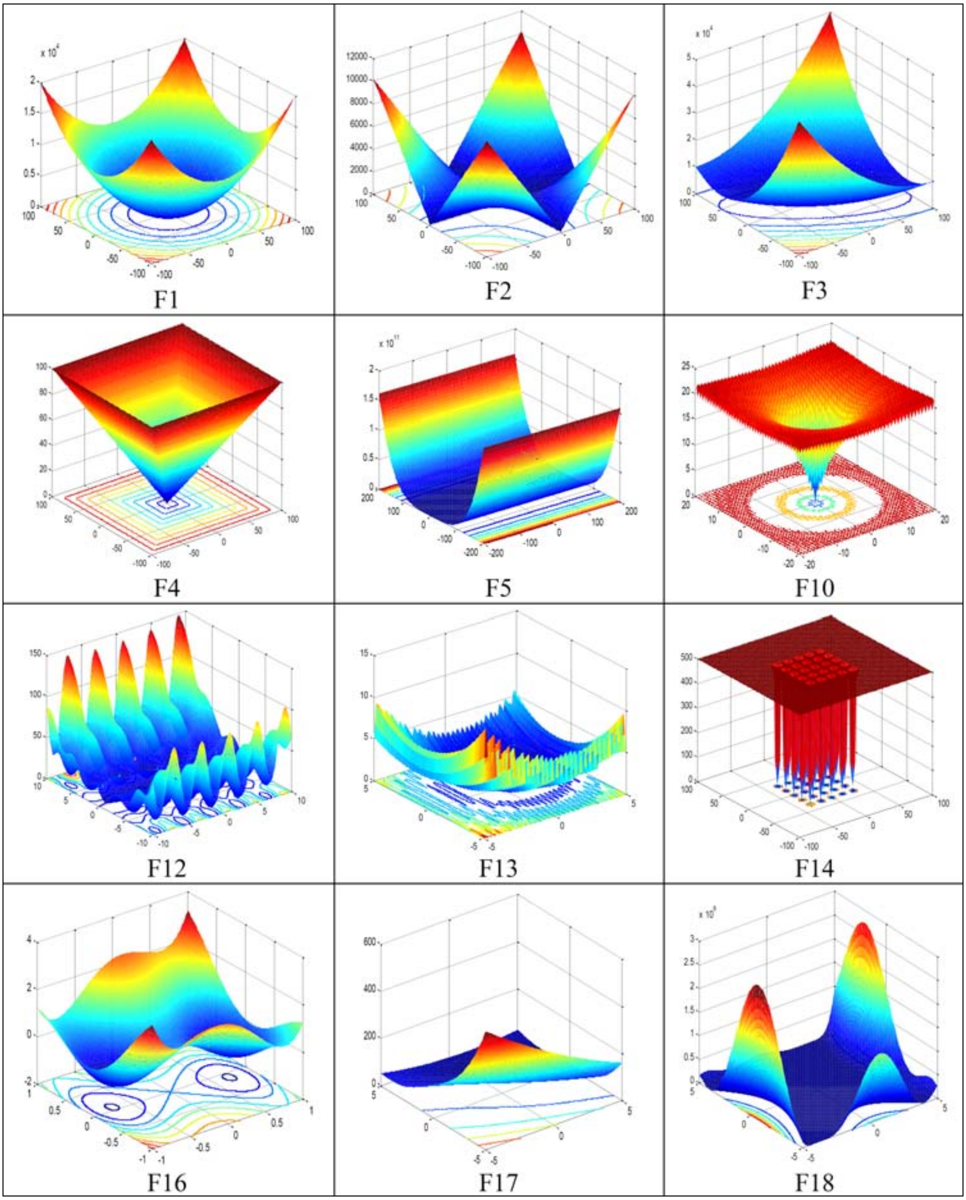
Table 1 Classification of benchmark problems

Test problem	*Dimension*	*Range of search space*	F_{min}
Unimodal benchmark problems			
$F1(x)=\sum_{i=1}^{d} x_i^2$	30, 50, 100	[–100, 100]	0
$F2(x)=\sum_{i=1}^{d} ix_i^2$	30, 50, 100	[–10, 10]	0
$F3(x)=\sum_{i=1}^{d} \lvert x_i^2\rvert+\prod_{i=1}^{d}\lvert x_i\rvert$	30, 50, 100	[–10, 10]	0
$F4(x)=\sum_{i=1}^{d}\left(\sum_{j-1}^{d} x_i\right)^2$	30, 50, 100	[–100, 100]	0
$F5(x)=\max_i\{\lvert x_i\rvert, 1\le i\le d\}$	30, 50, 100	[–100, 100]	0
$F6(x)=\sum_{i=1}^{d-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	30, 50, 100	[–30, 30]	0
$F7(x)=\sum_{i=1}^{d}([x_i+0.5])^2$	30, 50, 100	[–100, 100]	0
$F8(x)=\sum_{i=1}^{d} i\cdot x_i^4$	30, 50, 100	[–1.28, 1.28]	0
$F9(x)=\sum_{i=1}^{d} i\cdot x_i^4+rand[0,1)$	30, 50, 100	[–1.28, 1.28]	0
$F10(x)=\sum_{i=1}^{d}\lvert x_i\rvert^{i+1}$	30, 50, 100	[–1, 1]	0
Multimodal benchmark problems			
$F11(x)=\sum_{i=1}^{d} -x_i\sin(\sqrt{\lvert x_i\rvert})$	30, 50, 100	[–500, 500]	–418.9829 × d
$F12(x)=\sum_{i=1}^{d}[x_i^2-10\cos(2\pi x_i)+10]$	30, 50, 100	[–5.12, 5.12]	0
$F13(x)=\sum_{i=1}^{d} -20\exp\left(-0.2\sqrt{\frac{1}{2}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)+20+e\right)$	30, 50, 100	[–32, 32]	0

Table 1 Classification of benchmark problems (continued)

Test problem	*Dimension*	*Range of search space*	F_{min}
Multimodal benchmark problems			
$F14(x)=\frac{1}{4}\times10^{-3}\sum_{i=1}^{d}x_i^2-\prod_{i=1}^{d}\cos(x_i/\sqrt{i})+1$	30, 50, 100	[–600, 600]	0
$F15(x)=\frac{\pi}{d}\left\{10\sin(\pi y_1)+\sum_{i=1}^{d-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]+(y_n-1)^2\right\}+\sum_{i=1}^{d}u(x_i,10,100,4)$ $y_i=\frac{x_i+5}{4}$ $u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m & \text{if } x_i>a\\ k(-x_i-a)^m & \text{if } x_i<-a\\ 0 & \text{otherwise}\end{cases}$	30, 50, 100	[–50, 50]	0
$F16(x)=0.1\times\left\{\sin^2(3\pi x_1)+\sum_{i=1}^{d}(x_i-1)^2[1+\sin^2(1+3\pi x_1)]+(x_d-1)^2[1+\sin^2]\right\}+\sum_{i=1}^{d}u(x_i,5,100,4)$	30, 50, 100	[–50, 50]	0
$F17(x)=\sum_{i=1}^{d}\lvert x_i\sin(x_i)+0.1\,x_i\rvert$	30, 50, 100	[–10, 10]	0
$F18(x)=0.1d-\left(0.1\sum_{i=1}^{d}\cos(5\pi x_i)-\sum_{i=1}^{d}x_i^2\right)$	30, 50, 100	[–1, 1]	0
$F19(x)=\sum_{i=1}^{d}x_i^2+\left(\sum_{i=1}^{d}0.5\,i\,x_i\right)^2+\left(\sum_{i=1}^{d}0.5\,i\,x_i\right)^4$	30, 50, 100	[–5, 10]	0
$F20(x)=\sum_{i=1}^{d}(10^6)^{(i-1)/(d-1)}x_i^2$	30, 50, 100	[–100, 100]	0
$F21(x)=(-1)^{d+1}\prod_{i=1}^{d}\cos(x_i)\times e^{\left[-\sum_{i=1}^{d}(x_i-\pi)^2\right]}$	30, 50, 100	[–100, 100]	0
$F22(x)=1-\cos\left(2\pi\left(\sqrt{\sum_{i=1}^{d}x_i^2}\right)\right)+0.1\times\sqrt{\sum_{i=1}^{d}x_i^2}$	30, 50, 100	[–100, 100]	0
$F23(x)=0.5+\frac{\sin^2\left(\sqrt{\sum_{i=1}^{d}x_i^2}-0.5\right)}{\left(1+0.001\left(\sqrt{\sum_{i=1}^{d}x_i^2x_i^2}\right)\right)^2}$	30, 50, 100	[–100, 100]	0

Figure 3 2-D shape of some selected benchmarks (see online version for colours)



The proposed algorithm is implemented on these test problems having different dimensions with the same parameter setting that is used in classical GWO algorithm (Mirjalili et al., 2014) i.e., population of 30 wolves is considered and the termination criteria is taken as 500 iterations. As each wolf is evaluated once in each iteration, the total number of function evaluations used for each problem is 30 × 500=15,000.

For the numerical experiments, MATLAB 2010a with Core i5, 2.3 GHz, 4 GB RAM system has been used. For each of the test problem, the algorithm has been executed 30 times and each time the different seed has been used for the generation of random numbers. The results obtained are reported in Tables 2 to 7. In these tables various statistical measures such as average, worst, minimum, standard deviation and median values of the objective function value obtained by classical GWO and proposed A-GWO on the above mentioned test problems are reported. In these tables, the better results are highlighted in **bold**.

Table 2 Best, average, median, worst and standard deviation of objective function values for unimodal test problems of 30 dimension

Function	*Algorithm*	*Best*	*Average*	*Median*	*Worst*	*STD*
F1	GWO	2.10E-29	7.17E-28	2.79E-28	6.90E-27	1.33E-27
	A-GWO	**0**	**0**	**0**	**0**	**0**
F2	GWO	1.95E-30	9.73E-29	5.61E-29	5.97E-28	1.28E-28
	A-GWO	**0**	**0**	**0**	**0**	**0**
F3	GWO	9.89E-18	5.78E-17	7.57E-17	2.66E-16	5.92E-17
	A-GWO	**6.43E-226**	**7.42E-209**	**2.31E-220**	**2.16E-207**	**0**
F4	GWO	1.90E-08	3.84E-06	5.77E-05	1.32E-03	2.41E-04
	A-GWO	**0**	**0**	**0**	**0**	**0**
F5	GWO	6.14E-08	6.19E-07	9.50E-07	9.74E-06	1.71E-06
	A-GWO	**2.99E-212**	**4.05E-205**	**9.68E-208**	**6.73E-204**	**0**
F6	GWO	**2.61E+01**	**2.71E+01**	**2.71E+01**	**2.88E+01**	8.40E-01
	A-GWO	2.87E+01	2.89E+01	2.89E+01	2.90E+01	**9.38E-02**
F7	GWO	**6.21E-05**	7.57E-01	7.71E-01	**1.51E+00**	**4.38E-01**
	A-GWO	1.18E-01	**6.74E-01**	**2.13E-01**	3.38E+00	9.23E-01
F8	GWO	1.42E-55	2.94E-51	4.79E-52	3.50E-50	7.55E-51
	A-GWO	**0**	**0**	**0**	**0**	**0**
F9	GWO	4.19E-04	1.70E-03	1.84E-03	4.67E-03	9.66E-04
	A-GWO	**3.56E-06**	**1.22E-04**	**7.68E-05**	**6.69E-04**	**1.45E-04**
F10	GWO	4.86E-84	7.77E-64	4.74E-74	1.64E-62	3.09E-63
	A-GWO	**0**	**0**	**0**	**0**	**0**

Table 3 Best, average, median, worst and standard deviation of objective function values for multimodal test problems of 30 dimension

Function	*Algorithm*	*Best*	*Average*	*Median*	*Worst*	*STD*
F11	GWO	**–7.46E+03**	**–6.10E+03**	**–6.12E+03**	**–4.48E+03**	**7.65E+02**
	A-GWO	–4.50E+03	–3.51E+03	–3.51E+03	–2.61E+03	4.18E+02
F12	GWO	5.68E-14	1.01E+00	2.98E+00	1.04E+01	3.57E+00
	A-GWO	**0**	**0**	**0**	**0**	**0**
F13	GWO	7.55E-14	1.00E-13	1.05E-13	1.50E-13	1.84E-14
	A-GWO	**8.88E-16**	**8.88E-16**	**8.88E-16**	**8.88E-16**	**4.01E-31**
F14	GWO	0	0	5.40E-03	3.27E-02	1.00E-02
	A-GWO	**0**	**0**	**0**	**0**	**0**
F15	GWO	6.68E-03	**4.49E-02**	6.49E-02	**5.51E-01**	**9.44E-02**
	A-GWO	**4.20E-03**	1.39E-01	7.20E-03	1.32E+00	3.56E-01

Table 3 Best, average, median, worst and standard deviation of objective function values for multimodal test problems of 30 dimension (continued)

Function	*Algorithm*	*Best*	*Average*	*Median*	*Worst*	*STD*
F16	GWO	3.21E-01	**7.24E-01**	7.22E-01	**1.10E+00**	**1.85E-01**
	A-GWO	**8.38E-02**	8.13E-01	**1.65E-01**	2.65E+00	9.31E-01
F17	GWO	9.36E-17	7.13E-04	6.19E-04	2.28E-03	7.10E-04
	A-GWO	**1.48E-220**	**1.70E-208**	**9.01E-212**	**4.44E-207**	**0.00E+00**
F18	GWO	0	0	0	0	0
	A-GWO	**0**	**0**	**0**	**0**	**0**
F19	GWO	4.21E-10	1.06E-07	3.49E-08	7.19E-07	1.81E-07
	A-GWO	**0**	**0**	**0**	**0**	**0**
F20	GWO	5.49E-26	3.07E-24	1.25E-24	2.4E-23	5.42E-24
	A-GWO	**0**	**0**	**0**	**0**	**0**
F21	GWO	0	0	0	0	0
	A-GWO	**0**	**0**	**0**	**0**	**0**
F22	GWO	9.99E-02	1.83E-01	2.00E-01	2.00E-01	3.79E-02
	A-GWO	**0**	**1.02E-81**	**2.8E-155**	**3.07E-80**	**5.61E-81**
F23	GWO	9.72E-03	3.86E-02	3.72E-02	7.82E-02	1.58E-02
	A-GWO	**0**	**5.95E-03**	**0**	**3.72E-02**	**1.28E-02**

Table 4 Best, average, median, worst and standard deviation of objective function values for unimodal test problems of 50 dimension

Function	*Algorithm*	*Best*	*Average*	*Median*	*Worst*	*STD*
F1	GWO	7.66E-21	5.11E-20	4.05E-20	2.52E-19	4.68E-20
	A-GWO	**0**	**0**	**0**	**0**	**0**
F2	GWO	9.54E-22	1.61E-20	8.09E-21	1.79E-19	3.30E-20
	A-GWO	**0**	**0**	**0**	**0**	**0**
F3	GWO	7.45E-13	2.78E-12	2.72E-12	6.81E-12	1.37E-12
	A-GWO	**6.65E-220**	**2.38E-210**	**2.75E-214**	**6.90E-209**	**0**
F4	GWO	1.07E-03	1.39E-01	3.93E-02	1.15E+00	2.51E-01
	A-GWO	**0**	**0**	**0**	**0**	**0**
F5	GWO	4.00E-05	5.08E-04	3.48E-04	2.18E-03	4.96E-04
	A-GWO	**6.51E-211**	**2.56E-200**	**4.01E-204**	**7.64E-199**	0
F6	GWO	**4.61E+01**	**4.75E+01**	**4.71E+01**	4.95E+01	7.80E-01
	A-GWO	4.85E+01	4.88E+01	4.89E+01	**4.90E+01**	**1.74E-01**
F7	GWO	1.43E+00	2.78E+00	2.65E+00	**4.51E+00**	**7.59E-01**
	A-GWO	**3.52E-01**	**2.08E+00**	**1.58E+00**	7.12E+00	1.81E+00
F8	GWO	1.30E-40	6.55E-38	5.91E-39	8.26E-37	1.62E-37
	A-GWO	**0**	**0**	**0**	**0**	**0**
F9	GWO	8.64E-04	3.01E-03	2.36E-03	6.81E-03	1.55E-03
	A-GWO	**1.68E-05**	**1.69E-04**	**1.22E-04**	**5.87E-04**	**1.26E-04**
F10	GWO	4.86E-84	7.77E-64	4.74E-74	1.64E-62	3.09E-63
	A-GWO	**0**	**0**	**0**	**0**	**0**

Table 5 Best, average, median, worst and standard deviation of objective function values for multimodal test problems of 50 dimension

Function	*Algorithm*	*Best*	*Average*	*Median*	*Worst*	*STD*
F11	GWO	**–1.18E+04**	**–9.19E+03**	**–9.23E+03**	**–7.14E+03**	1.09E+03
	A-GWO	–5.64E+03	–4.61E+03	–4.64E+03	–3.55E+03	**5.21E+02**
F12	GWO	1.71E-13	5.70E+00	4.24E+00	1.68E+01	5.19E+00
	A-GWO	**0**	**0**	**0**	**0**	**0**
F13	GWO	9.39E-12	3.71E-11	3.19E-11	7.22E-11	1.71E-11
	A-GWO	**8.88E-16**	**8.88E-16**	**8.88E-16**	**8.88E-16**	**4.01E-31**
F14	GWO	0	2.39E-03	0	2.72E-02	6.54E-03
	A-GWO	**0**	**0**	**0**	**0**	**0**
F15	GWO	4.81E-02	**1.23E-01**	1.11E-01	**3.39E-01**	**5.91E-02**
	A-GWO	**8.82E-03**	3.66E-01	**1.59E-02**	1.30E+00	5.13E-01
F16	GWO	1.35E+00	2.13E+00	2.15E+00	**2.70E+00**	**3.26E-01**
	A-GWO	**1.81E-01**	**1.51E+00**	**4.03E-01**	4.84E+00	1.57E+00
F17	GWO	9.49E-13	8.19E-04	3.43E-04	2.88E-03	9.00E-04
	A-GWO	**1.13E-214**	**1.29E-205**	**6.11E-210**	**3.76E-204**	**0**
F18	GWO	0	3.26E-16	0	3.55E-15	7.18E-16
	A-GWO	**0**	**0**	**0**	**0**	**0**
F19	GWO	5.93E-04	4.16E-02	2.25E-02	3.41E-01	6.34E-02
	A-GWO	**0**	**0**	**0**	**0**	**0**
F20	GWO	1.01E-17	1.48E-16	8.30E-17	5.92E-16	1.69E-16
	A-GWO	**0**	**0**	**0**	**0**	**0**
F21	GWO	0	0	0	0	0
	A-GWO	**0**	**0**	**0**	**0**	**0**
F22	GWO	2.00E-01	2.47E-01	2.00E-01	3.00E-01	5.07E-02
	A-GWO	**0**	**2.87E-61**	**5.00E-132**	**8.62E-60**	**1.57E-60**
F23	GWO	3.72E-02	5.63E-02	3.72E-02	7.82E-02	2.08E-02
	A-GWO	**0**	**5.69E-03**	**0**	**6.02E-02**	**1.30E-02**

Table 6 Best, average, median, worst and standard deviation of objective function values for unimodal test problems of 100 dimension

Function	*Algorithm*	*Best*	*Average*	*Median*	*Worst*	*STD*
F1	GWO	3.81E-13	1.73E-12	1.55E-12	4.10E-12	1.07E-12
	A-GWO	**0**	**0**	**0**	**0**	**0**
F2	GWO	9.94E-14	5.40E-13	3.94E-13	1.57E-12	4.08E-13
	A-GWO	**0**	**0**	**0**	**0**	**0**
F3	GWO	1.64E-08	4.26E-08	4.05E-08	7.8E-08	1.68E-08
	A-GWO	**2.44E-216**	**1.19E-200**	**4.05E-208**	**3.55E-199**	**0**
F4	GWO	2.70E+01	6.10E+02	4.54E+02	2.84E+03	6.26E+02
	A-GWO	**0**	**0**	**0**	**0**	**0**
F5	GWO	4.29E-02	8.45E-01	6.19E-01	3.25E+00	8.30E-01
	A-GWO	**4.91E-205**	**1.10E-197**	**2.40E-200**	**2.32E-196**	**0**

Table 6 Best, average, median, worst and standard deviation of objective function values for unimodal test problems of 100 dimension (continued)

Function	*Algorithm*	*Best*	*Average*	*Median*	*Worst*	*STD*
F6	GWO	**9.69E+01**	**9.79E+01**	**9.83E+01**	**9.86E+01**	6.01E-01
	A-GWO	9.83E+01	9.89E+01	9.90E+01	9.90E+01	**1.95E-01**
F7	GWO	7.88E+00	1.01E+01	1.02E+01	**1.20E+01**	**8.94E-01**
	A-GWO	**1.54E+00**	**6.09E+00**	**3.85E+00**	2.43E+01	5.49E+00
F8	GWO	1.51E-26	4.18E-25	1.38E-25	5.33E-24	1.01E-24
	A-GWO	**0**	**0**	**0**	**0**	**0**
F9	GWO	3.96E-03	6.82E-03	6.63E-03	1.04E-02	1.95E-03
	A-GWO	**1.13E-05**	**1.66E-04**	**1.44E-04**	**6.78E-04**	**1.53E-04**
F10	GWO	4.86E-84	7.77E-64	4.74E-74	1.64E-62	3.09E-63
	A-GWO	**0**	**0**	**0**	**0**	**0**

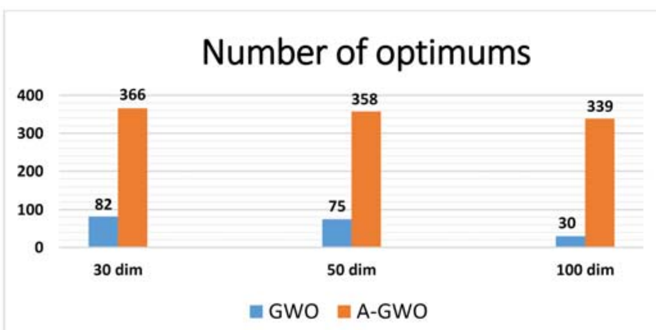
Table 7 Best, average, median, worst and standard deviation of objective function values for multimodal test problems of 100 dimension

Function	*Algorithm*	*Best*	*Average*	*Median*	*Worst*	*STD*
F11	GWO	**–1.85E+04**	**–1.44E+04**	**–1.60E+04**	**–5.49E+03**	4.05E+03
	A–GWO	–9.34E+03	–6.58E+03	–6.53E+03	–5.15E+03	**1.04E+03**
F12	GWO	8.65E-11	8.32E+00	5.57E+00	3.48E+01	8.26E+00
	A-GWO	**0**	**0**	**0**	**0**	**0**
F13	GWO	4.34E-08	1.42E-07	1.42E-07	2.53E-07	5.24E-08
	A-GWO	**8.88E-16**	**8.88E-16**	**8.88E-16**	**8.88E-16**	**4.01E-31**
F14	GWO	1.00E-13	9.36E-13	6.83E-13	3.14E-12	7.63E-13
	A-GWO	**0**	**0**	**0**	**0**	**0**
F15	GWO	**1.81E-01**	**2.95E-01**	**2.96E-01**	**3.98E-01**	**6.36E-02**
	A-GWO	3.01E-01	1.14E+00	1.20E+00	1.26E+00	2.21E-01
F16	GWO	5.96E+00	**6.84E+00**	**6.79E+00**	**7.86E+00**	**4.62E-01**
	A-GWO	**1.30E+00**	8.31E+00	9.99E+00	1.00E+01	2.51E+00
F17	GWO	9.48E-08	3.87E-03	3.53E-03	8.96E-03	2.42E-03
	A-GWO	**1.16E-212**	**1.04E-202**	**1.22E-206**	**2.89E-201**	**0.00E+00**
F18	GWO	1.6E-14	3.04E-14	3.02E-14	4.44E-14	6.72E-15
	A-GWO	**0**	**0**	**0**	**0**	**0**
F19	GWO	2.39E+01	1.03E+02	9.55E+01	2.57E+02	5.73E+01
	A-GWO	**0**	**0**	**0**	**0**	**0**
F20	GWO	4.66E-10	3.01E-09	1.74E-09	1.33E-08	3.24E-09
	A-GWO	**0**	**0**	**0**	**0**	**0**
F21	GWO	0	0	0	0	0
	A-GWO	**0**	**0**	**0**	**0**	**0**
F22	GWO	3.00E-01	3.60E-01	4.00E-01	5.00E-01	6.21E-02
	A-GWO	**0**	**7.62E-07**	**1.58E-101**	**2.29E-05**	**4.18E-06**
F23	GWO	7.82E-02	8.95E-02	7.82E-02	1.27E-01	2.08E-02
	A-GWO	**0**	**1.87E-02**	**9.98E-03**	**1.06E-01**	**2.25E-02**

4.2 Analysis of results

In this subsection, various experiments have been done to analyse the comparative performance between classical GWO and proposed A-GWO algorithms based on the results obtained as in Table 2 to 7. From these tables, it can be observed that for the dimension 30, 50 and 100, the proposed algorithm A-GWO is able to obtain the optima ($F_{min} = 0$) in problems F1, F2, F4, F8, F10, F12, F14, F18, F19, F20, F21, F22 and F23 and in problems F3, F5, F9, F13 and F17, the proposed algorithm outperform classical GWO in all the test statistics. While in the problems F6 and F11, classical GWO outperforms than proposed A-GWO algorithm. The number of optimums obtained by A-GWO and GWO is presented in Figure 4. It can be easily observed from Table 2 to 7 that the performance of A-GWO is not seriously deteriorated as the dimension increases. Therefore in terms of exploitation strength, the proposed algorithm AGWO outperform classical GWO, as unimodal test problems evaluate the exploitation ability of any stochastic search algorithm. As multimodal problems evaluate the local optima avoidance ability and exploration ability of any meta-heuristic algorithm, therefore on analysing the performance of the proposed algorithm on multimodal test problems the proposed algorithm A-GWO can be recommended over classical GWO to solve highly multimodal real-world optimisation problems. Therefore by analysing the results on all the test problems with dimensions 30, 50 and 100 on unimodal and multimodal test problems it can be concluded that for the problems with higher dimension also the proposed algorithm A-GWO can be used.

Figure 4 Number of global solutions obtained by A-GWO and classical GWO out of 690 runs* (see online version for colours)



Note: *Corresponding to 23 test problems and 30 runs for each problem.

4.3 Computational complexity of proposed algorithm

Meta-heuristic algorithms with less computational complexity are more suitable for real-world application problems with computationally expensive objective function. Therefore, the evaluation of any meta-heuristic algorithm is a very crucial task to observe the effectiveness of search algorithm. Both the algorithms classical GWO and proposed algorithm A-GWO have the same computation complexity in terms of worst time complexity as the structure of both the algorithms are same, only the acceleration factor is introduced in A-GWO that does not increase the computational complexity of the

GWO algorithm. Thus the complexity of both the algorithm is $O(T \times (N \times D \times C))$ where T represents the maximum number of iterations, N represents the size of the pack (number of wolves for the search process), D is the dimension of the problem and C represents the cost of objective function.

4.4 Diversity analysis in A-GWO using average distance between hunting wolves

Since the acceleration coefficient w has been integrated in A-GWO to enhance the diversity or exploration strength and to increase the exploitation in A-GWO the coefficient C is restricted in unit circle in whole search process of algorithm. To analyse these characteristics average distance between the hunting wolves has been calculated and presented in Figures 5 and 6 corresponding to unimodal problems F2, F4, F5 and F8 and multimodal problems F12, F17, F19, F20, F21 and F23. The Euclidean distance $\|\,.\,\|_2$ between any two vectors or search agents $x = (x_1, x_2, \ldots, x_d)$ and $y = (y_1, y_2, \ldots, y_d)$ can be calculated as:

$$\|x - y\|_2 = \sqrt{\sum_{j=1}^{d} (x_j - y_j)^2}$$

Figure 5 Average distance between hunting wolves to show the diversity enhancement and exploitation in A-GWO for functions F2, F4, F5, F8, F12 and F17 (see online version for colours)

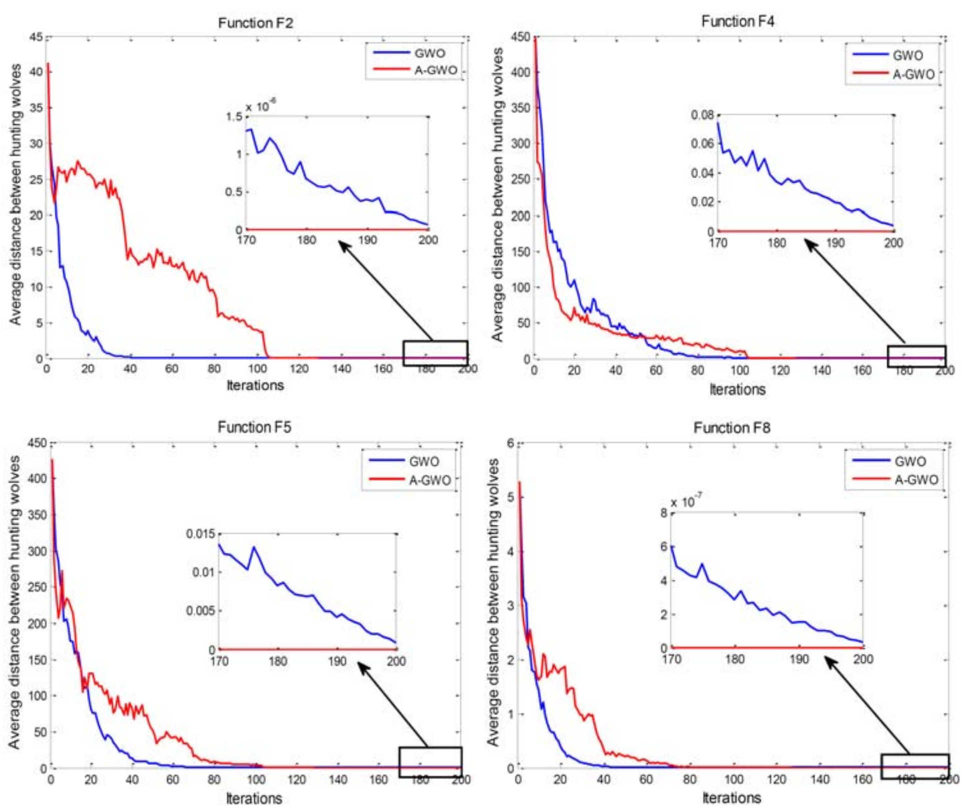


Figure 5 Average distance between hunting wolves to show the diversity enhancement and exploitation in A-GWO for functions F2, F4, F5, F8, F12 and F17 (continued) (see online version for colours)

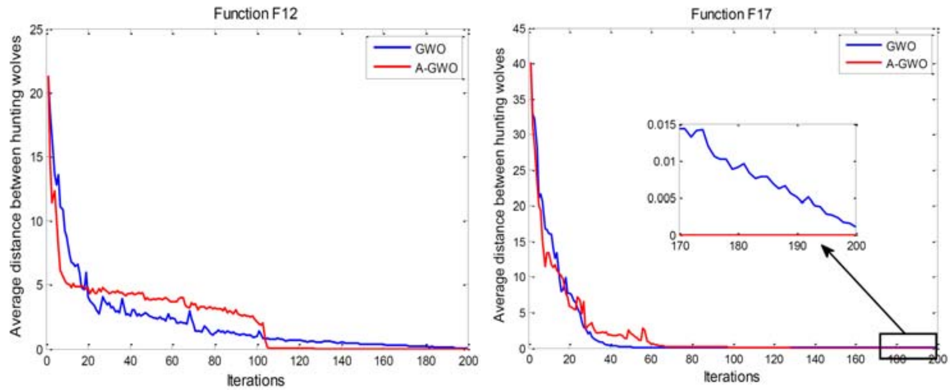
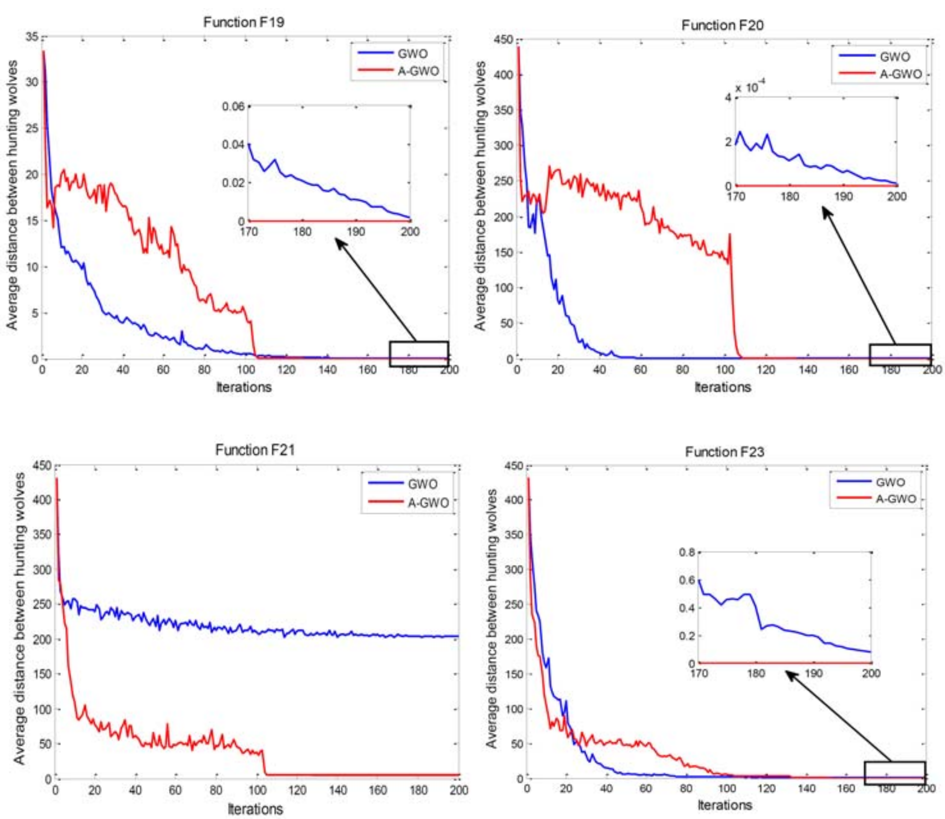


Figure 6 Average distance between hunting wolves to show the diversity enhancement and exploitation in A-GWO for functions F19, F20, F21 and F23 (see online version for colours)



From Figures 5 and 6, it can be analysed easily that as the distance between the hunting wolves is increased in A-GWO as compared to classical GWO until there is a possibility to discover the more promising regions of a search space and as the graph in A-GWO are more deviated which shows the better exploitation around visited search regions. When the wolves are able to locate the optima, i.e., when wolves have updated their states near the optima then the exploitation is occurred more quickly. In view of this, in most of the magnified figures, the distance graph in A-GWO is horizontal which shows that the wolf population has converged to optima because the obtained solution on these problems (Tables 2 to 7) are either optima or very near to optima. Thus the coefficient C is very meaningful as it is restricted for exploitation in whole algorithm.

4.5 Performance index analysis

In this section, the proposed A-GWO algorithm has been compared with classical GWO in terms of success rate, time complexity and error value by performance index (PI) analysis metric (Deep and Thakur, 2007). The relative performance of an algorithm using PI can be calculated as follows:

$$PI = \frac{1}{N_P}\sum_{i=1}^{N_P} \mu_1\alpha_1 + \mu_2\alpha_2 + \mu_3\alpha_3$$

where

$$\alpha_1 = \frac{SR^i}{TR^i}, \alpha_2 = \frac{MT^i}{AT^i}, \quad \text{and} \quad \alpha_3 = \frac{ME^i}{AE^i}$$

SR^i number of successful runs of the problem i

SR^i total number conducted for the problem i

MT^i minimum of average execution time taken by all the algorithms to solve the problem i

AT^i average execution time taken by an algorithm to solve the problem i

ME^i minimum of average error achieved by all the algorithms to solve the problem i

AE^P average error achieved by an algorithm to solve the problem i

N^P total number of problems.

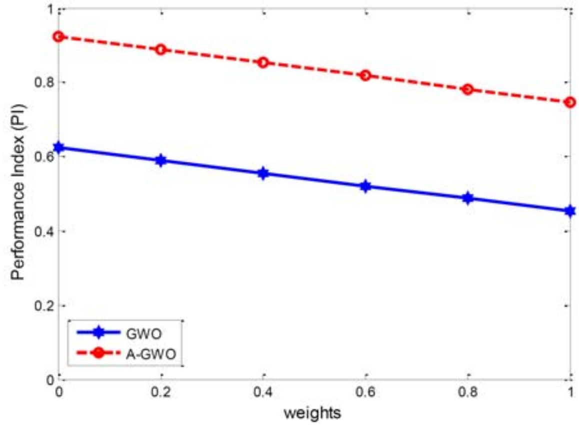
μ_1, μ_2 and μ_3 are non-negative weights ($\mu_1 + \mu_2 + \mu_3 = 1$) associated with success rate α_1, computational time term α_2, and error term α_3. The three different cases based on the weights assigned to the error, time and success terms are as follows:

Case 1 $\mu_1 = w, \mu_2 = \frac{(1-w)}{2}$ and $\mu_3 = \frac{(1-w)}{2}, 0 \le w \le 1.$

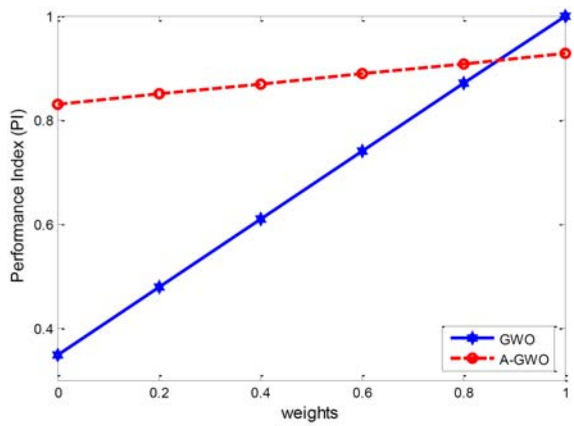
Case 2 $\mu_1 = \frac{(1-w)}{2}, \mu_2 = w$ and $\mu_3 = \frac{(1-w)}{2}, 0 \le w \le 1.$

Case 3 $\mu_1 = \frac{(1-w)}{2}, \mu_2 = \frac{(1-w)}{2}$ and $\mu_3 = w, 0 \le w \le 1.$

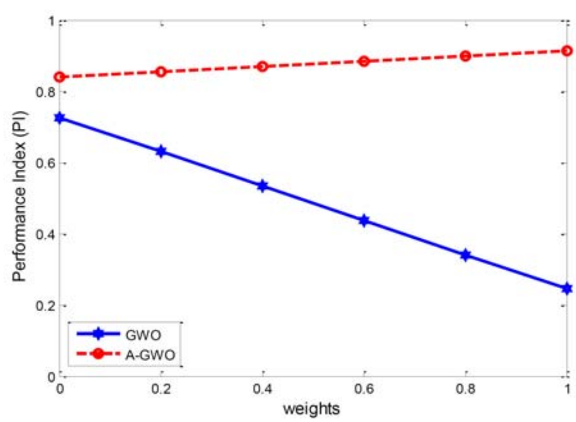
Figure 7 PI curves for the cases 1, 2 and 3 corresponding to 50 dimensional problem (see online version for colours)



(a)



(b)



(c)

The PI curves for all the above three cases are plotted in Figures 7(a), 7(b) and 7(c) respectively. In the PI graphs the horizontal axis represents the weights assigned to the success rate, time and error terms. The PI value is presented on vertical axes.

In the case 1, error and execution time are equally weighted. It can be analysed from the Figure 7(a) that PI of proposed A-GWO is much higher than classical GWO for all the weights. In case 2, an equal weight is assigned to the success and error term. From the Figure 7(b), it is clear that the PI curve is increases for A-GWO with the weight *w* and for all the weights except $w \geq 0.9$, the PI of proposed A-GWO is higher than classical GWO. In the case 3, the execution time and success rate are equally weighted. From 7(c) it can be analysed that PI of the proposed A-GWO algorithm is higher than classical GWO. Overall, from the PI curves, it can be concluded that when the user's requirement is less error and more success in a reasonable time of execution then the proposed algorithm A-GWO is preferred over classical GWO.

4.6 Effect of acceleration factor on the performance of A-GWO

In this section, 30, 50 and 100-dimensional test problems are used to investigate the impact of the acceleration factor on the performance of proposed algorithm A-GWO. In Figure 8, the graphs are plotted corresponding to the intervals.

1 [–0.2, 0.2]

2 [–0.4, 0.4]

3 [–0.6, 0.6]

4 [–1, 1].

In the figure, these intervals are numbered as 1, 2, 3 and 4 on X-axis. Vertical axis represents the mean objective function value of the test functions. From the figure it can be verified that the smaller value of *w* is very efficient to explore the search regions and enhances the performance of proposed AGWO algorithm. Therefore the optimal setting of the parameter is used in the paper as [–0.2, 0.2].

Figure 8 Distribution of objective function values with varying acceleration factor (*w*)

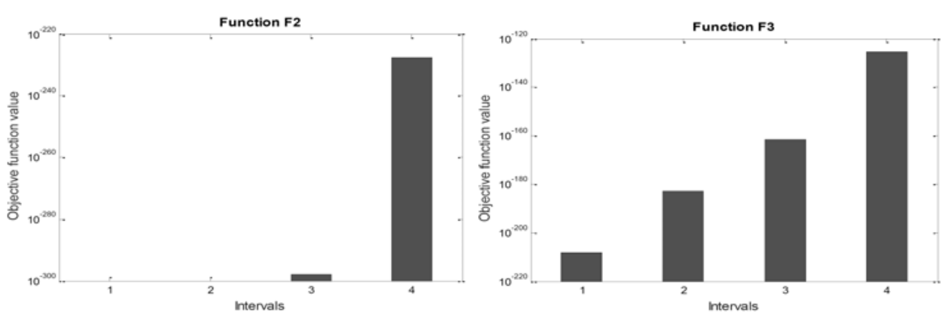
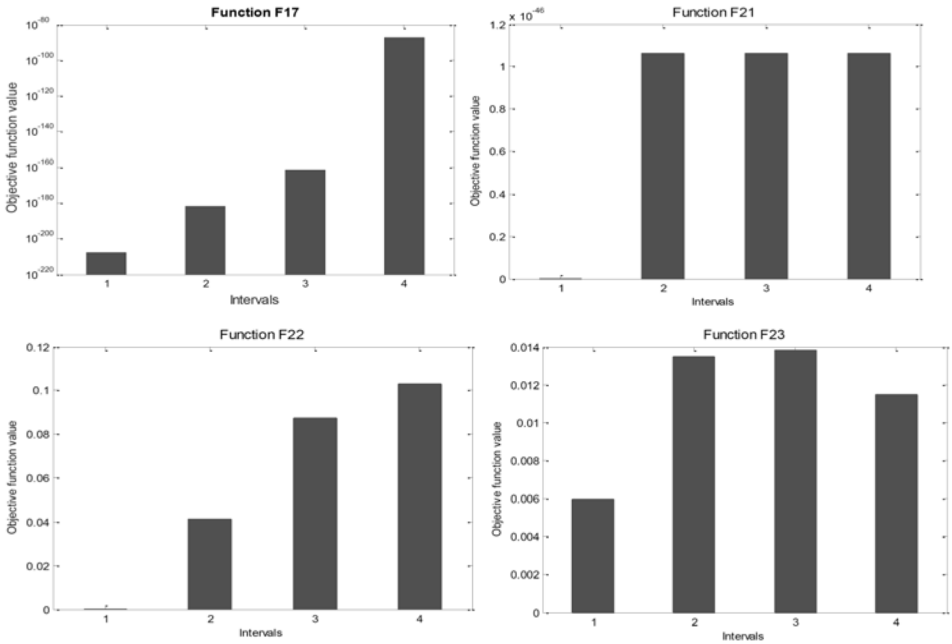


Figure 8 Distribution of objective function values with varying acceleration factor (w) (continued)



4.7 Comparison of the proposed algorithm A-GWO with other variants of GWO and recent algorithms

In this section, the proposed algorithm is compared with other variants of GWO that has been proposed to overcome the problem of stagnation in local optima and to maintain a suitable balance between exploration and exploitation. The proposed algorithm is compared in this section with classical GWO (Mirjalili et al., 2014), variants of GWO – modified GWO (mGWO) (Mittal et al., 2016), weighted GWO (represented as w1GWO) (Rodríguez et al., 2017), fitness weighted-based GWO (represented as w2GWO) (Rodríguez et al., 2017), astrophysics-inspired grey wolf optimiser (M-GWO) (Kumar and Kumar, 2018) and with recent optimisation meta-heuristic algorithms – Sine cosine algorithm (SCA) (Mirjalili, 2016), mothflame optimisation (MFO) algorithm (Mirjalili, 2015) and with salp swarm algorithm (SSA) (Mirjalili et al., 2017). For a fair comparison between these algorithms same population size 30 of wolves and 500 iterations are taken i.e., 15,000 function evaluations are used in all the algorithm. Tables 8 to 10 shows the average objective function values obtained over 30 independent runs corresponding to various algorithms. In the table the dimension of the problems are varied from 30 to 100 to observe the robustness of the proposed algorithm.

Table 8 Average objective function values over 30 runs corresponding to proposed algorithm A-GWO and other algorithms

Test problem	*Dim*	*GWO*	*mGWO*	*w1GWO*	*w2GWO*	*M-GWO*	*SSA*	*MFO*	*SCA*	*A-GWO*
F1	30	7.17E-28	4.05E-36	5.02E-31	2.56E-28	6.47E-05	2.24E-07	3.35E+03	1.07E+01	**0**
	50	5.11E-20	7.61E-26	2.96E-22	2.15E-20	1.09E-05	6.04E-01	1.03E+04	1.13E+03	**0**
	100	1.73E-12	2.68E-16	3.99E-14	8.16E-13	1.45E-02	1.37E+03	6.02E+04	1.13E+04	**0**
F2	30	9.73E-29	3.64E-37	5.85E-32	2.82E-29	3.34E+00	2.27E+00	7.34E+02	1.76E+00	**0**
	50	1.61E-20	2.27E-26	4.15E-23	4.16E-21	4.16E-03	3.47E+01	2.71E+03	1.55E+02	**0**
	100	5.40E-13	7.84E-17	1.82E-14	3.31E-13	1.47E+00	8.71E+02	2.59E+04	4.73E+03	**0**
F3	30	5.78E-17	1.18E-21	9.86E-19	7.19E-17	9.78E-04	2.13E+00	4.02E+01	1.59E-02	**7.42E-209**
	50	2.78E-12	5.35E-16	6.51E-14	2.17E-12	2.29E-02	8.50E+00	7.67E+01	5.39E-01	**2.38E-210**
	100	4.26E-08	1.47E-10	3.34E-09	2.95E-08	5.77E-01	4.74E+01	2.42E+02	9.84E+00	**1.19E-200**
F4	30	3.84E-06	2.10E-07	1.67E-06	2.51E-05	2.54E+02	1.58E+03	2.03E+04	9.01E+03	**0**
	50	1.39E-01	3.38E-02	2.30E-01	2.53E-01	4.21E+03	1.09E+04	6.59E+04	4.93E+04	**0**
	100	6.10E+02	6.39E+02	4.31E+02	5.15E+02	3.16E+04	4.47E+04	2.29E+05	2.38E+05	**0**
F5	30	6.19E-07	1.31E-09	1.48E-07	7.22E-07	1.99E-01	1.17E+01	7.07E+01	3.19E+01	**4.05E-205**
	50	5.08E-04	1.03E-05	3.20E-04	4.22E-04	5.12E+00	1.95E+01	8.43E+01	6.82E+01	**2.56E-200**
	100	8.45E-01	1.24E+00	1.45E+00	9.21E-01	3.59E+01	2.69E+01	9.29E+01	8.98E+01	**1.10E-197**
F6	30	2.71E+01	**2.69E+01**	2.70E+01	2.71E+01	2.83E+01	2.15E+02	1.07E+07	5.65E+04	2.89E+01
	50	4.75E+01	4.72E+01	**4.70E+01**	4.75E+01	5.08E+01	1.50E+03	8.54E+05	5.12E+06	4.88E+01
	100	**9.79E+01**	9.75E+01	**9.79E+01**	9.80E+01	4.84E+02	1.72E+05	1.64E+08	1.22E+08	9.89E+01
F7	30	7.57E-01	6.37E-01	7.11E-01	7.07E-01	1.67E+00	**2.05E-07**	1.69E+03	2.39E+01	6.74E-01
	50	2.78E+00	2.46E+00	2.64E+00	2.60E+00	5.42E+00	1.10E+00	9.53E+03	1.09E+03	**2.08E+00**
	100	1.01E+01	9.70E+00	9.60E+00	10.14E+00	2.01E+01	1.43E+03	6.61E+04	1.06E+04	**6.09E+00**

Table 9 Average objective function values over 30 runs corresponding to proposed algorithm A-GWO and other algorithms

Test problem	*Dim*	*GWO*	*mGWO*	*w1GWO*	*w2GWO*	*M-GWO*	*SSA*	*MFO*	*SCA*	*A-GWO*
F8	30	2.94E-51	2.09E-64	2.81E-55	5.80E-51	2.07E-13	3.14E-11	2.33E+00	1.57E-02	**0**
	50	6.55E-38	1.43E-46	3.83E-41	2.15E-38	3.14E-09	1.66E-04	2.38E+01	2.81E+00	**0**
	100	4.18E-25	1.30E-30	1.76E-27	4.56E-26	7.02E-05	2.41E-01	2.68E+02	1.32E+02	**0**
F9	30	1.70E-03	1.74E-03	1.89E-03	2.05E-03	7.90E-03	1.80E-01	5.16E+00	8.76E-02	**1.22E-04**
	50	3.01E-03	2.22E-03	2.72E-03	2.67E-03	1.56E-02	4.92E-01	2.54E+01	2.17E+00	**1.69E-04**
	100	6.82E-03	4.64E-03	5.85E-03	7.30E-03	5.05E-02	2.61E+00	2.72E+02	1.25E+02	**1.66E-04**
F10	30	7.77E-64	7.11E-122	6.04E-106	1.72E-100	2.02E-24	2.11E-06	1.37E-09	1.41E-05	**0**
	50	7.77E-64	1.69E-105	1.16E-90	4.20E-88	7.52E-20	2.17E-06	2.87E-05	1.21E-02	**0**
	100	7.77E-64	6.89E-45	1.17E-35	6.73E-28	3.94E-06	2.99E-06	7.25E-03	2.45E-01	**0**
F11	30	–6.10E+03	–5.67E+03	–6.15E+03	–5.97E+03	–4.21E+03	–7.47E+03	**–8.49E+03**	–3.78E+03	–3.51E+03
	50	–9.19E+03	–7.99E+03	–9.09E+03	–9.17E+03	–5.22E+03	–1.18E+04	**–1.33E+04**	–4.85E+03	–4.61E+03
	100	–1.44E+04	–1.36E+04	–1.62E+04	–1.56E+04	–7.45E+03	–2.11E+04	**–2.34E+04**	–6.79E+03	–6.58E+03
F12	30	1.01E+00	6.97E-02	2.18E+00	2.83E+00	8.16E+00	5.74E+01	1.68E+02	3.62E+01	**0**
	50	5.70E+00	4.75E-01	1.72E+00	3.84E+00	2.52E+01	9.07E+01	3.31E+02	1.11E+02	**0**
	100	8.32E+00	6.09E-01	6.37E+00	8.11E+00	7.53E+01	2.31E+02	8.74E+02	2.50E+02	**0**
F13	30	1.00E-13	2.24E-14	4.99E-14	9.31E-14	1.39E+01	2.76E+00	1.68E+01	1.49E+01	**8.88E-16**
	50	3.71E-11	1.41E-13	3.15E-12	2.62E-11	1.99E+01	4.69E+00	1.93E+01	1.87E+01	**8.88E-16**
	100	1.42E-07	1.76E-09	2.06E-08	1.04E-07	2.00E+01	1.04E+01	1.99E+01	1.79E+01	**8.88E-16**
F14	30	8.88E-16	3.70E-03	1.30E-03	4.73E-03	1.95E-02	2.15E-02	3.23E+01	9.15E-01	**0**
	50	2.39E-03	5.54E-04	2.43E-03	2.51E-03	7.93E-02	4.50E-01	9.33E+01	1.12E+01	**0**
	100	9.36E-13	1.19E-03	1.25E-03	3.27E-03	8.48E-01	1.33E+01	5.43E+02	1.03E+02	**0**
F15	30	4.49E-02	**4.26E-02**	4.45E-02	5.70E-02	1.21E-01	7.77E+00	1.01E+01	4.62E+04	1.39E-01
	50	1.23E-01	**1.01E-01**	1.09E-01	1.65E-01	3.06E-01	1.16E+01	3.50E+07	1.62E+07	3.66E-01
	100	2.95E-01	**2.46E-01**	2.77E-01	3.92E-01	1.35E+00	3.45E+01	2.78E+08	3.57E+08	1.14E+00

Table 10 Comparison of average objective function values over 30 runs corresponding to proposed algorithm A-GWO and other algorithms

Test problem	*Dim*	*GWO*	*mGWO*	*w1GWO*	*w2GWO*	*M-GWO*	*SSA*	*MFO*	*SCA*	*A-GWO*
F16	30	7.24E-01	5.30E-01	**5.10E-01**	8.26E-01	1.28E+00	1.66E+01	4.19E+01	2.43E+05	8.13E-01
	50	2.13E+00	1.79E+00	2.04E+00	2.29E+00	3.52E+00	7.84E+01	8.70E+07	3.67E+07	**1.51E+00**
	100	6.84E+00	**6.28E+00**	6.80E+00	6.98E+00	1.56E+01	4.02E+03	6.73E+08	5.66E+08	8.31E+00
F17	30	7.13E-04	9.04E-05	3.75E-04	5.03E-04	4.78E-03	4.25E+00	4.28E+00	1.09E+00	**1.70E-208**
	50	8.19E-04	1.58E-04	8.59E-04	1.05E-03	3.39E-02	9.63E+00	1.12E+01	6.86E+00	**1.29E-205**
	100	3.87E-03	1.95E-04	1.84E-03	4.29E-03	2.00E-01	2.83E+01	6.94E+01	2.75E+01	**1.04E-202**
F18	30	0	0	0	0	8.53E-08	1.16E+00	6.67E-01	1.34E-02	**0**
	50	3.26E-16	0	1.48E-16	2.96E-17	1.80E-05	1.94E+00	2.75E+00	2.32E-01	**0**
	100	3.04E-14	2.19E-15	1.06E-14	2.46E-14	3.54E-03	4.59E+00	1.31E+01	2.23E+00	**0**
F19	30	1.06E-07	2.89E-10	1.23E-09	5.62E-08	1.30E+01	5.04E+01	3.51E+02	2.64E+01	**0**
	50	4.16E-02	5.29E-03	1.08E-02	6.90E-02	1.43E+02	4.01E+02	9.27E+02	1.53E+02	**0**
	100	1.03E+02	7.28E+01	6.02E+01	1.34E+02	6.37E+02	1.83E+03	2.76E+03	7.04E+02	**0**
F20	30	3.07E-24	1.29E-32	1.28E-27	1.05E-24	2.29E-01	1.36E+07	7.84E+07	1.50E+03	**0**
	50	1.48E-16	1.37E-22	3.77E-19	4.36E-17	3.94E+01	2.25E+07	3.23E+08	1.54E+05	**0**
	100	3.01E-09	4.42E-13	5.70E-11	1.54E-09	7.43E+03	8.55E+07	1.70E+09	1.65E+07	**0**
F21	30	0	0	0	0	0	0.00E+00	0.00E+00	0.00E+00	**0**
	50	0	0	0	0	0	0.00E+00	0.00E+00	0.00E+00	**0**
	100	0	0	0	0	0	0.00E+00	0.00E+00	0.00E+00	**0**
F22	30	1.83E-01	1.50E-01	1.73E-01	1.86E-01	2.57E-01	2.03E+00	5.16E+00	9.95E-01	**1.02E-81**
	50	2.47E-01	1.91E-01	2.33E-01	2.56E-01	4.64E-01	5.04E+00	1.50E+01	4.19E+00	**2.87E-61**
	100	3.60E-01	2.60E-01	3.30E-01	3.53E-01	1.26E+00	1.16E+01	3.47E+01	1.15E+01	**7.62E-07**
F23	30	3.86E-02	2.25E-02	3.49E-02	3.63E-02	6.84E-02	4.18E-01	4.95E-01	2.13E-01	**5.95E-03**
	50	5.63E-02	2.99E-02	5.36E-02	4.70E-02	1.71E-01	4.89E-01	5.00E-01	4.76E-01	**5.69E-03**
	100	8.95E-02	6.32E-02	9.49E-02	1.01E-01	4.31E-01	4.99E-01	5.00E-01	5.00E-01	**1.87E-02**

Inspecting the results, it is evident that A-GWO is more efficient than other variants of GWO in terms of determining the solution of the problem. In all the unimodal test problems (F1–F10) except F6 proposed algorithm outperforms other variants in all the dimension while in multimodal test problems (F11–F23) proposed algorithm outperforms than other variants of GWO in all the test functions except F11, F15 for all the dimension 30, 50 and 100. Thus on analysing the comparative performance of the proposed algorithm A-GWO, it can be concluded that the proposed algorithm is capable of exploring the search space more efficiently due to the acceleration coefficient and also capable for the local exploitation of the search space by restricting the coefficient C in a unit circle. Moreover, the performance of the proposed algorithm A-GWO on multi-modal benchmark test problems ensures that the algorithm is more reliable to overcome the problem of stagnation in local optima.

4.8 Statistical analysis

The statistical metrics used in Tables 8–10 were mean and standard deviation. The mean values show the average performance and the standard deviation shows the stability of the algorithm. Such metrics compare algorithms on average of 30 runs, but do not include each run. They do not show the significance of the results either. The statistical analysis (Derrac et al., 2011) of the results obtained from classical GWO and proposed A-GWO is necessary to evaluate the performance of algorithms to observe the significant improvement. In this subsection, to investigate the statistical validity of the results, a non-parametric pairwise Wilcoxon signed rank test is employed between GWO and proposed Accelerated GWO algorithms, and the obtained results are reported in Table 11. The test is conducted with 5% level of significance. In Table 11 the statistical results are reported for dimensions 30, 50 and 100 of unimodal and multimodal test problems. In the table sign '+' indicates that the proposed algorithm A-GWO performs significantly better than classical GWO, '–' indicates that the classical GWO performs significantly better than proposed A-GWO algorithm and '=' indicates that both the algorithms perform statistically same. Also, the *p*-values obtained by applying Wilcoxon test are also reported in the same table. Thus the statistical analysis ensures the better performance of proposed algorithm A-GWO compared to classical GWO in most of the test problems.

Table 11 Statistical results on unimodal and multimodal benchmark test problems using Wilcoxon signed rank test

Test function	*p-value*	*Conclusion*	*Test function*	*p-value*	*Conclusion*
		Dimension = 30			
F1	1.73E-06	+	F12	1.71E-06	+
F2	1.73E-06	+	F13	1.62E-06	+
F3	1.73E-06	+	F14	7.80E-03	+
F4	1.73E-06	+	F15	1.57E-02	+
F5	1.73E-06	+	F16	5.86E-01	+
F6	1.73E-06	–	F17	1.73E-06	+
F7	1.53E-01	=	F18	1.00E+00	=
F8	1.73E-06	+	F19	1.73E-06	+
F9	1.73E-06	+	F20	1.73E-06	+
F10	1.73E-06	+	F21	1.00E+00	=
F11	1.73E-06	–	F22	2.57E-07	+
			F23	3.93E-06	+

Table 11 Statistical results on unimodal and multimodal benchmark test problems using Wilcoxon signed rank test (continued)

Test function	*p-value*	*Conclusion*	*Test function*	*p-value*	*Conclusion*
		Dimension = 50			
F1	1.73E-06	+	F12	1.73E-06	+
F2	1.73E-06	+	F13	1.73E-06	+
F3	1.73E-06	+	F14	1.56E-02	+
F4	1.73E-06	+	F15	4.05E-01	=
F5	1.73E-06	+	F16	8.59E-02	+
F6	3.88E-06	–	F17	1.73E-06	+
F7	4.28E-02	+	F18	7.80E-03	+
F8	1.73E-06	+	F19	1.73E-06	+
F9	1.73E-06	+	F20	1.73E-06	+
F10	1.73E-06	+	F21	1.00E+00	=
F11	1.73E-06	–	F22	8.12E-07	+
			F23	1.48E-06	+
		Dimension = 100			
F1	1.73E-06	+	F12	1.73E-06	+
F2	1.73E-06	+	F13	1.73E-06	+
F3	1.73E-06	+	F14	1.73E-06	+
F4	1.73E-06	+	F15	1.92E-06	–
F5	1.73E-06	+	F16	1.70E-03	–
F6	1.73E-06	–	F17	1.73E-06	+
F7	1.10E-03	+	F18	1.66E-06	+
F8	1.73E-06	+	F19	1.73E-06	+
F9	1.73E-06	+	F20	1.73E-06	+
F10	1.73E-06	+	F21	1.00E+00	=
F11	5.75E-06	–	F22	1.01E-06	+
			F23	1.70E-06	+

4.9 Convergence analysis

To analyse the convergence speed and accuracy of obtaining optimal solution, the convergence graphs corresponding to some typical problems of classical benchmark set with dimension 100 are illustrated in Figures 9 and 10. In the figures, convergence curves are plotted corresponding to the median of best objective function values obtained in 30 runs. The horizontal axis represents the iterations and the vertical axis represents the fitness of objective function. Inspecting the results, it can be observed that the proposed algorithm converges towards the global optima faster (in terms of function evaluation cost) than classical GWO algorithm in most of the test problems. In the convergence curves, it can also be analysed that after half of the maximum number of iterations, the proposed algorithm A-GWO shows their impact in terms of convergence rate. Thus the convergence graphs verified the advantages of introducing acceleration factor in the proposed algorithm. In the convergences curves the break of graph represents the achievement of optima for the problem.

In the convergence curve, various curves are plotted corresponding to various acceleration factors. The curves corresponding to accelerated factors –0.1, 0.1, –0.05, 0.05 are denoted in the figures by ac1-GWO, ac2-GWO, ac3-GWO and ac4-GWO respectively. These curves corresponding to various accelerated factors are plotted to demonstrate that any random values with the interval [0.2, 0.2] shows the similar performance.

Figure 9 Convergence curves for selected unimodal test problems in classical set of benchmark problems (see online version for colours)

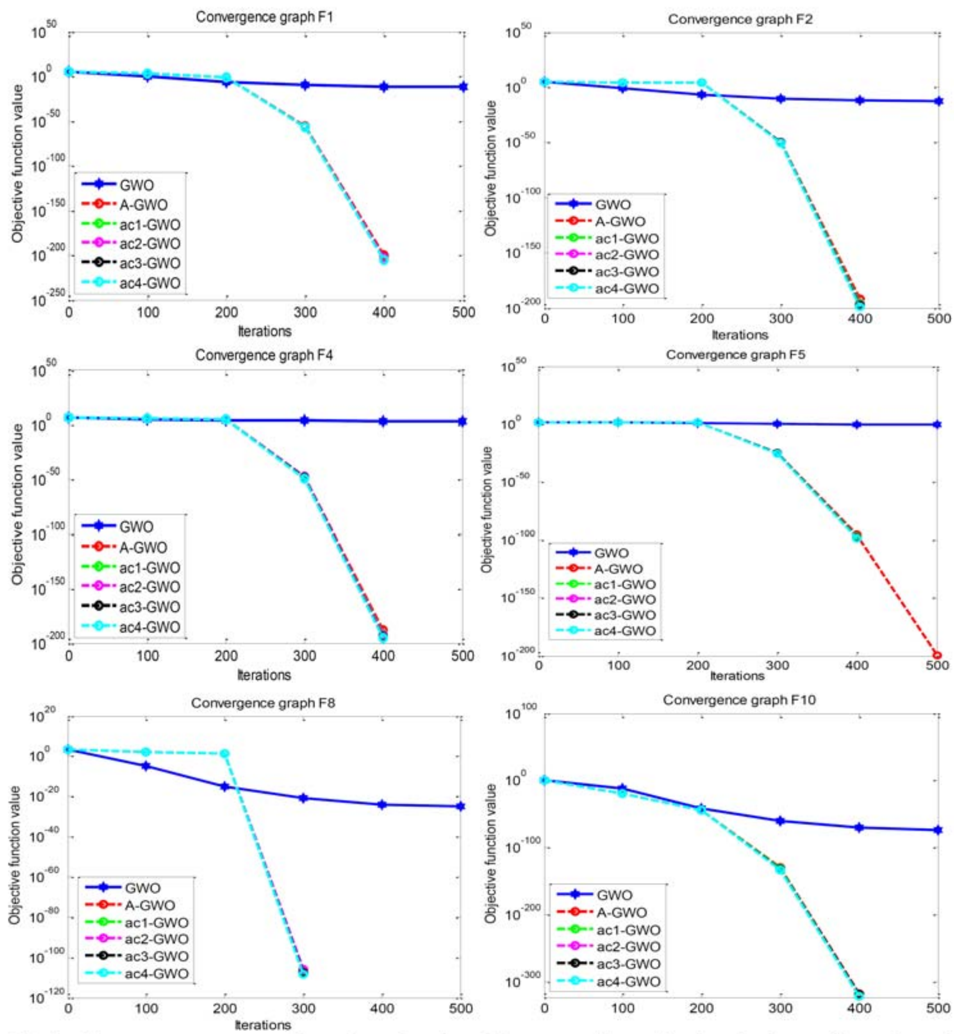
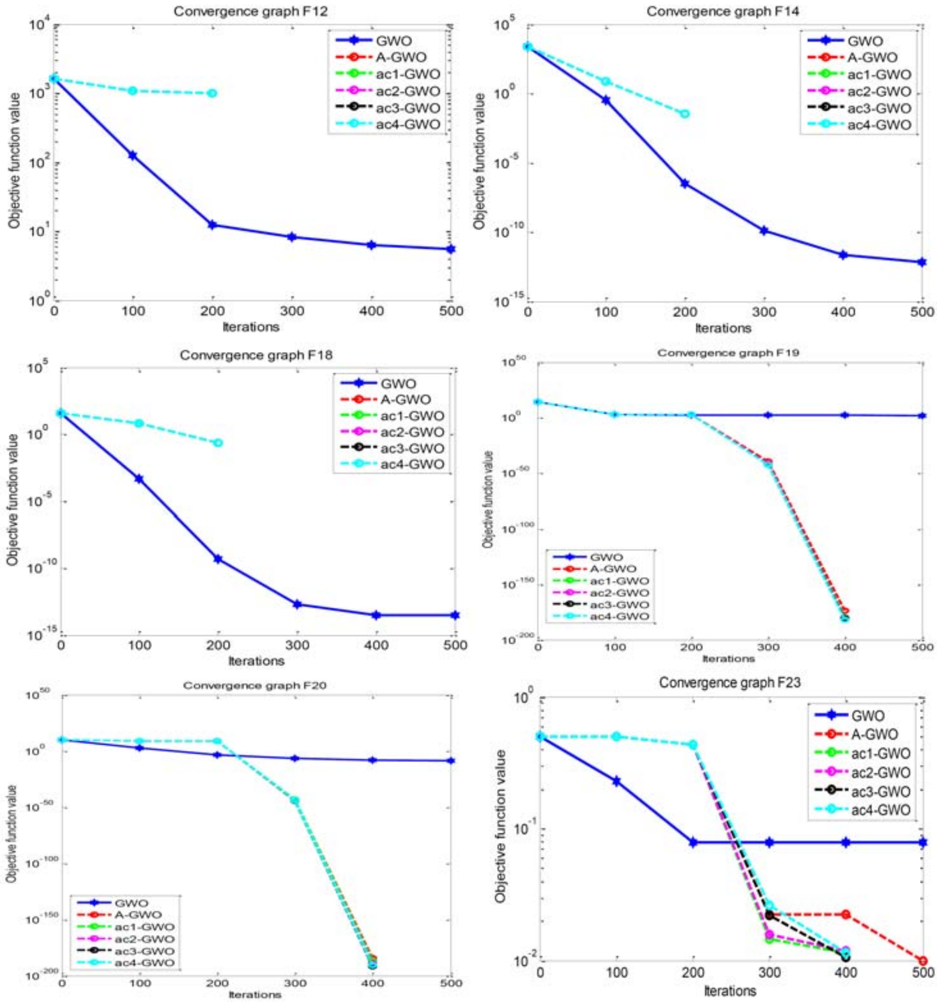


Figure 10 Convergence curves for selected multimodal test problems in classical set of benchmark problems (see online version for colours)



5 Applications of A-GWO

In this Section, the proposed algorithm A-GWO is employed to solve five engineering optimisation test problems – Gear train design, Estimation of parameter for frequency modulated, beam design problem, Himmelblau's optimisation problem and minimising the cost of life support system in a space capsule. The Himmelblau's optimisation problem is used as a challenging problem by a large number of works in the literature. As these problems are with some constraints, therefore to deal with the constraints a simple constraint handling is used in which the wolves are sorted in increasing order of constraint violation (Deb, 2000) and in this way top three wolves (less violated solutions) are selected as alpha, beta and delta of the wolf pack for the search process of prey in each generation of algorithm. The following subsection present and analyse the results obtained.

5.1 Gear train design problem

Gear train design problem can be considered as discrete optimisation problems as in this problem the decision variables x_1, x_2, x_3 and x_4 represent the numbers of tooths for four gears of a train. The objective of gear train design problem is to find the optimal number of a tooth to minimise the gear ratio (Gandomi, 2014; Sandgren, 1990). The discrete parameters in the algorithm are handled by rounding them to the nearest integral value after updating the solution using a considered algorithm. Mathematically, the problem can be formulated as follows:

$$\text{Min} \quad f_1(x) = \left(\frac{1}{6.931} - \frac{x_2 x_3}{x_1 x_4} \right)^2, \quad \text{where } x = (x_1, x_2, x_3, x_4)$$

$$\text{s.t.} \quad 12 \le x_i \le 60, \text{ and } x_i \in Z^+ \quad \forall i = 1, 2, 3, 4.$$

The obtained best solutions by proposed algorithm A-GWO and classical GWO are presented in Table 12. This problem is solved in the literature by augmented lagrange multiplier (ALM) (Kannan and Kramer, 1994), GA (Deb and Goyal, 1996), ABC algorithm (Karaboga and Basturk, 2007), CS algorithm (Gandomi et al., 2013) and mine blast algorithm (MBA) (Sadollah et al., 2013). From the comparison of results, it can be observed that the proposed algorithm A-GWO, is better in terms of finding the solution of problems as well as in used function evaluation cost. As CS and MBA provide the same objective function value as m-SCA but with more computational cost.

Table 12 Comparison of results on gear train design problem

Algorithm	*Optimal decision variables*				*Objective function value ($f_{1,min}$)*	*Number of function evaluations*
	x_1	x_2	x_3	x_4		
A-GWO	43	16	19	49	2.7009E-12	2,965
GWO	51	13	30	53	1.3616E-09	2,965
GA	33	14	17	50	1.3616E-09	NA
ALM	33	15	13	41	2.4073E-08	NA
ABC	43	16	19	49	2.7800E-11	40,000
MBA	43	16	19	49	2.7009E-12	10,000
CS	43	16	19	49	2.7009E-12	5,000

5.2 Estimation of parameter for frequency-modulated (FM) (Das and Suganthan, 2012)

The objective of this problem is to determine the decision parameters of frequency-modulated synthesiser. This problem is highly complex and multimodal with strong epistasis. Moreover, this problem consists of six decision parameters. Mathematically, the FM problem can be stated as follows:

$$\text{Min:}\quad f_2(X) = \sum_{t=1}^{100} \left(Y(t) - Y_0(t)\right)^2, \quad X = (a_1, \omega_1, a_2, \omega_2, a_3, \omega_3)$$

$$\text{s.t.}\quad -6.4 \le a_1, \omega_1, a_2, \omega_2, a_3, \omega_3 \le 6.35$$

where

$$Y(t) = a_1 \sin\left(\omega_1 t\theta + a_2 \sin\left(\omega_2 t\theta + a_3 \sin\left(\omega_3 t\theta\right)\right)\right)$$

The expression for target sound waves is given by

$$Y_0(t) - a_1 \sin\left(5t \times \frac{2\pi}{100} + 1.5 \sin\left(4.8t \times \frac{2\pi}{100} + 2 \sin\left(4.9t \times \frac{2\pi}{100}\right)\right)\right)$$

The obtained best solutions by A-GWO and classical GWO are presented in Table 13. In the literature, this problem has been solved by various optimisation search algorithms. In Table 13, the results obtained from PSO (Eberhart and Kennedy, 1995), G-CMA-ES (Liang et al., 2006; Van den Bergh and Engelbrecht, 2004) and CPSOH (Van den Bergh and Engelbrecht, 2004; Auger and Hansen, 2005) are also reported. From the results, it can be analysed that the proposed algorithm A-GWO is better in terms of best, mean and worst objective function value. Therefore, from the results, obtained from A-GWO it can be concluded that the proposed algorithm A-GWO is more efficient as compared to other algorithms to estimate the parameters for frequency-modulated.

Table 13 Comparison of results on parameter estimation for frequency-modulated

Algorithm	*Min*	*Mean*	*Max*	*STD*
A-GWO	2.91	15.87	24.20	6.12
GWO	10.66	18.16	25.10	4.74
G-CMA-ES	3.33	38.75	55.09	16.77
CPSOH	3.45	27.08	42.52	60.61
PSO	25.24	27.631	29.65	1.17

5.3 I-beam design problem

This case study is the modified form of the original problem (Gold and Krishnamurty, 1997). This is one of the important structural optimisation application problems. The objective of this problem is to minimise the vertical deflection of a beam with satisfying the cross-sectional area and stress constraints simultaneously. Mathematically the problem can be stated as:

$$\text{Min}\quad f_3(x) = \frac{5{,}000}{\frac{x_3(x_1 - x_4)}{12} + \frac{x_2 x_4^3}{+} + 2x_2x_4\left(\frac{x_1 - x_4}{2}\right)^2},$$

$$x = (x_1, x_2, x_3, x_4) = (b, h, t_w, t_f)$$

$$\text{s.t.}\quad g_1(x) = 2x_2x_4 + x_3(x_1 - 2x_4) \le 300$$

$$g_2(x) = \frac{18x_1 \times 10^4}{x_3(x_1 - 2x_4)^2 + 2x_2x_4\left(4x_4^2 + 3x_1(x_1 - 2x_4)\right)} + \frac{15x_2 \times 10^3}{(x_1 - 2x_4)_3^3 + 2x_4x_2^3}$$

$$10 \le x_1 \le 80, \quad 10 \le xl2 \le 50, \quad 0.9 \le x_3, x_4 \le 5.$$

In this paper, this problem is solved using classical GWO and A-GWO with same function evaluations used in Gandomi et al. (2013) by CS. This problem is also solved by ARSM (Wang, 2003) and improved ARSM (Wang, 2003) and the obtained results are reported in Table 14. From the tables, it can be easily observed that A-GWO solve this problem with better objective function value as compared to other algorithms.

Table 14 Comparison of results on I-beam design problem

Algorithm	*Optimal solution*				$f_{3,\min}$
	x_1	x_2	x_3	X_4	
A-GWO (present study)	80	50	0.9	2.32172	0.013074
ARSM	80	37.05	1.71	2.31	0.01570
Improved ARSM	79.99	48.42	0.9	2.40	0.1310
CS	80	50	0.9	2.3216715	0.0130747
GWO	80	50	0.9	2.32177	0.013075

5.4 Himmelblau's optimization problem

This nonlinear problem was originally proposed by Himmelblau (1972). This problem has five decision variables x_1, x_2, x_3, x_4 and x_5 and six nonlinear inequality constraints and 10 bound constraints. Mathematical the problem can be stated as follows:

$$\text{Min}\quad f_4(y) = 5.3578547y_3^2 + 0.8356891y_1y_5 + 37.293239y_1 - 40792.141$$

$$y = (y_1, y_2, y_3, y_4, y_5)$$

$$\text{s.t.}\quad g_1(y) = 85.334407\ 0.0056858y_2y_5 + 0.0026y_1y_4 - 0.0022053y_3y_5 \le 0$$

$$g_2(y) = 80.51249 + 0.0071317y_2y_5 + 0.0029955y_1y_2 + 0.0021813y_3^4 \le 0$$

$$g_3(y) = 9.300961 + 0.0047026y_3y_5 + 0.0012547y_1y_3 + 0.0019085y_3y_4 \le 0$$

$$0 \le g_1(y) \le 92$$

$$90 \le g_2(y) \le 110$$

$$20 \le g_3(y) \le 25$$

$$78 \le y_1 \le 102$$

$$33 \le y_2 \le 45$$

$$27 \le y_3 \le 45$$

$$27 \le y_4 \le 45$$

$$27 \le y_5 \le 45$$

The constrained classical GWO and A-GWO are run 30 times, and the results are reported in Table 15. This problem is solved in Gandomi et al. (2013) by the CS algorithm. Same function evaluations are used to solve this problem by A-GWO. In is evident in Table 15 that the proposed algorithm is very competitive as compared to other state-of-the art algorithms and provides a feasible solution.

Table 15 Comparative results on Himmelblau problem

Algorithm	*Optimum value* $f_{4,\min}$	*Constraint value*		
		$0 \le g_1(y) \le 92$	$90 \le g_2(y) \le 110$	$20 \le g_3(y) \le 25$
Himmelblau (1972)	–30,373.949	NA	NA	NA
Homaifar et al. (1994)	–30,005.700	91.6562	99.5369	20.0255
Gen and Cheng (1997)	–30,183.576	NA	NA	NA
GWO	–31,025.314	91.9989	100.4035	20.0001
Shi and Eberhart (1998)	–31,025.561	93.2853 (infeasible)	100.4047	20.000
CS (Gandomi et al., 2013)	–30,665.233	91.9999	98.8407	20.0004
Lee and Geem (2005)	–30,665.500	92.0000	98.8405	19.9999
A-GWO (present study)	–31,015.740	91.9755	100.3794	20.0079

5.5 *Space capsule problem of life support system (Tillman et al., 1980)*

This is a nonlinear reliability problem of a life-support system having a complex system. This system (Tillman et al., 1980) has four components with reliability $x_i(I = 1, 2, 3, 4)$. The goal of this problem is to minimise the system cost with constraint on reliability. The final expression for the overall system reliability can be defined as:

$$R_s = 1 - x_3 \times \left[(1-x_1)(1-x_4)\right]^2 - (1-x_3)\left[1 - x_2\left\{1-(1-x_1)(1-x_4)\right\}\right]^2$$

and the system cost is given by

$$C_s = 2k_1 x_1^{p_1} + 2k_2 x_2^{p_2} + k_3 x_3^{p_3} + 2k_4 x_4^{p_4}$$

where $k_1 = k_2 = 100$, $k_3 = 200$, $k_4 = 300$, $p_1 = p_2 = p_3 = p_4 = 0.6$.

Mathematically the problem can be stated as:

$$\text{Min} \quad C_s = 2k_1 x_1^{p_1} + 2k_2 x_2^{p_2} + k_3 x_3^{p_3} + 2k_4 x_4^{p_4}$$

$$\text{s.t.} \quad 0.5 \le x_i \le 1 \quad i = 1, 2, 3, 4.$$

$$0.9 \le R_s \le 1$$

Here x_i is the reliability of the component i.

Table 16 Optimisation results of A-GWO on life support system problem

Algorithm	*Optimal solution*				C_s	R_s
	x_1	x_2	x_3	x_4		
A-GWO	0.50000	0.83892	0.50000	0.50000	*641.823562*	0.90000
I-NESA	0.50000	0.83892	0.50000	0.50000	641.823700	0.90000
SA	0.50006	0.83887	0.50001	0.50002	641.833200	0.90001
ACO	0.50000	0.83890	0.50000	0.50000	641.823562	0.90000
GWO	0.50095	0.83775	0.50025	0.50015	641.903000	0.90001

The classical GWO and A-GWO are run 30 times with the wolf pack of size 100 and 500 maximum number of iterations. Thus the total number of function evaluation is 50,000. The numerical results acquired from the A-GWO and classical GWO are listed in Table 16. In the same table, the results of Simulated Annealing (SA) (Ravi et al., 1997), ACO (Shelokar et al., 2002) and Non-equilibrium simulated-annealing algorithm (I-NESA) (Ravi et al., 1997) are also shown. The comparison of results shows the better accuracy of the proposed A-GWO than other optimisation methods.

Some concluding comments that can be pointed out based on the present work are as follows:

1 promising search regions in A-GWO are saved since the states of wolves are improved with the help of leading hunters

2 integration of acceleration coefficient in A-GWO guarantees the enhanced exploration of a search space during the search process

3 the acceleration coefficient provides a higher chance for the wolves to prevent from the local optima

4 the coefficient C emphasises on the exploitation of a search space around the pre-discovered regions

5 the diversity plot of search agents demonstrates the diversity within the wolf population and exploitation ability of grey wolves near the obtained solutions.

6 Conclusions and future scope

The paper presents an improved GWO based on acceleration factor that is introduced in GWO to enhance the exploration by the grey wolves especially at the later generations as an insufficient exploration of a search space has been observed after the half of the maximum number of iterations when the parameter $\vec{A}$ lies between (–1, 1). Therefore, in that situation, to explore the more promising regions of a search space the ability of GWO in terms of exploration is improved by proposing A-GWO. Exploitation strength of classical GWO is also increased by restricting the vector $\vec{C}$ in a unit circle to exploit the regions around the visited search regions. A set of 23 well-known benchmark problems of different categories unimodal and multimodal with dimension 30, 50 and 100 has been taken to evaluate the performance of proposed algorithms. The statistical, convergence and PI analysis verified the strength of the proposed algorithm. The convergence analysis also ensures that the proposed algorithm A-GWO accelerate the convergence rate of the proposed algorithm. Various acceleration factor parameters are presented to verify the parameter selection in the paper. The proposed algorithm is also employed on five engineering application problems which conclude that for solving real-world optimisation problems the proposed algorithm can be preferred over classical GWO.

Towards the other research directions, in future, the constrained version of proposed AGWO algorithm using various constrained handling techniques can be developed to solve real-world application problems. Also, the discrete version of A-GWO can be designed to solve the optimisation problems with discrete variables.

Acknowledgements

The first author gratefully acknowledges the Ministry of Human Resource and Development (MHRD), Government of India for the financial support. Grant No. MHR-02-41-113-429.

References

Auger, A. and Hansen, N. (2005) 'A restart CMA evolution strategy with increasing population size', *The IEEE Congress on Evolutionary Computation 2005*, IEEE, Vol. 2, pp.1769–1776.

Bansal, J.C., Sharma, H., Jadon, S.S. and Clerc, M. (2014) 'Spider monkey optimization algorithm for numerical optimization', *Memetic Computing*, Vol. 6, No. 1, pp.31–47.

Das, S. and Suganthan, P.N. (2012) *Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems*, Technical Report, pp.1–42.

Deb, K. (2000) 'An efficient constraint handling method for genetic algorithms', *Computer Methods in Applied Mechanics and Engineering*, Vol. 186, Nos. 2–4, pp.311–338.

Deb, K. and Goyal, M. (1996) 'A combined genetic adaptive search (GeneAS) for engineering design', *Computer Science and Informatics*, Vol. 26, pp.30–45.

Deep, K. and Thakur, M. (2007) 'A new mutation operator for real coded genetic algorithms', *Applied Mathematics and Computation*, Vol. 193, No. 1, pp.211–230.

Derrac, J., García, S., Molina, D. and Herrera, F. (2011) 'A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms', *Swarm and Evolutionary Computation*, Vol. 1, No. 1, pp.3–18.

Diouane, Y., Gratton, S. and Vicente, L.N. (2015) 'Globally convergent evolution strategies for constrained optimization', *Computational Optimization and Applications*, Vol. 62, No. 2, pp.323-346.

Dorigo, M. and Birattari, M. (2011) 'Ant colony optimization', in *Encyclopedia of Machine Learning*, pp.36–39, Springer, Boston, MA.

Eberhart, R. and Kennedy, J. (1995) 'A new optimizer using particle swarm theory', *Proceedings of the Sixth International Symposium on Micro Machine and Human Science, MHS' 95*, IEEE, pp.39–43.

Gandomi, A.H. (2014) 'Interior search algorithm (ISA): a novel approach for global optimization', *ISA Transactions*, Vol. 53, No. 4, pp.1168–1183.

Gandomi, A.H., Yang, X.S. and Alavi, A.H. (2013) 'Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems', *Engineering with Computers*, Vol. 29, No. 1, pp.17–35.

Gao, W.F., Liu, S.Y. and Huang, L.L. (2013) 'A novel artificial bee colony algorithm based on modified search equation and orthogonal learning', *IEEE Transactions on Cybernetics*, Vol. 43, No. 3, pp.1011–1024.

Gao, W.F., Liu, S.Y. and Huang, L.L. (2014) 'Enhancing artificial bee colony algorithm using more information-based search equations', *Information Sciences*, Vol. 270, pp.112–33.

Geem, Z.W., Kim, J.H. and Loganathan, G.V. (2001) 'A new heuristic optimization algorithm: harmony search', *Simulation*, Vol. 76, No. 2, pp.60–68.

Gen, M. and Cheng, R. (1997) *Genetic Algorithms and Engineering Design*, Wiley J, New York.

Gold, S. and Krishnamurty, S. (1997) 'Tradeoffs in robust engineering design', *Proceedings of DETC*, September, Vol. 97.

Gupta, S. and Deep, K. (2018a) 'A novel random walk grey wolf optimizer', S*warm and Evolutionary Computation* [online] https://doi.org/10.1016/j.swevo.2018.01.001.

Gupta, S. and Deep, K. (2018b) 'Random walk grey wolf optimizer for constrained engineering optimization problems', C*omputational Intelligence* [online] https://doi.org/10.1111/coin.12160.

Heidari, A.A. and Pahlavani, P. (2017) 'An efficient modified grey wolf optimizer with Lévy flight for optimization tasks', *Applied Soft Computing*, Vol. 60, pp.115–134.

Himmelblau, D.M. (1972) *Applied Nonlinear Programming*, McGraw-Hill Companies.

Holland, J.H. (1992) *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, MIT Press.

Homaifar, A., Qi, C.X. and Lai, S.H. (1994) 'Constrained optimization via genetic algorithms', *Simulation*, Vol. 62, No. 4, pp.242–253.

Jayakumar, N., Subramanian, S., Ganesan, S. and Elanchezhian, E.B. (2016) 'Grey wolf optimization for combined heat and power dispatch with cogeneration systems', *International Journal of Electrical Power & Energy Systems*, Vol. 74, pp.252–264.

Kamboj, V.K., Bath, S.K. and Dhillon, J.S. (2016) 'Solution of non-convex economic load dispatch problem using grey wolf optimizer', *Neural Computing and Applications*, Vol. 27, No. 5, pp.1301–1316.

Kannan, B.K. and Kramer, S.N. (1994) 'An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design', *Journal of Mechanical Design*, Vol. 116, No. 2, pp.405–411.

Karaboga, D. and Basturk, B. (2007) 'A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm', *Journal of Global Optimization*, Vol. 39, No. 3, pp.459–471.

Kohli, M. and Arora, S. (2018) 'Chaotic grey wolf optimization algorithm for constrained optimization problems', *Journal of Computational Design and Engineering*, Vol. 5, No. 4, pp.458–472.

Kumar, V. and Kumar, D. (2017) 'An astrophysics-inspired Grey wolf algorithm for numerical optimization and its application to engineering design problems', *Advances in Engineering Software*, Vol. 112, pp.231–254.

Lee, K.S. and Geem, Z.W. (2005) 'A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice', *Computer Methods in Applied Mechanics and Engineering*, Vol. 194, Nos. 36–38, pp.3902–3933.

Liang, J.J., Qin, A.K., Suganthan, P.N. and Baskar, S. (2006) 'Comprehensive learning particle swarm optimizer for global optimization of multimodal functions', *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 3, pp.281–295.

Long, W., Jiao, J., Liang, X. and Tang, M. (2018) 'An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization', *Engineering Applications of Artificial Intelligence*, Vol. 68, pp.63–80.

Mirjalili, S. (2015) 'Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm', *Knowledge-Based Systems*, Vol. 89, pp.228–249.

Mirjalili, S. (2016) 'SCA: a sine cosine algorithm for solving optimization problems', *Knowledge-Based Systems*, Vol. 96, pp.120–33.

Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H. and Mirjalili, S.M. (2017) 'Salp swarm algorithm: a bio-inspired optimizer for engineering design problems', *Advances in Engineering Software*, Vol. 114, pp.163–191.

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

Mittal, N., Singh, U. and Sohi, B.S. (2016) 'Modified grey wolf optimizer for global engineering optimization', *Applied Computational Intelligence and Soft Computing*, Vol. 8.

Muangkote, N., Sunat, K. and Chiewchanwattana, S. (2014) 'An improved grey wolf optimizer for training q-Gaussian radial basis functional-link nets', *International Computer Science and Engineering Conference (ICSEC)*, IEEE, pp.209–214.

Muro, C., Escobedo, R., Spector, L. and Coppinger, R.P. (2011) 'Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations', *Behavioural Processes*, Vol. 88, No. 3, pp.192–197.

Rashedi, E., Nezamabadi-Pour, H. and Saryazdi, S. (2009) 'GSA: a gravitational search algorithm', *Information Sciences*, Vol. 179, No. 13, pp.2232–2248.

Ravi, V., Murty, B.S. and Reddy, J. (1997) 'Nonequilibrium simulated-annealing algorithm applied to reliability optimization of complex systems', *IEEE Transactions on Reliability*, Vol. 46, No. 2, pp.233–239.

Rodríguez, L., Castillo, O., Soria, J., Melin, P., Valdez, F., Gonzalez, C.I., Martinez, G.E. and Soto, J. (2017) 'A fuzzy hierarchical operator in the grey wolf optimizer algorithm', *Applied Soft Computing*, Vol. 57, pp.315–328.

Sadollah, A., Bahreininejad, A., Eskandar, H. and Hamdi, M. (2013) 'Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems', *Applied Soft Computing*, Vol. 13, No. 5, pp.2592–2612.

Sandgren, E. (1990) 'Nonlinear integer and discrete programming in mechanical design optimization', *Journal of Mechanical Design*, Vol. 112, No. 2, pp.223–229.

Saremi, S., Mirjalili, S.Z. and Mirjalili, S.M. (2015) 'Evolutionary population dynamics and grey wolf optimizer', *Neural Computing and Applications*, Vol. 25, No. 5, pp.1257–1263.

Shelokar, P.S., Jayaraman, V.K. and Kulkarni, B.D. (2002) 'Ant algorithm for single and multiobjective reliability optimization problems', *Quality and Reliability Engineering International*, Vol. 18, No. 6, pp.497–514.

Shi, Y. and Eberhart, R. (1998) 'A modified particle swarm optimizer', *IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence*, IEEE, pp.69–73.

Simon, D. (2008) 'Biogeography-based optimization', *IEEE Transactions on Evolutionary Computation*, Vol. 12, No. 6, pp.702–713.

Song, H.M., Sulaiman, M.H. and Mohamed, M.R. (2014) 'An application of grey wolf optimizer for solving combined economic emission dispatch problems', *International Review on Modelling and Simulations*, Vol. 7, No. 5, pp.838–844.

Storn, R. and Price, K. 1997) 'Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces', *Journal of Global Optimization*, Vol. 11, No. 4, pp.341–59.

Sulaiman, M.H., Mustaffa, Z., Mohamed, M.R. and Aliman, O. (2015) 'Using the gray wolf optimizer for solving optimal reactive power dispatch problem', *Applied Soft Computing*, Vol. 32, pp.286–292.

Sultana, U., Khairuddin, A.B., Mokhtar, A.S., Zareen, N. and Sultana, B. (2016) 'Grey wolf optimizer based placement and sizing of multiple distributed generation in the distribution system', *Energy*, Vol. 111, pp.525–536.

Tillman, F.A., Hwang, C.L. and Kuo, W. (1980) *Optimization of Systems Reliability*, pp.187–191.

Van den Bergh, F. and Engelbrecht, A.P. (2004) 'A cooperative approach to particle swarm optimization', *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, pp.225–239.

Wang, G.G. (2003) 'Adaptive response surface method using inherited Latin hypercube design points', *Journal of Mechanical Design*, Vol. 125, No. 2, pp.210–220.

Wolpert, D.H. and Macready, W.G. (1995) *No Free Lunch Theorems for Search*, Technical Report SFITR-95-02-010, Santa Fe Institute; July.

Yang, X.S. and Deb, S. (2009) 'Cuckoo search via Lévy flights', '*World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*, IEEE, pp.210–214.

Zhang, S., Zhou, Y. Li, Z. and Pan, W. (2016) 'Grey wolf optimizer for unmanned combat aerial vehicle path planning', *Advances in Engineering Software*, Vol. 99, pp.121–36.