Hindawi Publishing Corporation
Applied Computational Intelligence and Soft Computing
Volume 2016, Article ID 7950348, 16 pages
http://dx.doi.org/10.1155/2016/7950348

Research Article

Modified Grey Wolf Optimizer for Global Engineering Optimization

Nitin Mittal,[1] Urvinder Singh,[2] and Balwinder Singh Sohi[1]

[1] *Department of Electronics and Communication Engineering, Chandigarh University, Mohali, Punjab 140413, India*
[2] *Department of Electronics and Communication Engineering, Thapar University, Patiala, Punjab 147004, India*

Correspondence should be addressed to Nitin Mittal; mittal.nitin84@gmail.com

Received 30 November 2015; Accepted 3 April 2016

Academic Editor: Samuel Huang

Nature-inspired algorithms are becoming popular among researchers due to their simplicity and flexibility. The nature-inspired metaheuristic algorithms are analysed in terms of their key features like their diversity and adaptation, exploration and exploitation, and attractions and diffusion mechanisms. The success and challenges concerning these algorithms are based on their parameter tuning and parameter control. A comparatively new algorithm motivated by the social hierarchy and hunting behavior of grey wolves is Grey Wolf Optimizer (GWO), which is a very successful algorithm for solving real mechanical and optical engineering problems. In the original GWO, half of the iterations are devoted to exploration and the other half are dedicated to exploitation, overlooking the impact of right balance between these two to guarantee an accurate approximation of global optimum. To overcome this shortcoming, a modified GWO (mGWO) is proposed, which focuses on proper balance between exploration and exploitation that leads to an optimal performance of the algorithm. Simulations based on benchmark problems and WSN clustering problem demonstrate the effectiveness, efficiency, and stability of mGWO compared with the basic GWO and some well-known algorithms.

1. Introduction

Metaheuristic algorithms are powerful methods for solving many real-world engineering problems. The majority of these algorithms have been derived from the survival of fittest theory of evolutionary algorithms, collective intelligence of swarm particles, behavior of biological inspired algorithms, and/or logical behavior of physical algorithms in nature.

Evolutionary algorithms are those who mimic the evolutionary processes in nature. The evolutionary algorithms are based on survival of fittest candidate for a given environment. These algorithms begin with a population (set of solutions) which tries to survive in an environment (defined with fitness evaluation). The parent population shares its properties of adaptation to the environment to the children with various mechanisms of evolution such as genetic crossover and mutation. The process continues over a number of generations (iterative process) till the solutions are found to be most suitable for the environment. Some of the evolutionary algorithms are Genetic Algorithm (GA) [1], Evolution Strategies (ES) [2], Genetic Programming (GP) [3], Differential Evolution (DE) [4], and Biogeography-Based Optimization (BBO) [5–9].

The physical algorithms are inspired by physical processes such as heating and cooling of materials (Simulated Annealing [10]), discrete cultural information which is treated as in between genetic and culture evolution (Memetic Algorithm [11]), harmony of music played by musicians (Harmony Search [12, 13]), cultural behavior of frogs (Shuffled Frog-Leaping Algorithm [14]), Gravitational Search algorithm [15], Multiverse Optimizer (MVO) [16], and Chemical Reaction Optimization (CRO) [17].

Swarm intelligence is the group of natural metaheuristics inspired by the “collective intelligence” of swarms. The collective intelligence is built up through a population of homogeneous agents interacting with each other and with their environment. Example of such intelligence is found among colonies of ants, flocks of birds, schools of fish, and so forth. Particle Swarm Optimization [18] is developed based on the swarm behavior of birds. The firefly algorithm [19] is formulated based on the flashing behavior of fireflies. Bat Algorithm (BA) [20] is based on the echolocation behavior of bats.

Ant Colony Optimization (ACO) [21, 22] is inspired by the pheromone trail laying behavior of real ant colonies. A new evolutionary optimization algorithm, Cuckoo Search (CS) Algorithm [23], is inspired by lifestyle of cuckoo birds. The major algorithms include Ant Colony Optimization (ACO) [21, 22], Particle Swarm Optimization (PSO) [18], Artificial Bee Colony (ABC) Algorithm [24], Fish Swarm Algorithm (FSA) [25], Glowworm Swarm Optimization (GSO) [26], Grey Wolf Optimizer (GWO) [27], Fruit Fly Optimization Algorithm (FFOA) [28], Bat Algorithm (BA) [20], Novel Bat Algorithm (NBA) [29], Dragonfly Algorithm (DA) [30], Cat Swarm Optimization (CSO) [31], Cuckoo Search (CS) Algorithm [23], Cuckoo Optimization Algorithm (COA) [32], and Spider Monkey Optimization (SMO) Algorithm [33].

The biologically inspired algorithms comprise natural metaheuristics derived from living phenomena and behavior of biological organisms. The intelligence derived with bioinspired algorithms is decentralized, distributed, self-organizing, and adaptive in nature under uncertain environments. The major algorithms in this field include Artificial Immune Systems (AIS) [34], Bacterial Foraging Optimization (BFO) [35], and Krill Herd Algorithm [36].

Because of their inherent advantages, such algorithms can be applied to various applications including power systems operations and control, job scheduling problems, clustering and routing problems, batch process scheduling, image processing, and pattern recognition problems.

GWO is recently developed heuristics inspired from the leadership hierarchy and hunting mechanism of grey wolves in nature and has been successfully applied for solving economic dispatch problems [37], feature subset selection [38], optimal design of double later grids [39], time forecasting [40], flow shop scheduling problem [41], optimal power flow problem [42], and optimizing key values in the cryptography algorithms [43]. A number of variants are also proposed to improve the performance of basic GWO that include binary GWO [44], a hybrid version of GWO with PSO [45], integration of DE with GWO [46], and parallelized GWO [47, 48].

Every optimization algorithm stated above needs to address the exploration and exploitation of a search space. In order to be successful, an optimization algorithm needs to establish a good ratio between exploration and exploitation. In this paper, a modified GWO (mGWO) is proposed to balance the exploration and exploitation trade-off in original GWO algorithm. Different functions with diverse slopes are employed to tune the parameters of GWO algorithm for varying exploration and exploitation combinations over the course of iterations. Increasing the exploration in comparison to exploitation increases the convergence speed and avoids the local minima trapping effect.

The rest of the paper is organized as follows. Section 2 gives the overview of original GWO. The proposed mGWO algorithm is explained in Section 3. The experimental results are demonstrated in Section 4. Section 5 solves the clustering problem in WSN for cluster head selection to demonstrate the applicability of the proposed algorithm. Finally, Section 6 concludes the paper.

2. Overview of Grey Wolf Optimizer Algorithm

Grey Wolf Optimizer (GWO) is a typical swarm-intelligence algorithm which is inspired from the leadership hierarchy and hunting mechanism of grey wolves in nature. Grey wolves are considered as apex predators; they have average group size of 5–12. In the hierarchy of GWO, alpha (α) is considered the most dominating member among the group. The rest of the subordinates to α are beta (β) and delta (δ) which help to control the majority of wolves in the hierarchy that are considered as omega (ω). The ω wolves are of lowest ranking in the hierarchy.

The mathematical model of hunting mechanism of grey wolves consists of the following:

(i) Tracking, chasing, and approaching the prey.

(ii) Pursuing, encircling, and harassing the prey until it stops moving.

(iii) Attacking the prey.

2.1. Encircling Prey.

2.1. Encircling Prey. Grey wolves encircle the prey during the hunt which can be mathematically written as [27]

$$\begin{aligned}\vec{D} &= \left|\vec{C} \cdot \overrightarrow{X_p}(t) - \vec{X}(t)\right|, \\ \vec{X}(t+1) &= \overrightarrow{X_p}(t) - \vec{A} \cdot \vec{D},\end{aligned} \tag{1}$$

where t indicates the current iteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors, $\overrightarrow{X_p}$ is the position vector of the prey, and $\vec{X}$ indicates the position vector of a grey wolf.

The vectors $\vec{A}$ and $\vec{C}$ are calculated as follows:

$$\begin{aligned}\vec{A} &= 2\vec{a} \cdot \overrightarrow{r_1} - \vec{a}, \\ \vec{C} &= 2 \cdot \overrightarrow{r_2},\end{aligned} \tag{2}$$

where components of a are linearly decreased from 2 to 0 over the course of iterations and r_1 and r_2 are random vectors in $[0, 1]$.

2.2. Hunting. Hunting of prey is usually guided by α and β, and δ will participate occasionally. The best candidate solutions, that is, α, β, and δ, have better knowledge about the potential location of prey. The other search agents (ω) update their positions according to the position of three best search agents. The following formulas are proposed in this regard:

$$\begin{aligned}\overrightarrow{D_\alpha} &= \left|\overrightarrow{C_1} \cdot \overrightarrow{X_\alpha} - \vec{X}\right|, \\ \overrightarrow{D_\beta} &= \left|\overrightarrow{C_2} \cdot \overrightarrow{X_\beta} - \vec{X}\right|, \\ \overrightarrow{D_\delta} &= \left|\overrightarrow{C_3} \cdot \overrightarrow{X_\delta} - \vec{X}\right|,\end{aligned} \tag{3}$$

$$\begin{aligned}\overrightarrow{X_1} &= \overrightarrow{X_\alpha} - \overrightarrow{A_1}\cdot\left(\overrightarrow{D_\alpha}\right),\\ \overrightarrow{X_2} &= \overrightarrow{X_\beta} - \overrightarrow{A_2}\cdot\left(\overrightarrow{D_\beta}\right),\\ \overrightarrow{X_3} &= \overrightarrow{X_\delta} - \overrightarrow{A_3}\cdot\left(\overrightarrow{D_\delta}\right),\end{aligned} \quad (4)$$

$$\vec{X}(t+1) = \frac{\overrightarrow{X_1}(t) + \overrightarrow{X_2}(t) + \overrightarrow{X_3}(t)}{3}. \quad (5)$$

2.3. Attacking Prey. In order to mathematically model for approaching the prey, we decrease the value of $\vec{a}$. The fluctuation range of $\vec{A}$ is also decreased by $\vec{a}$. $\vec{A}$ is a random value in the interval $[-a, a]$ where a is decreased linearly from 2 to 0 over the course of iterations. When random values of $\vec{A}$ are in $[-1, 1]$, the next position of a search agent can be in any position between its current position and the position of the prey. The value $|A| < 1$ forces the wolves to attack the prey.

After the attack again they search for the prey in the next iteration, wherein they again find the next best solution α among all wolves. This process repeats till the termination criterion is fulfilled.

3. Modified GWO Algorithm

Finding the global minimum is a common, challenging task among all minimization methods. In population-based optimization methods, generally, the desirable way to converge towards the global minimum can be divided into two basic phases. In the early stages of the optimization, the individuals should be encouraged to scatter throughout the entire search space. In other words, they should try to explore the whole search space instead of clustering around local minima. In the latter stages, the individuals have to exploit information gathered to converge on the global minimum. In GWO, with fine-adjusting of the parameters a and A, we can balance these two phases in order to find global minimum with fast convergence speed.

Although different improvements of individual-based algorithms promote local optima avoidance, the literature shows that population-based algorithms are better in handling this issue. Regardless of the differences between population-based algorithms, the common approach is the division of optimization process to two conflicting milestones: exploration versus exploitation. The exploration encourages candidate solutions to change abruptly and stochastically. This mechanism improves the diversity of the solutions and causes high exploration of the search space. In contrast, the exploitation aims for improving the quality of solutions by searching locally around the obtained promising solutions in the exploration. In this milestone, candidate solutions are obliged to change less suddenly and search locally.

Exploration and exploitation are two conflicting milestones where promoting one results in degrading the other. A right balance between these two milestones can guarantee a very accurate approximation of the global optimum using population-based algorithms. On the one hand, mere exploration of the search space prevents an algorithm from finding an accurate approximation of the global optimum. On the other hand, mere exploitation results in local optima stagnation and again low quality of the approximated optimum.

In GWO, the transition between exploration and exploitation is generated by the adaptive values of a and A. In this, half of the iterations are devoted to exploration ($|A| \geq 1$) and the other half are used for exploitation ($|A| < 1$), as shown in Figure 1(a). Generally, higher exploration of search space results in lower probability of local optima stagnation. There are various possibilities to enhance the exploration rate as shown in Figure 1(b), in which exponential functions are used instead of linear function to decrease the value of a over the course of iterations. Too much exploration is similar to too much randomness and will probably not give good optimization results. But too much exploitation is related to too little randomness. Therefore, there must be a balance between exploration and exploitation.

In GWO, the value of a decreases linearly from 2 to 0 using the update equation as follows:

$$a = 2\left(1 - \frac{t}{T}\right), \quad (6)$$

where T indicates the maximum number of iterations and t is the current iteration. Our mGWO employs exponential function for the decay of a over the course of iterations. Consider

$$a = 2\left(1 - \frac{t^2}{T^2}\right) \quad (7)$$

as shown in Figure 1(c). Using this exponential decay function, the numbers of iterations used for exploration and exploitation are 70% and 30%, respectively.

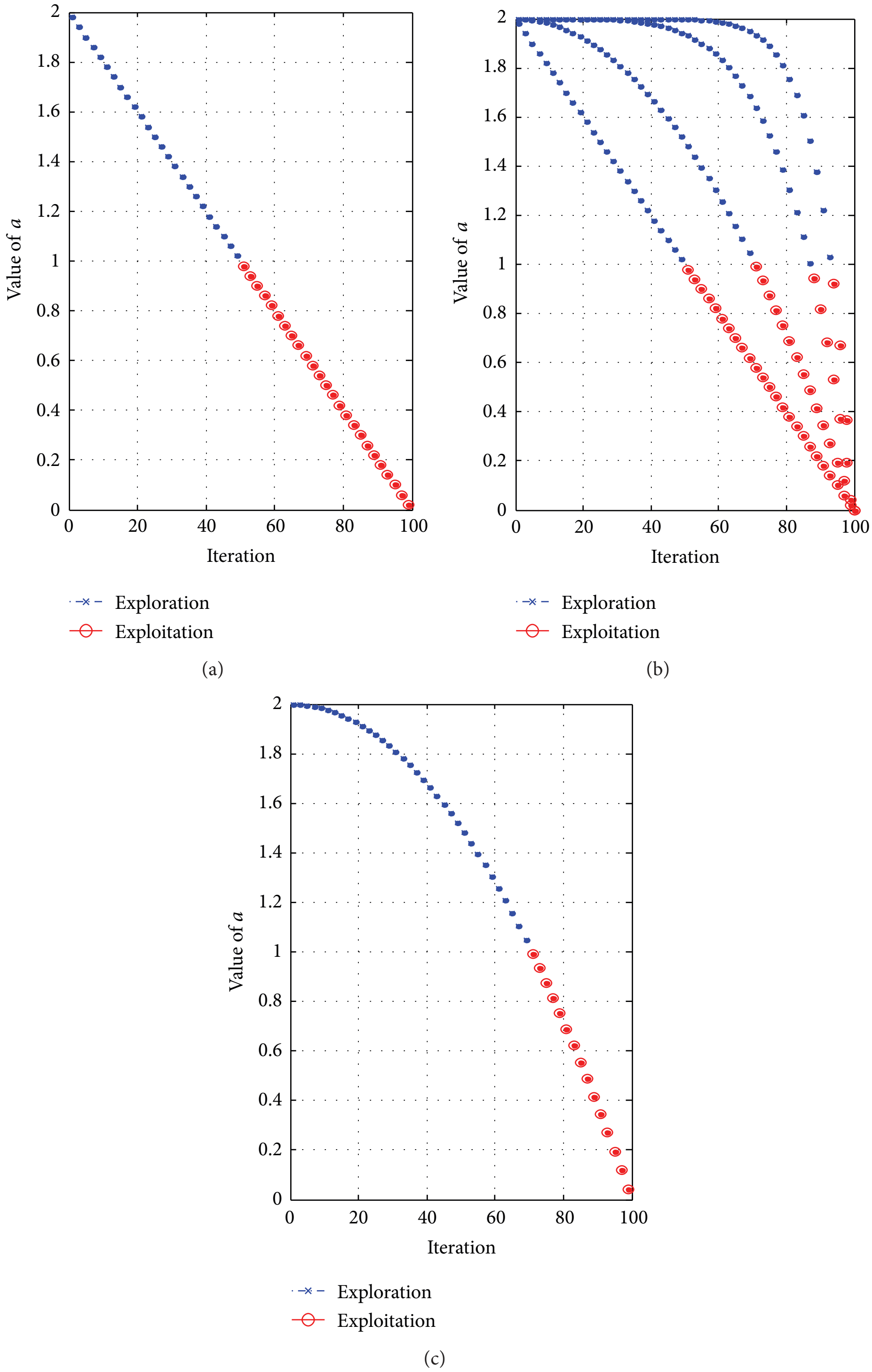
The pseudocode of mGWO is given in Algorithm 1.

4. Results and Discussion

This section investigates the effectiveness of mGWO in practice. It is common in this field to benchmark the performance of algorithms on a set of mathematical functions with known global optima. We also follow the same process and employ 27 benchmark functions for comparison. The test functions are divided to four groups: unimodal, multimodal, fixed-dimension multimodal, and composite benchmark functions. The unimodal functions (F_1–F_7) are suitable for benchmarking the exploitation of algorithms since they have one global optimum and no local optima. On the contrary, multimodal functions (F_8–F_{13}) have a large number of local optima and are helpful to examine exploration and local optima avoidance of algorithms.

The mathematical formulation of the employed test functions is presented in Tables 1–4. We consider 30 variables for unimodal and multimodal test function for further improving their difficulties.

Since heuristic algorithms are stochastic optimization techniques, they have to be run at least more than 10 times to generate meaningful statistical results. It is again a common strategy that an algorithm is run on a problem



(a)

(b)

(c)

Figure 1: (a) Updating the value of a for GWO, (b) some samples of possible functions for updating a over the course of iterations, and (c) updating the value of a over the course of iterations for mGWO.

m times and average/standard deviation/median of the best obtained solution in the last iteration are calculated as the metrics of performance. We follow the same method to generate and report the results over 30 independent runs. In order to verify the performance of mGWO algorithm, PSO, BA, CS, and GWO algorithms are chosen. Note that we utilized 30 search agents and 3000 iterations for each of the algorithms.

The convergence curves of unimodal, multimodal, fixed-dimension multimodal, and composite benchmark functions for the competitive optimization algorithms are given in Figures 2, 3, 4, and 5, respectively. As Table 5 shows,

```
Initialize the search agent (grey wolf) population $X_i$ ($i = 1, 2, \ldots, n$)
Initialize $a$, $A$ and $C$
Calculate the fitness of each search agent
$X_\alpha$ = the best (or dominating) search agent
$X_\beta$ = the second best search agent
$X_\delta$ = the third best search agent
while ($t <$ Maximum number of iterations)
        for each search agent
                update the position of the current search agent by (5)
        end for
        update $a$ by (7)
        update $A$ and $C$ by (2)
        calculate the fitness of all search agents
        update $X_\alpha$, $X_\beta$ and $X_\delta$
        $t = t + 1$
end while
Return $X_\alpha$
```

Algorithm 1: Pseudocode of mGWO algorithm.

Table 1: Unimodal benchmark functions.

Function	Dim	Range	$f_{\min}$
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	$[-100, 100]$	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	$[-10, 10]$	0
$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j \right)^2$	30	$[-100, 100]$	0
$F_4(x) = \max_i \{ \lvert x_i \rvert, 1 \le i \le n \}$	30	$[-100, 100]$	0
$F_5(x) = \sum_{i=1}^{n-1} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + \left(x_i - 1 \right)^2 \right]$	30	$[-30, 30]$	0
$F_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	$[-100, 100]$	0
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + \text{random}(0, 1)$	30	$[-1.28, 1.28]$	0

mGWO algorithm provides the best results in 5 out of 7 unimodal benchmark test functions. The mGWO algorithm also provides very competitive results compared to CS on F_5 and F_6. As discussed above, unimodal functions are suitable for benchmarking exploitation of the algorithms. Therefore, these results evidence high exploitation capability of the mGWO algorithm.

The statistical results of the algorithms on multimodal test function are presented in Table 6. It may be seen that mGWO algorithm highly outperforms other algorithms on F_9, F_{10}, F_{11}, and F_{12}. It should be noted that mGWO algorithm outperforms other algorithms on these multimodal test functions except PSO for F_{13}. The results of multimodal test function strongly prove that high exploration of mGWO algorithm is a suitable mechanism for avoiding local solutions. Since the multimodal functions have an exponential number of local solutions, the results show that mGWO algorithm is able to explore the search space extensively and find promising regions of the search space. In addition, high local optima avoidance of this algorithm is another finding that can be inferred from these results.

The rest of the results, which belong to F_{14}–F_{23} and F_{24}–F_{27}, are provided in Tables 7 and 8, respectively. The results are consistent with those of other test functions, in which mGWO shows very competitive results compared to other algorithms.

5. Cluster Head Selection in WSN Using mGWO

Cluster head (CH) selection problem is a well-known problem in the field of wireless sensor networks (WSNs) in which the energy consumption cost of the network should be minimized [49–53]. In this paper, this problem is solved using mGWO algorithm and compared with GA, PSO, BA, CS, and GWO.

The main challenges in designing and planning the operations of WSNs are to optimize energy consumption and prolong network lifetime. Cluster-based routing techniques, such as the well-known low-energy adaptive clustering hierarchy (LEACH) [50], are used to achieve scalable solutions and extend the network lifetime until the last node dies (LND). In order to achieve prolonged network lifetime in cluster-based routing techniques, the lifetime of the CHs plays an important role. Improper cluster formation may cause some CHs to be overloaded. Such overload may cause high energy consumption of the CH and degrade the overall performance of the WSN. Therefore, proper CH selection is the most important issue for clustering sensor nodes. Designing an energy efficient clustering algorithm is not an easy task. Therefore, nature-inspired optimization algorithms may be applied to tackle cluster-based routing problem in WSN. Evolutionary algorithms (EAs) have been used in recent years as metaheuristics to address energy-aware routing challenges by designing intelligent models

Table 2: Multimodal benchmark functions.

Function	Dim	Range	$f_{\min}$
$F_8(x) = \sum_{i=1}^{n} - x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	30	$[-500, 500]$	-418.9829×5
$F_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]$	0
$F_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	30	$[-32, 32]$	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]$	0
$F_{12}(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4},\ u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	$[-50, 50]$	0
$F_{13}(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2\left[1 + \sin^2(3\pi x_i + 1)\right] + (x_n - 1)^2\left[1 + \sin^2(2\pi x_n)\right]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	$[-50, 50]$	0

Table 3: Fixed-dimension multimodal benchmark functions.

Function	Dim	Range	$f_{\min}$
$F_{14}(x) = \left(\dfrac{1}{500} + \sum_{j=1}^{25} \dfrac{1}{j + \sum_{i=1}^{2}\left(x_i - a_{ij}\right)^6}\right)^{-1}$	2	$[-65, 65]$	1
$F_{15}(x) = \sum_{i=1}^{11}\left[a_i - \dfrac{x_1\left(b_i^2 + b_i x_2\right)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	$[-5, 5]$	0.00030
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \dfrac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	−1.0316
$F_{17}(x) = \left(x_2 - \dfrac{5.1}{4\pi^2}x_1^2 + \dfrac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \dfrac{1}{8\pi}\right)\cos x_1 + 10$	2	$[-5, 5]$	0.398
$F_{18}(x) = \left[1 + \left(x_1 + x_2 + 1\right)^2\left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right]$ $\cdot\left[30 + \left(2x_1 - 3x_2\right)^2\left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)\right]$	2	$[-2, 2]$	3
$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	3	$[1, 3]$	−3.86
$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	6	$[0, 1]$	−3.32
$F_{21}(x) = -\sum_{i=1}^{5}\left[\left(X - a_i\right)\left(X - a_i\right)^T + c_i\right]^{-1}$	4	$[0, 10]$	−10.1532
$F_{22}(x) = -\sum_{i=1}^{7}\left[\left(X - a_i\right)\left(X - a_i\right)^T + c_i\right]^{-1}$	4	$[0, 10]$	−10.4028
$F_{23}(x) = -\sum_{i=1}^{10}\left[\left(X - a_i\right)\left(X - a_i\right)^T + c_i\right]^{-1}$	4	$[0, 10]$	−10.5363

that collaborate together to optimize an appropriate energy-aware objective function [52]. GWO is one of the powerful heuristics that can be applied for efficient load balanced clustering. In this paper, mGWO based clustering algorithm is used to solve the abovementioned load balancing problem. The algorithm forms clusters in such a way that the overall energy consumption of the network is minimized. Total energy consumption in the network is the sum of the total energy dissipated from the non-CHs to send information to their respective CHs and the total energy consumed by CH nodes to aggregate the information and send it to the base station (BS).

Consider a WSN of n sensor nodes randomly deployed in the sensing field and organized into K clusters: $C_1, C_2, \ldots, C_K$. The fitness function for the energy consumption may be defined as

$$f = \left(\sum_{i=1}^{K}\sum_{s\in C_i} E_{\mathrm{TX}_{s,\mathrm{CH}_i}} + E_{\mathrm{RX}} + E_{\mathrm{DA}}\right) + \sum_{i=1}^{K} E_{\mathrm{TX}_{\mathrm{CH}_i,\mathrm{BS}}}, \tag{8}$$

where K is the total number of CHs, $s \in C_i$ is a non-CH associated with the ith CH, and $E_{\mathrm{TX}_{\mathrm{node}_1,\mathrm{node}_2}}$ is the energy dissipated for transmitting data from node_1 to node_2.

In order to calculate radio energy transmission and reception costs, a k-bit message and also the transmitter-receiver separation distance d are given by

$$E_{\mathrm{TX}} = \begin{cases} kE_{\mathrm{elec}} + k\varepsilon_{\mathrm{friss\_amp}}d^2, & \text{if } d < d_0 \\ kE_{\mathrm{elec}} + k\varepsilon_{\mathrm{two\_ray\_amp}}d^4, & \text{if } d \geq d_0. \end{cases} \tag{9}$$

The term E_{elec} denotes the per-bit energy dissipation during transmission. The per-bit amplification energy is proportional to d^4 when the transmission distance exceeds the threshold d_0 (called crossover distance) and otherwise is proportional to d^2. The parameters $\varepsilon_{\mathrm{friss\_amp}}$ and $\varepsilon_{\mathrm{two\_ray\_amp}}$ denote transmitter amplification parameters for free-space and multipath fading models, respectively. The value of d_0 is given by

$$d_0 = \sqrt{\frac{\varepsilon_{\mathrm{friss\_amp}}}{\varepsilon_{\mathrm{two\_ray\_amp}}}}. \tag{10}$$

The reception energy of the k-bit data message can be expressed by

$$E_{\mathrm{RX}} = kE_{\mathrm{elec}}, \tag{11}$$

where E_{elec} denotes the per-bit energy dissipation during reception.

E_{DA} is the data aggregation energy expenditure and is set as $E_{\mathrm{DA}} = 5\,\mathrm{nj/bit}$. The values of other parameters are set to $E_{\mathrm{elec}} = 50\,\mathrm{nj/bit}$, $\varepsilon_{\mathrm{friss\_amp}} = 100\,\mathrm{pj/bit/m^2}$, and $\varepsilon_{\mathrm{two\_ray\_amp}} = 0.0013\,\mathrm{pj/bit/m^4}$, respectively [51].

For the simulation setup, 100 nodes are randomly deployed in a 100 m × 100 m area of the sensing field. BS is placed at the center of the field. The initial energy of all homogeneous nodes is set to $E_0 = 1\,\mathrm{J}$. During this analysis, three parameters, namely, first node dead (FND), half nodes dead (HND), and last node dead (LND) are employed to outline the network lifetime.

Table 4: Composite benchmark functions.

Function	Dim	Range	$f_{\min}$
F_{24} (CF1): f_1, f_2 = Rastrigin's function f_3, f_4 = Weierstrass' function f_5, f_6 = Griewank's function f_7, f_8 = Ackley's function f_9, f_{10} = sphere function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [1/5, 1/5, 5/0.5, 5/0.5, 5/100, 5/100, 5/32, 5/32, 5/100, 5/100]$	10	$[-5, 5]$	0
F_{25} (CF2): f_1, f_2 = Ackley's function f_3, f_4 = Rastrigin's function f_5, f_6 = sphere function f_7, f_8 = Weierstrass's function f_9, f_{10} = Griewank's function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1, 2, 1.5, 1.5, 1, 1, 1.5, 1.5, 2, 2]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [2 * 5/32, 5/32, 2 * 1, 1, 2 * 5/100, 5/100, 2 * 10, 10, 2 * 5/60, 5/60]$	10	$[-5, 5]$	0
F_{26} (CF3): f_1, f_2 = expanded Schaffer Rosenbrock's function f_3, f_4 = Rastrigin's function f_5, f_6 = expanded Griewank's and Rosenbrock's function f_7, f_8 = Weierstrass's function f_9, f_{10} = Griewank's function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, 1, 1, 2, 2, 2, 2, 2]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [5 * 5/100, 5/100, 5 * 1, 1, 5 * 1, 1, 5 * 10, 10, 5 * 5/200, 5/200]$	10	$[-5, 5]$	0
F_{27} (CF4): f_1 = Weierstrass's function f_2 = expanded Schaffer Rosenbrock's function f_3 = expanded Griewank's and Rosenbrock's function f_4 = Ackley's function f_5 = Rastrigin's function f_6 = Griewank's function f_7 = expanded Schaffer Rosenbrock's noncont function f_8 = Rastrigin's noncont function f_9 = elliptic function f_{10} = sphere noise function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [2, 2, 2, 2, 2, 2, 2, 2, 2, 2]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [10, 5/20, 1, 5/32, 1, 5/100, 5/50, 1, 5/100, 5/100]$	10	$[-5, 5]$	0

Table 5: Results of unimodal benchmark functions.

F	PSO		BA		CS		GWO		mGWO	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
F_1	$5.31E-11$	$9.76E-11$	22546.833	6668.3781	$2.89E-16$	$3.33E-16$	$2.65E-155$	$1.18E-154$	**6.44E − 205**	**0.00E + 00**
F_2	$5.67E-07$	$1.26E-06$	2864.8141	9441.2388	$5.92E-09$	$5.43E-09$	$7.23E-92$	$8.41E-92$	**3.34E − 119**	**4.95E − 119**
F_3	3228.8636	1932.4945	61977.294	26054.622	1.0408052	0.9448328	$1.78E-42$	$4.14E-42$	**2.74E − 52**	**1.19E − 51**
F_4	40.759779	16.786821	56.421446	10.918884	10.803036	3.0246624	$5.86E-37$	$2.57E-36$	**1.20E − 51**	**4.25E − 51**
F_5	62.716929	49.490404	18305153	12467683	**22.524975**	15.539303	26.799858	**0.5767914**	26.900044	0.852116
F_6	$2.89E-11$	$4.27E-11$	19775.038	7361.6353	**4.99E − 16**	**1.30E − 15**	0.7070627	0.2983174	0.7862954	0.2449287
F_7	0.0542038	0.0175609	5.0503673	1.6183878	0.0466985	0.0214709	0.0003805	0.0002072	**0.0002609**	**0.000176**

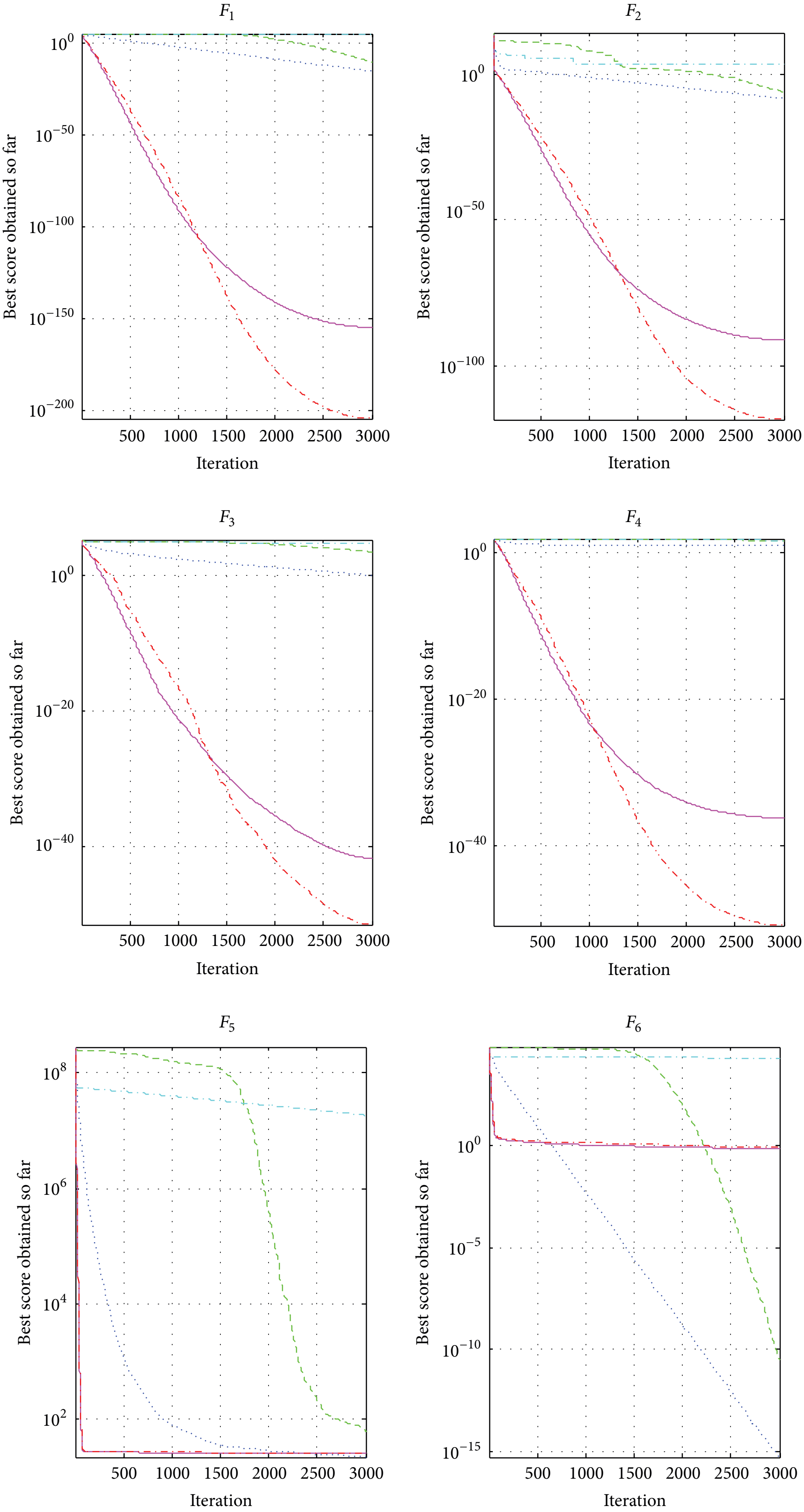


Figure 2: Continued.

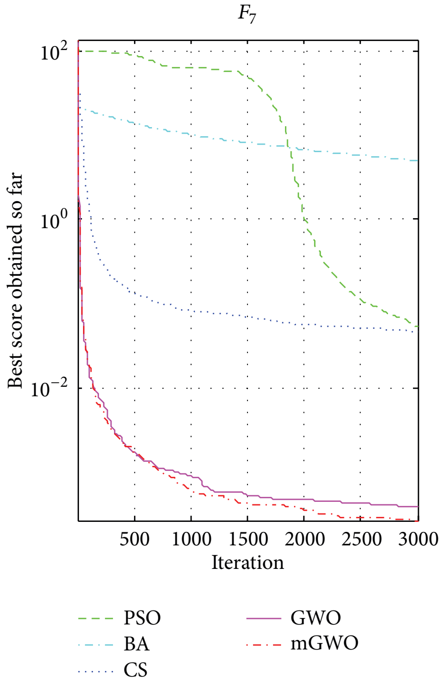


Figure 2: Convergence graph of unimodal benchmark functions.

Table 6: Results of multimodal benchmark functions.

F	PSO		BA		CS		GWO		mGWO	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
F_8	—	—	—	—	**−9958.7573**	**382.10737**	−5697.6258	1036.3742	−5712.462	924.47382
F_9	65.433009	14.197664	96.065525	32.511237	33.785055	6.9716023	**0**	**0**	**0**	**0**
F_{10}	10.769295	9.1708499	17.231924	1.1118681	0.9966469	0.4914999	8.88E − 15	2.27E − 15	**7.82E − 15**	**7.94E − 16**
F_{11}	0.0173265	0.0205398	237.38699	81.345844	0.0008613	0.0038519	0.0025395	0.0066692	**0**	**0**
F_{12}	0.3185579	0.5824145	18644754	18385611	0.1635701	0.4113459	0.0478581	**0.0208711**	**0.0469178**	0.0216789
F_{13}	**0.020326**	**0.0488385**	76481054	53670684	0.3009574	1.1566959	0.6303231	0.1787968	0.6051939	0.1740609

Table 7: Results of fixed-dimension multimodal functions.

F	PSO		BA		CS		GWO		mGWO	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
F_{14}	**0.9980038**	**0**	9.7544907	7.0552637	**0.9980038**	**0**	4.1298947	4.2619933	4.8165637	4.3097562
F_{15}	0.0011042	0.0003066	0.0032289	0.005698	**0.0003075**	**1.11E − 19**	0.0044241	0.0081813	0.007327	0.0098145
F_{16}	−1.0316285	2.28E − 16	−0.9908202	0.1825	−1.0316285	**2.28E − 16**	−1.0316285	1.75E − 09	**−1.0316284**	4.86E − 09
F_{17}	0.3978874	0	**0.3978874**	**1.34E − 10**	**0.3978874**	0	0.3978978	4.63E − 05	0.3978876	5.08E − 07
F_{18}	**3**	**8.15E − 16**	12.45	20.119315	**3**	1.39E − 15	7.0500022	18.11215	3.0000017	2.87E − 06
F_{19}	**−3.8627821**	**2.24E − 15**	**−3.8627821**	1.72E − 08	**−3.8627821**	2.28E − 15	−3.862402	0.0016777	−3.8611515	0.00319
F_{20}	−3.2625486	0.0609909	−3.2566017	0.0606853	**−3.3219952**	**5.19E − 16**	−3.1770991	0.26433	−3.2734635	0.0609917
F_{21}	−6.5146454	3.1931088	−5.7607751	3.1257537	**−10.1532**	**3.65E − 15**	−9.6433311	1.569109	−9.2742029	2.1934501
F_{22}	−7.9202597	3.4833672	−5.2362723	2.7999111	**−10.402941**	**1.95E − 15**	−10.402879	4.66E − 05	−10.136952	1.1884781
F_{23}	−7.8374538	3.4395208	−5.9516938	3.8940879	**−10.53641**	**1.95E − 15**	−9.724888	2.4976263	−10.536237	0.0001251

Table 9 shows the best results obtained for the CH selection problem in WSN. The results of Table 9 show that mGWO algorithm is able to find the best results compared to other algorithms. The results of mGWO are closely followed by the CS and GWO algorithms.

6. Conclusion

This paper proposed a modification to the Grey Wolf Optimizer named mGWO, inspired by the hunting behavior of grey wolves in nature. An exponential decay function is used to balance the exploration and exploitation in the search

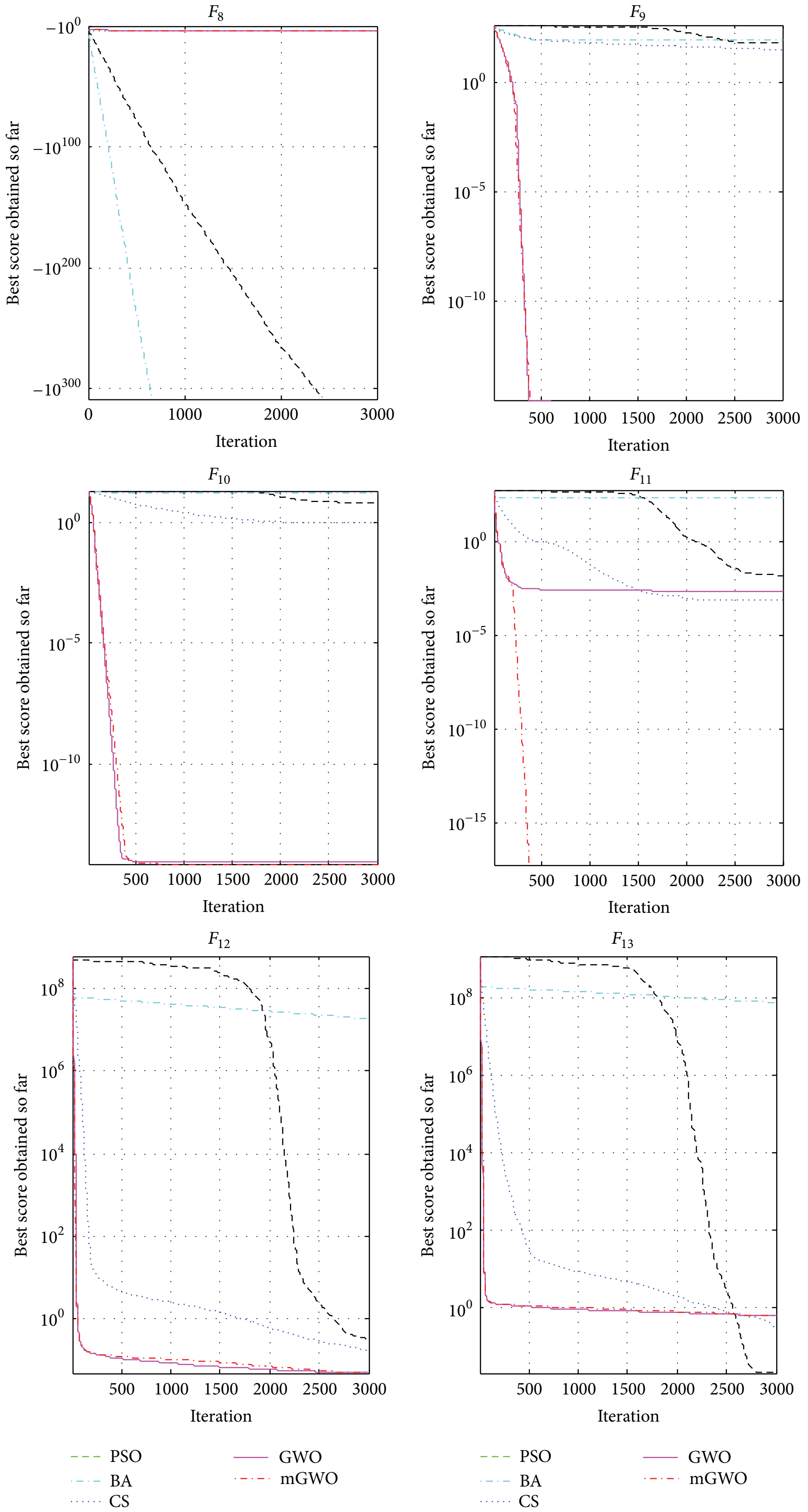


Figure 3: Convergence graph of multimodal benchmark functions.

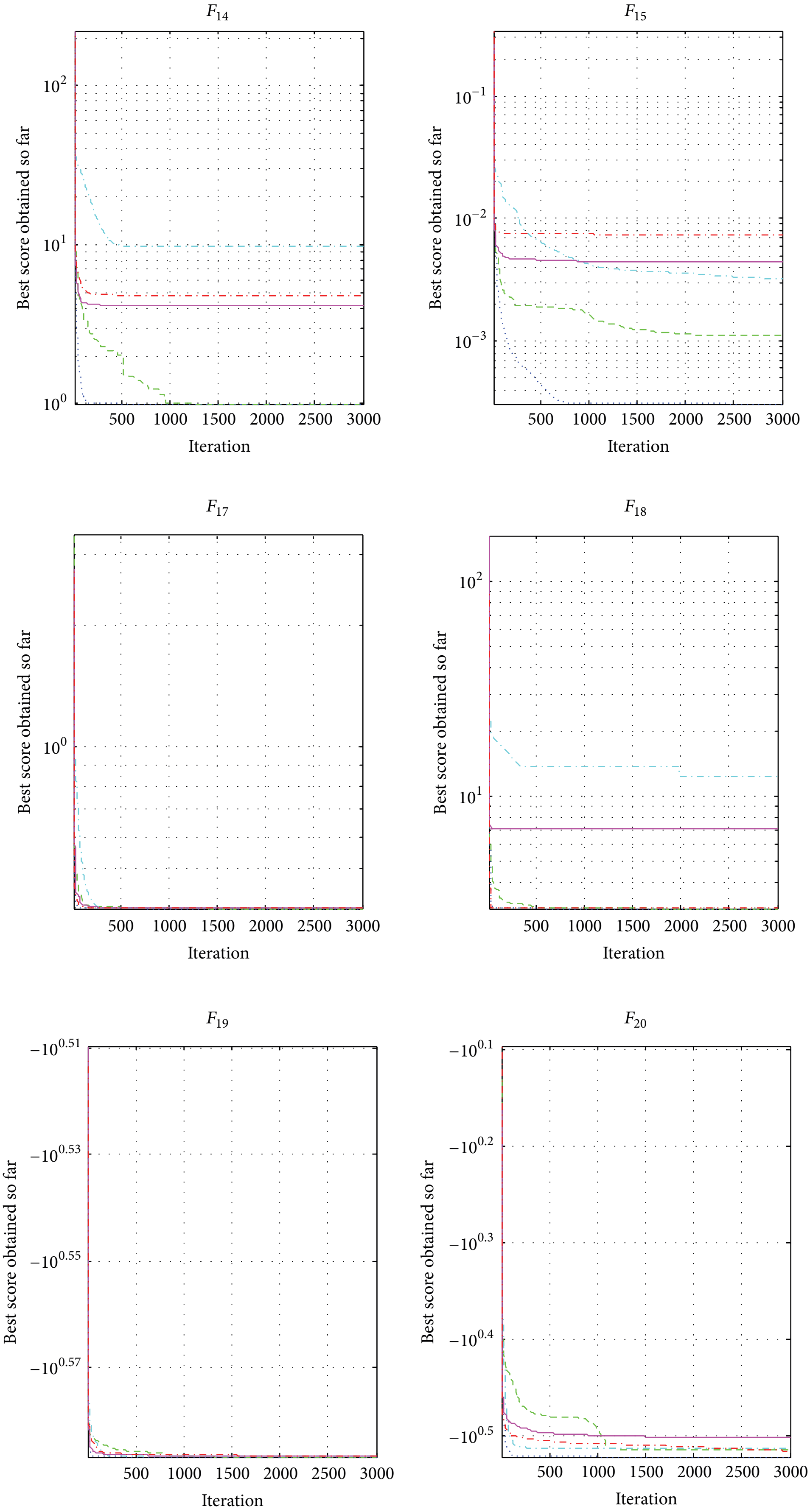


Figure 4: Continued.

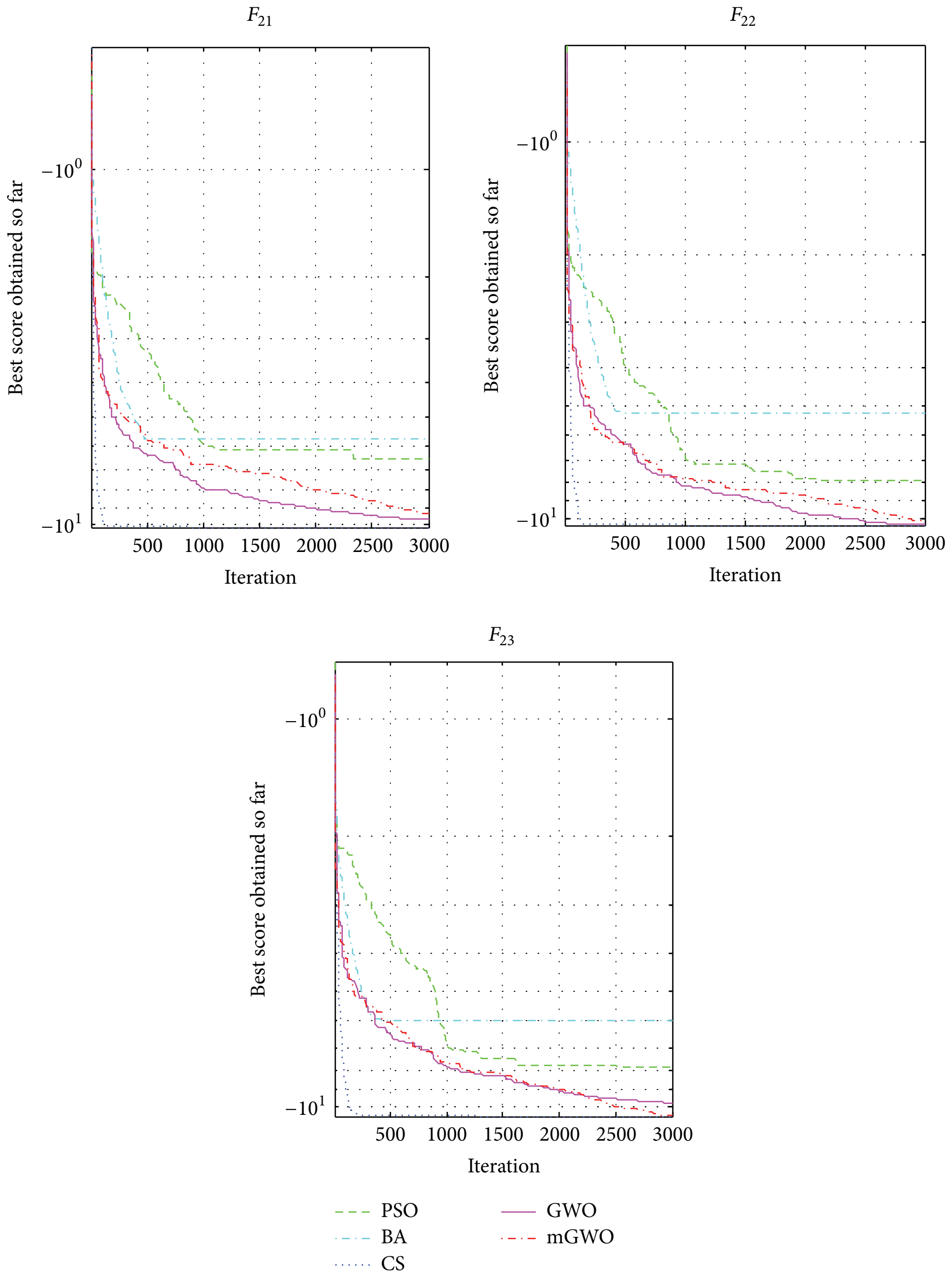


Figure 4: Convergence graph of fixed-dimension multimodal functions.

Table 8: Results of composite functions.

F	PSO		BA		CS		GWO		mGWO	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
F_{24}	375.1142	149.1547	774.141	269.3536	413.6706	177.65048	**337.9168**	**173.3873**	371.8084	194.0954
F_{25}	167.0097	31.86513	490.7166	180.3477	132.158	24.5817	133.9568	**23.17741**	**132.3655**	25.86431
F_{26}	955.6119	280.8871	1476.118	**117.9418**	915.0283	262.3901	943.7718	277.9337	**911.9168**	319.8876
F_{27}	437.6602	183.1154	1328.464	**177.0158**	395	267.08204	559.3846	301.082	**390.904**	279.606

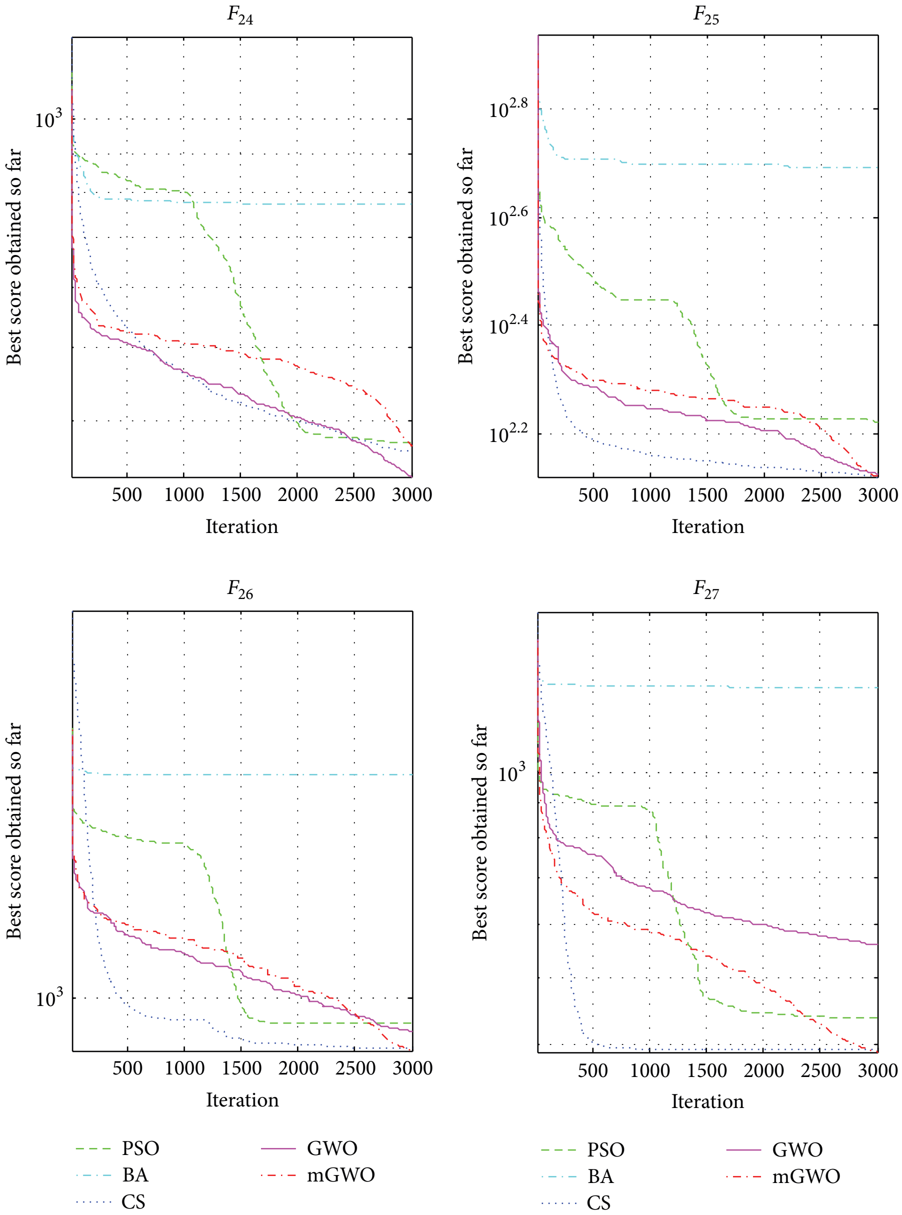


Figure 5: Convergence graph of composite functions.

Table 9: Comparison results of CH selection problem in WSN.

Algorithm	Optimal cost of FF	FND	HND	LND
GA	39.798 mJ	2047.1	2678.2	3337.5
PSO	39.854 mJ	2034.8	2665.2	3324.6
BA	40.324 mJ	1996.7	2654.8	3298.9
CS	38.412 mJ	2079.2	2702.5	3372.4
GWO	38.560 mJ	2073.9	2696.3	3369.8
mGWO	38.209 mJ	2084.6	2714.4	3382.2

space over the course of iterations. The results proved that the proposed algorithm benefits from high exploration in comparison to the standard GWO.

The paper also considered the clustering problem in WSN in which the CH selection is performed using the proposed mGWO algorithm, which is a challenging and NP hard problem. The results show that the proposed method is found to be very effective for real-world applications due to fast convergence and fewer chances to get stuck at local minima. It can be concluded that the proposed algorithm is able to outperform the current well-known and powerful algorithms in the literature. The results prove the competence and superiority of mGWO to existing metaheuristic algorithms and it has an ability to become an effective tool for solving real word optimization problems.

Competing Interests

The authors declare that they have no competing interests.

References

[1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.

[2] T. Back, F. Hoffmeister, and H. P. Schwefel, "A survey of evolution strategies," in *Proceedings of the 4th International Conference on Genetic Algorithms*, San Diego, Calif, USA, July 1991.

[3] J. R. Koza, *Genetic Programming: On the Programming of Computers by Natural Selection*, MIT Press, Cambridge, UK, 1992.

[4] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[5] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.

[6] W. Gong, Z. Cai, C. X. Ling, and H. Li, "A real-coded biogeography-based optimization with mutation," *Applied Mathematics and Computation*, vol. 216, no. 9, pp. 2749–2758, 2010.

[7] W. Gong, Z. Cai, and C. X. Ling, "DE/BBO: a hybrid differential evolution with biogeography-based optimization for global numerical optimization," *Soft Computing*, vol. 15, no. 4, pp. 645–665, 2011.

[8] H. Ma and D. Simon, "Blended biogeography-based optimization for constrained optimization," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 3, pp. 517–525, 2011.

[9] U. Singh and T. S. Kamal, "Design of non-uniform circular antenna arrays using biogeography-based optimisation," *IET Microwaves, Antennas and Propagation*, vol. 5, no. 11, pp. 1365–1370, 2011.

[10] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[11] P. Moscato, "On evolution, search, optimization, genetic algorithms and martial arts: towards Memetic Algorithms," Caltech Concurrent Computation Program Report 826, 1989.

[12] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.

[13] K. S. Lee and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 36–38, pp. 3902–3933, 2005.

[14] M. Eusuff, K. Lansey, and F. Pasha, "Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization," *Engineering Optimization*, vol. 38, no. 2, pp. 129–154, 2006.

[15] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.

[16] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: a nature-inspired algorithm for global optimization," *Neural Computing & Applications*, vol. 27, no. 2, pp. 495–513, 2016.

[17] A. Y. S. Lam and V. O. K. Li, "Chemical-reaction-inspired metaheuristic for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 381–399, 2010.

[18] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Network*, vol. 4, pp. 1942–1948, Perth, Australia, December 1995.

[19] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimization," *International Journal of Bio-Inspired Computation*, vol. 2, no. 2, pp. 78–84, 2010.

[20] X.-S. Yang and A. H. Gandomi, "Bat algorithm: a novel approach for global engineering optimization," *Engineering Computations*, vol. 29, no. 5, pp. 464–483, 2012.

[21] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.

[22] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.

[23] X.-S. Yang and S. Deb, "Engineering optimisation by cuckoo search," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 1, no. 4, pp. 330–343, 2010.

[24] D. Karaboga and B. Akay, "A comparative study of artificial Bee colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, 2009.

[25] X. Li, Z. Shao, and J. Qian, "An optimizing method base on autonomous animates: fish swarm algorithm," *Systems Engineering—Theory & Practice*, vol. 22, pp. 32–38, 2002.

[26] K. Krishnanand and D. Ghose, "Glowworm swarm optimisation: a new method for optimising multi-modal functions," *International Journal of Computational Intelligence Studies*, vol. 1, no. 1, pp. 93–119, 2009.

[27] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[28] W.-T. Pan, "A new fruit fly optimization algorithm: taking the financial distress model as an example," *Knowledge-Based Systems*, vol. 26, pp. 69–74, 2012.

[29] X.-B. Meng, X. Z. Gao, Y. Liu, and H. Zhang, "A novel bat algorithm with habitat selection and Doppler effect in echoes for optimization," *Expert Systems with Applications*, vol. 42, no. 17-18, pp. 6350–6364, 2015.

[30] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Computing & Applications*, 2015.

[31] S. C. Chu and P. W. Tsai, "Computational intelligence based on the behaviour of cats," *International Journal of Innovative Computing Information and Control*, vol. 3, pp. 163–173, 2007.

[32] R. Rajabioun, "Cuckoo optimization algorithm," *Applied Soft Computing Journal*, vol. 11, no. 8, pp. 5508–5518, 2011.

[33] J. C. Bansal, H. Sharma, S. S. Jadon, and M. Clerc, "Spider Monkey Optimization algorithm for numerical optimization," *Memetic Computing*, vol. 6, no. 1, pp. 31–47, 2014.

[34] D. Dasgupta, *Artificial Immune Systems and Their Applications*, Springer, 1999.

[35] S. Das, A. Biswas, S. Dasgupta, and A. Abraham, "Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications," *Studies in Computational Intelligence*, vol. 203, pp. 23–55, 2009.

[36] A. H. Gandomi and A. H. Alavi, "Krill herd: a new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 4831–4845, 2012.

[37] V. K. Kamboj, S. K. Bath, and J. S. Dhillon, "Solution of non-convex economic load dispatch problem using Grey Wolf Optimizer," *Neural Computing and Applications*, 2015.

[38] E. Emary, H. M. Zawbaa, C. Grosan, and A. E. Hassenian, "Feature subset selection approach by gray-wolf optimization," in *Afro-European Conference for Industrial Advancement*, vol. 334 of *Advances in Intelligent Systems and Computing*, Springer, 2015.

[39] S. Gholizadeh, "Optimal design of double layer grids considering nonlinear behaviour by sequential grey wolf algorithm," *Journal of Optimization in Civil Engineering*, vol. 5, no. 4, pp. 511–523, 2015.

[40] Y. Yusof and Z. Mustaffa, "Time series forecasting of energy commodity using grey wolf optimizer," in *Proceedings of the International Multi Conference of Engineers and Computer Scientists (IMECS '15)*, vol. 1, Hong Kong, March 2015.

[41] G. M. Komaki and V. Kayvanfar, "Grey wolf optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time," *Journal of Computational Science*, vol. 8, pp. 109–120, 2015.

[42] A. A. El-Fergany and H. M. Hasanien, "Single and multi-objective optimal power flow using grey wolf optimizer and differential evolution algorithms," *Electric Power Components and Systems*, vol. 43, no. 13, pp. 1548–1559, 2015.

[43] K. Shankar and P. Eswaran, "A secure visual secret share (VSS) creation scheme in visual cryptography using elliptic curve cryptography with optimization technique," *Australian Journal of Basic & Applied Science*, vol. 9, no. 36, pp. 150–163, 2015.

[44] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, 2016.

[45] V. K. Kamboj, "A novel hybrid PSOGWOapproach for unit commitment problem," *Neural Computing and Applications*, 2015.

[46] A. Zhu, C. Xu, Z. Li, J. Wu, and Z. Liu, "Hybridizing grey Wolf optimization with differential evolution for global optimization and test scheduling for 3D stacked SoC," *Journal of Systems Engineering and Electronics*, vol. 26, no. 2, pp. 317–328, 2015.

[47] T.-S. Pan, T.-K. Dao, T.-T. Nguyen, and S.-C. Chu, "A communication strategy for paralleling grey wolf optimizer," *Advances in Intelligent Systems and Computing*, vol. 388, pp. 253–262, 2015.

[48] J. Jayapriya and M. Arock, "A parallel GWO technique for aligning multiple molecular sequences," in *Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI '15)*, pp. 210–215, IEEE, Kochi, India, August 2015.

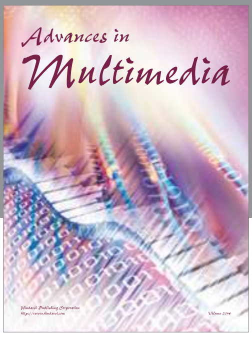
[49] M. M. Afsar and M.-H. Tayarani-N, "Clustering in sensor networks: a literature survey," *Journal of Network and Computer Applications*, vol. 46, pp. 198–226, 2014.

[50] W. B. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Siences (HICSS '00)*, p. 223, IEEE, January 2000.

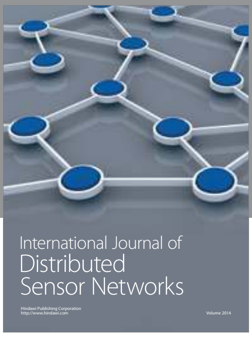
[51] N. Mittal and U. Singh, "Distance-based residual energy-efficient stable election protocol for WSNs," *Arabian Journal for Science and Engineering*, vol. 40, no. 6, pp. 1637–1646, 2015.

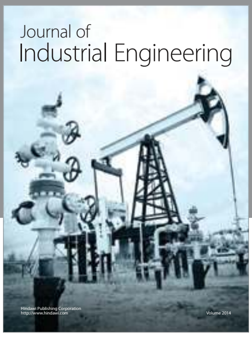
[52] E. A. Khalil and B. A. Attea, "Energy-aware evolutionary routing protocol for dynamic clustering of wireless sensor networks," *Swarm and Evolutionary Computation*, vol. 1, no. 4, pp. 195–203, 2011.

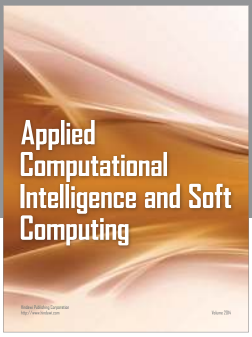
[53] B. A. Attea and E. A. Khalil, "A new evolutionary based routing protocol for clustered heterogeneous wireless sensor networks," *Applied Soft Computing*, vol. 12, no. 7, pp. 1950–1957, 2012.

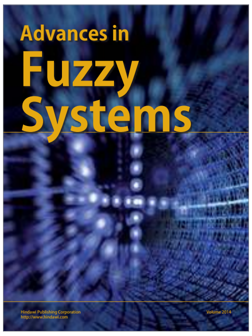

Advances in
Multimedia

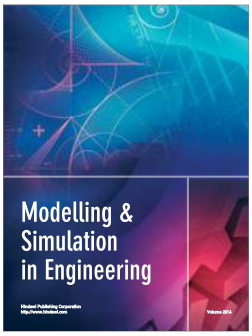
The Scientific
World Journal

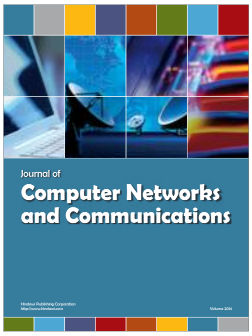

International Journal of
Distributed
Sensor Networks

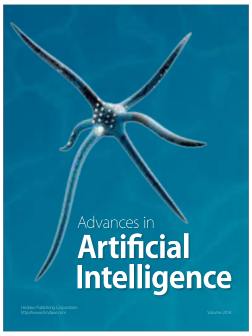

Journal of
Industrial Engineering

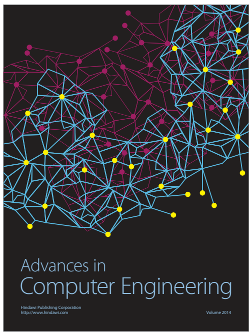

Applied
Computational
Intelligence and Soft
Computing


Advances in
Fuzzy
Systems

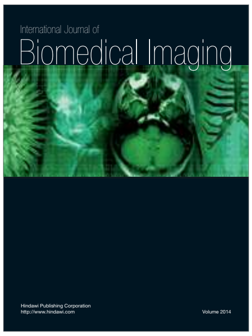

Modelling &
Simulation
in Engineering

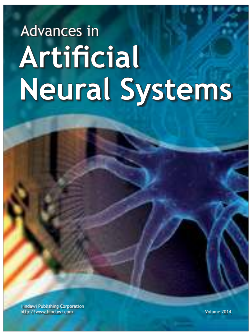

Journal of
Computer Networks
and Communications

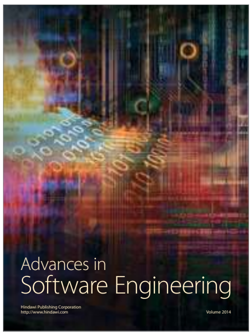

Advances in
Artificial
Intelligence


Advances in
Computer Engineering

International Journal of
Computer Games
Technology

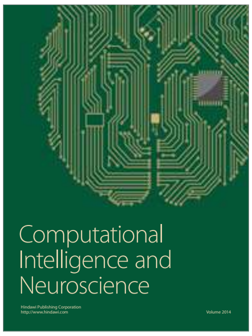

International Journal of
Biomedical Imaging


Advances in
Artificial
Neural Systems


Advances in
Software Engineering

Journal of
Robotics

Advances in
Human-Computer
Interaction


Computational
Intelligence and
Neuroscience

International Journal of
Reconfigurable
Computing

Journal of
Electrical and Computer
Engineering