algorithms

MDPI

Article

A Novel Dynamic Generalized Opposition-Based Grey Wolf Optimization Algorithm

Yanzhen Xing [1,2,3,*], Donghui Wang [1,2] and Leiou Wang [1,2]

1 Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China; wangdh@mail.ioa.ac.cn (D.W.); wangleiou@mail.ioa.ac.cn (L.W.)
2 Key Laboratory of Technology for Autonomous Underwater Vehicles, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China
3 University of Chinese Academy of Sciences, Beijing 100049, China
* Correspondence: xyz@mail.ioa.ac.cn; Tel.: +86-010-8254-7792

Received: 12 March 2018; Accepted: 9 April 2018; Published: 13 April 2018

Abstract: To enhance the convergence speed and calculation precision of the grey wolf optimization algorithm (GWO), this paper proposes a dynamic generalized opposition-based grey wolf optimization algorithm (DOGWO). A dynamic generalized opposition-based learning strategy enhances the diversity of search populations and increases the potential of finding better solutions which can accelerate the convergence speed, improve the calculation precision, and avoid local optima to some extent. Furthermore, 23 benchmark functions were employed to evaluate the DOGWO algorithm. Experimental results show that the proposed DOGWO algorithm could provide very competitive results compared with other analyzed algorithms, with a faster convergence speed, higher calculation precision, and stronger stability.

Keywords: grey wolf optimizer; generalized opposition-based learning; function optimization; heuristic algorithm; meta-heuristic

1. Introduction

Many methods have been proposed to solve varied optimization problems. Exact optimization approaches (i.e., approaches guaranteeing the convergence to an optimal solution) have been proved to be valid and useful, but many still experience considerable difficulties when dealing with complex and large-scale optimization problems. Therefore, numerous meta-heuristics have been proposed to deal with these problems. Among meta-heuristics, hybrid meta-heuristics combining exact and heuristic approaches have been successfully developed and applied to many optimization problems such as C. Blum et al., G.R. Raidl et al., C. Blum et al., F. D'Andreagiovanni et al., F. D'Andreagiovanni et al, and J.A. Egea et al. [1–6].

As a result of their superior optimization performance and simplicity, meta-heuristic optimization algorithms have recently become very popular and are currently applied in a variety of fields. Meta-heuristic algorithms predominantly benefit from stochastic operators which can avoid local optima as opposed to traditional deterministic approaches [7–9]. Among them, nature-inspired meta-heuristic optimization algorithms are widely used in this field because of their flexibility, simplicity, good performance, and robustness. Nature-inspired meta-heuristic algorithms are predominantly used to solve optimization problems by mimicking not only physical phenomena but also biological or social behaviors. They can be classified into three categories: evolution-based, physics-based, and swarm-based approaches.

Evolution-based algorithms are inspired by evolutionary processes found in nature. Genetic algorithm (GA) [10], biogeography-based Optimization (BBO) [11] and differential evolution (DE) [12] are the most popular evolution-based algorithms. In addition, a new evolution-based polar bear optimization algorithm (PBO) [13] has been proposed which imitates the survival and hunting behaviors of polar bears and presents a novel birth and death mechanism to control the population. Physics-based algorithms are inspired by natural physical phenomena. The most popular algorithms are simulated annealing (SA) [14] and gravitational search algorithm (GSA) [15]. Moreover, some new physics-based algorithms, such as the black hole algorithm (BH) [16] and ray optimization (RO) [17], have recently been proposed. Swarm-based algorithms which mimic the biological behaviors of animals are very popular. The most well-known swarm-based algorithm is the particle swarm optimization (PSO) [18] which was inspired by the social behavior of birds. Two additional classic swarm-based algorithms are the ant colony optimization (ACO) [19] and the artificial bee colony algorithm (ABC) [20] which imitate the behaviors of ant and bee colonies, respectively. Recently, many novel swarm-based algorithms imitating different population behaviors have been proposed, including the firefly algorithm (FA) [21], the bat algorithm (BA) [22], the cuckoo search (CS) [23], the social spider optimization algorithm (SSO) [24], the grey wolf optimizer (GWO) [25], the dragonfly algorithm (DA) [26], the ant lion optimizer (ALO) [27], the moth-flame optimization algorithm (MFO) [28] and the whale optimization algorithm (WOA) [29]. Swarm-based optimization algorithms have been widely used in engineering, industry and other fields as a result of their excellent optimization performance and simplicity.

Proposed by Seyedali Mirjalili in 2014, the Grey Wolf Optimizer algorithm (GWO) is a swarm-based meta-heuristic optimization algorithm and was inspired by the hunting and search behaviors of grey wolves [25]. Thanks to its easy implementation, minor parameters, and good optimization performance, the GWO has been applied to various optimization problems and many engineering projects, such as power systems, photovoltaic systems, automated offshore crane design, feature selection in neural networks, etc. [30–35]. Moreover, many researchers have attempted different strategies to enhance the performance of GWO [36–43]. In this article, we will present a novel enhanced GWO optimization algorithm known as EOGWO by using a dynamic generalized opposition-based learning strategy (DGOBL). The crux of the DGOBL strategy is to increase the diversity of a population, a significant goal for a meta-heuristic optimization algorithm. Moreover, the DGOBL strategy uses a dynamic search interval instead of a fixed search interval which can improve the likelihood of finding solutions closer to global optimum in a short time. We validate the proposed DOGWO algorithm on 23 benchmark functions. The results show that the proposed DOGWO algorithm outperforms other algorithms mentioned in this paper with fast convergence speed, high calculation precision, and strong stability.

The rest of the paper is organized as follows. In Section 2, the standard GWO algorithm is briefly introduced. The dynamic generalized opposition-based learning strategy is introduced in detail and a new dynamic generalized opposition-based Grey Wolf Optimization algorithm (DOGWO) is presented in Section 3. Several simulation experiments are conducted in Section 4 and a comparative study on DOGWO and other optimization algorithms with various benchmarks is also presented. Section 4 also describes the results and gives a detailed analysis about the results. Finally, the work is concluded in Section 5.

2. The Grey Wolf Optimizer

Proposed by S. Mirjalilli in 2014, the GWO algorithm was inspired by the grey wolf's unique hunting strategies, notably prey searching. A typical social hierarchy within grey wolf packs is shown in Figure 1.

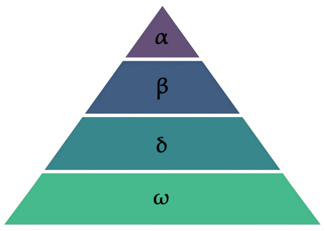


Figure 1. Social hierarchy of grey wolves.

The GWO algorithm assumes the grey wolf pack presents four levels: alpha (α) at the first level, beta (β) at the second level, delta (δ) at the third level, and omega (ω) at the last level. α wolves are leaders of wolf packs; they are responsible for making decisions that concern the whole pack. β wolves are subordinate wolves that help α in pack activities. δ wolves are designated to specific tasks such as sentinels, scouts, caretakers and so on; they submit to α and β wolves but dominate the ω wolves. ω wolves which are the lowest ranking wolves in a pack; they exist to maintain and support the dominance structure and satisfy the entire pack.

To mathematically describe the GWO algorithm, we consider α as the fittest solution and β and δ as the second and third best solutions, respectively. Other solutions represent ω wolves. In the GWO algorithm, hunting activities are guided by α, β and δ wolves; ω wolves obey wolves of the other three social levels.

According to C. Muro [44], the main phases of grey wolf hunting are:

- Tracking, chasing, and approaching the prey.
- Pursuing, encircling, and harassing the prey until it stops moving.
- Attacking towards the prey.

2.1. Encircling Prey

To mathematically model the encircling behavior of grey wolves according to one dimension, the following equations are proposed:

$$D = \left| C \cdot X_p(t) - X_w(t) \right| \tag{1}$$

$$X_w(t+1) = X_p(t) - A \cdot D \tag{2}$$

where t is the current iteration, $X_p(t)$ is the position vector of the prey, and $X_w(t)$ is the position of a grey wolf. A and C are coefficient vectors which can be calculated as follows:

$$A = 2a \times rand_1 - a \tag{3}$$

$$C = 2 \times rand_2 \tag{4}$$

where a is the convergence factor which will decrease from 2 to 0 linearly over the iteration. $rand_1$ and $rand_2$ are two random numbers in the range [0, 1] that play a significant role in the free derivation of the algorithm.

2.2. Hunting

In many situations, we have no idea about the location of the optimum (prey). In the GWO algorithm, the author presumes that α, β, and δ have better knowledge about the possible location of prey. Therefore, these three wolves are responsible for guiding other wolves to search for the prey. To mathematically simulate the hunting behavior, the distances of α, β, and δ to the prey are calculated by Equation (5); these three wolves will decide with respect to their hierarchical ranking about the potential position of prey which is described by Equation (6). Finally, other wolves will update their positions according to the command of α, β, and δ, which is shown by Equation (7).

$$D_\alpha(t) = \left| C_\alpha \cdot X_\alpha(t) - X(t) \right|, D_\beta(t) = \left| C_\beta \cdot X_\beta(t) - X(t) \right|, D_\delta(t) = \left| C_\delta \cdot X_\delta(t) - X(t) \right| \tag{5}$$

$$X_1(t) = X_\alpha(t) - A_1 \cdot D_\alpha(t), X_2(t) = X_\beta(t) - A_2 \cdot D_\beta(t), X_3(t) = X_\delta(t) - A_3 \cdot D_\delta(t) \tag{6}$$

$$X(t+1) = [X_1(t) + X_2(t) + X_3(t)]/3 \tag{7}$$

2.3. Attacking

In the GWO algorithm, the varied parameter A dominates the pack to either diverge from the prey or gather to the prey. This can be regarded as exploitation and exploration behavior in the process of searching for the optimum. It is defined as follows:

when $|A| < 1$, the wolves pack will gather to attack the prey;

when $|A| > 1$, the wolves pack will diverge and search for the new potential prey.

3. Dynamic Generalized Opposition-Based Learning Grey Wolf Optimizer (DOGWO)

To enhance the global search ability and accuracy of the GWO algorithm, the dynamic generalized opposition-based learning strategy (DGOBL) is appended to GWO. In this section, we will introduce the DGOBL strategy and present an enhanced GWO algorithm referred to as DOGWO.

3.1. Opposition-Based Learning (OBL)

Opposition-based Learning (OBL) is a computational intelligence strategy which was first proposed by Tizhoosh [45]. The OBL has proved to be an effective strategy to enhance the meta-heuristic optimization algorithms; it has been applied to many optimization algorithms such as the differential evolution algorithm, the genetic algorithm, the particle swarm optimization algorithm, the ant colony optimization algorithm, etc. [46–54]. According to OBL, the probability that the opposite individual is closer to the optimum than the current individual is 50%. Therefore, it will generate the opposite individual of the current individual, evaluate the fitness of both individuals, and select the better one as a new individual which can consequently improve the quality of the search population.

3.1.1. Opposite Number

Let $x \in [lb, ub]$ be a real number. The opposite number of x is defined by:

$$x^* = lb + ub - x \tag{8}$$

3.1.2. Opposite Point

Opposite definition in high-dimension can be described similarly as follows:

Let $X = (x_1, x_2, x_3, \ldots, x_D)$ be a point in a D-dimensional space, where $x_1, x_2, x_3, \ldots, x_D \in R$, where $x_j \in [lb_j + ub_j]$, $j \in 1, 2, \ldots, D$. The opposite point $X^* = (x_1^*, x_2^*, x_3^*, \ldots, x_D^*)$ is defined by:

$$x_j^* = lb_j + ub_j - x_j \tag{9}$$

3.2. Region Transformation Search Strategy (RTS)

3.2.1. Region Transformation Search

Let X be a search agent and $X \in P(t)$ where P(t) is the current population and t indicates the iteration. If Φ transform is applied to X, then X will become X*. The transformation can be defined by:

$$X^* = \Phi(X) \tag{10}$$

After the Φ transform, the search space of X will altered from S(t) to a new search space $S'(t)$, which can be defined as follows:

$$S'(t) = \Phi(S(t)) \tag{11}$$

3.2.2. RTS-Based Optimization

Iterative optimization process can be treated as the process of region transformation search in which the solution will transform from the current search space to a new search space after each iteration.

Let P(t) be the current search population of which the search space is S(t) and the population number is N. If Φ transform is applied to every search agent in P(t), then the transformed search agents will compose a new population $P'(t)$.

The central tenet of RTS-based optimization is that if we choose the best N search agents between P(t) and the transformed $P'(t)$ to form a new population P(t + 1) for the next generation, then we can easily know that $\forall x \in P(t+1)$ which is better than $\forall x \in \{P(t) \cup P'(t) - P(t+1)\}$ [55].

3.3. Dynamic Generalized Opposition-Based Learning Strategy (DGOBL)

3.3.1. The Concept of DGOBL

The focus of dynamic generalized opposition-based learning strategy (DGOBL) is to transform the fixed search space of individuals to a dynamic search space which can provide more chances to find solutions closer to the optimum. The DGOBL can be explained as follows:

Let x be an individual in current search space S, where $x \in [lb, ub]$. The opposite point x* in the transformed space S* can then be defined by:

$$x^* = R(lb_j + ub_j\ ) - x \tag{12}$$

where R is the transforming factor which is a random number in the range [0, 1].

According to the definition, $x \in [lb, ub]$, then $x^* \in [R(lb + ub) - lb, R(lb + ub) - ub]$. The center of search space will transform from a fixed position $\frac{lb+ub}{2}$ to a random position in range $[-\frac{lb+ub}{2}, \frac{lb+ub}{2}]$.

If the current population number is N and dimension of individual is D, for an individual $X_i = (X_{i1}, X_{i2}, \ldots, X_{iD})$, the generalized opposition-based individual can be described as $X_i^* = (X_{i1}^*, X_{i2}^*, \ldots, X_{iD}^*)$ which can be calculated by:

$$X_{i,j}^* = R[lb_j(t) + ub_j(t)] - X_{ij},\ i = 1, 2, \ldots, N; j = 1, 2, \ldots, D \tag{13}$$

where R is a random number in range [0, 1], t indicates the iteration, and $lb_j(t)$, $ub_j(t)$ are dynamic boundaries which can be obtained by the following equation:

$$lb_j(t) = \min(X_{i,j});\ ub_j(t) = \max(X_{i,j}) \tag{14}$$

It may be possible that the transformed individual $X_{i,j}^*$ jumps out of the boundary $[X_{min}, X_{max}]$. In this case, the transformed individual will be reset to be a random value in the interval as follows:

$$X_{i,j}^* = rand(lb_j(t), ub_j(t)),\ \text{if } X_{i,j}^* < X_{min} \text{ or } X_{i,j}^* > X_{max} \tag{15}$$

3.3.2. Optimization Mechanism Based on DGOBL and RTS

To both avoid the randomness of Region Tranformation Search (RTS) and to take advantage of Dynamic Generalized Opposition-based Learning (DGOBL), an effective optimization mechanism based on DGOBL and RTS is proposed.

Let $X = (x_1, x_2, x_3, \ldots, x_D)$ be a point in a D-dimensional space and F(x) is an optimization function. According to the definition above, $X^* = (x_1^*, x_2^*, x_3^*, \ldots, x_D^*)$ is the opposite point of $X = (x_1, x_2, x_3, \ldots, x_D)$. Then, the fitness of individual x_j and x_j^* is evaluated respectively and denoted as $F(x_j)$ and $F(x_j^*)$. Finally, the superior one is chosen as a new point by comparing the value of $F(x_j)$ and $F(x_j^*)$ [56,57].

The optimization mechanism based on DGOBL and RTS is shown in Figure 2. The current population P(t) has three individuals x_1, x_2, x_3 where t indicates the iteration. According to the dynamic generalized opposition-based learning strategy, three transformed individuals x_1^*, x_2^*, x_3^*

compose the opposite population OP(t). By using the optimization mechanism, three fittest individuals x_1, x_2*, x_3* are chosen as a new population P′(t).

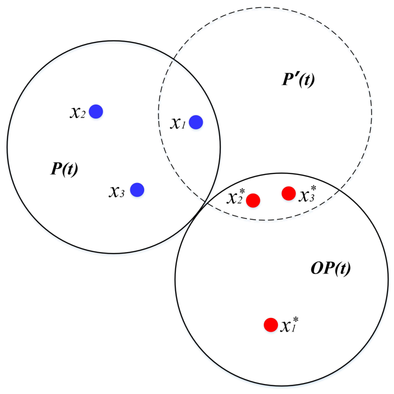


Figure 2. The optimization mechanism based on Dynamic Generalized Opposition-based Learning (DGOBL) and Region Tranformation Search (RTS).

3.4. Enhancing GWO with DGOBL Strategy (DOGWO)

Applying DGOBL to GWO can increase the number of potential points and, accordingly, expand the search area. It also will help to improve the robustness of the modified algorithm. Moreover, DGOBL can provide a better population to search for the optimum. Consequently, this can increase the convergence speed. In addition, the pseudo code of the improved DOGWO is shown as follows (Algorithm 1):

Algorithm 1: Dynamic Generalized Opposition-Based Grey Wolf Optimizer.

```
1 Initialize the original position of alpha, beta and delta
2 Randomly initialize the positions of search agents
3 set loop counter L = 0
4 While L ≤ Max_iteration do
5    Update the dynamic interval boundaries according to Equation (14)
6    Set the DGOBL jumping strategy according to Equation (15)
7    for i = 1 to Searchagent_NO do
8        for j = 1 to Dim do
9            $OP_{ij} = r*[a_j(t) + b_j(t)] - P_{ij}$
10       end
11   end
12   Calculate the fitness value of $P_{ij}$ and $OP_{ij}$
13   if fitness of $OP_{ij} < P_{ij}$
14           $P_{ij} = OP_{ij}$;
15   else
16           $P_{ij} = P_{ij}$;
17   end
18   Choose alpha, beta, delta according to the fitness value
19   $X_\alpha$ = the best search agent
20   $X_\beta$ = the second best search agent
21   $X_\delta$ = the third best search agent
22   for each search agent do
23       Update the position of current search according to Equation (7)
24   end
25   Calculate the fitness value of all search agents
26   Update $X_\alpha$, $X_\beta$, and $X_\delta$
27   L = L + 1;
28 end
29 return $X_\alpha$
```

4. Experiments and Discussion

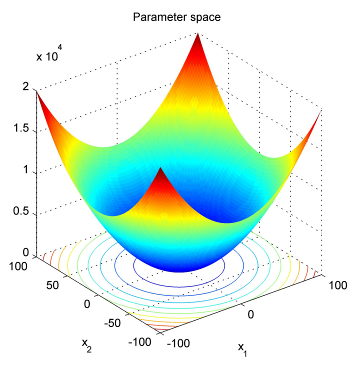
4.1. Benchmark Functions

In this section, 23 benchmark functions commonly used in research were applied to evaluate the optimal performance of the proposed DOGWO algorithm [25–29,37,42,43]. The benchmark functions used were all minimization functions which can be divided into three categories: unimodal, multimodal, and fixed-dimension multimodal. According to S. Mirjalili et al. [25], the unimodal functions are suitable for benchmarking optimum exploitation ability of an algorithm; similarly, the multimodal functions are suitable for benchmarking the ability of optimum exploration of an algorithm. These benchmark functions are listed in Table 1 where F1–F6 are unimodal functions, F7–F12 are multimodal functions, and F13–F23 are fixed-dimension multimodal benchmark functions. Moreover, the 2-D versions of these 23 benchmark functions are presented in Figure 3 for a better analysis of the form and search space.

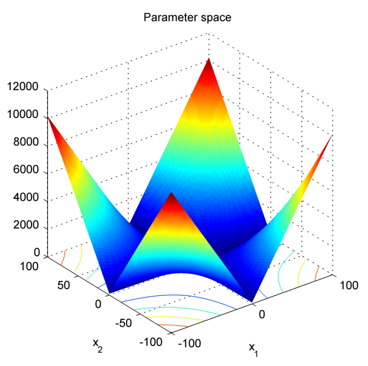
Table 1. Benchmark functions.

Function	Dim [1]	Range [2]	f_{min} [3]
$F_1(x) = \sum_{i=1}^{n} x_i^2$	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
$F_5(x) = \sum_{i=1}^{n} \mathrm{i}x_\mathrm{i}^4 + \mathrm{random}[0,1)$	30	$[-1.28, 1.28]$	0
$F_6(x) = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$	30	$[-100, 100]$	0
$F_7(x) = \sum_{i=1}^{n} \lvert x_i \sin(x_i) + 0.1x_i \rvert$	30	$[-30, 30]$	0
$F_8(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]$	0
$F_9(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	30	$[-32, 32]$	0
$F_{10}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_1^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]$	0
$F_{11}(x) = \frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m; x_i > a \\ 0; -a < x_i < a \\ k(-x_i - a)^m; x_i < a \end{cases}$	30	$[-50, 50]$	0
$F_{12}(x) = 0.1\{sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + 10sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	$[-50, 50]$	0
$F_{13}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})}\right)^{-1}$	2	$[-65, 65]$	1
$F_{14}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	$[-5, 5]$	0.00030
$F_{15}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	-1.0316
$F_{16}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cosx_1 + 10$	2	$[-5, 5]$	0.398
$F_{17}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_1^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_1^2))]$	2	$[-2, 2]$	3
$F_{18}(x) = -\cos(x_1)\cos(x_2) \times \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$	2	$[-100, 100]$	-1
$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right)$	3	$[1, 3]$	-3.86
$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right)$	6	$[0, 1]$	-3.32
$F_{21}(x) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	$[0, 10]$	-10.1532
$F_{22}(x) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	$[0, 10]$	-10.4028
$F_{23}(x) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	$[0, 10]$	-10.5363

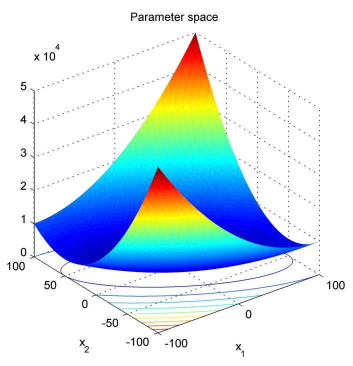
[1] Dim: dimension of the function. [2] Range: boundary of the function's search space. [3] f_{min}: the optimum of the function.



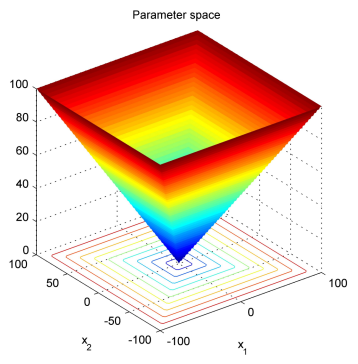
(F1)



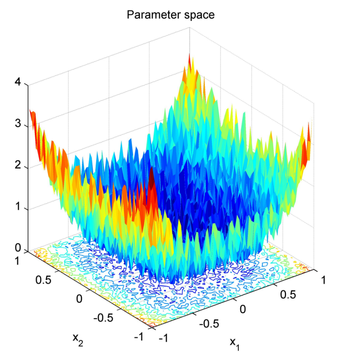
(F2)



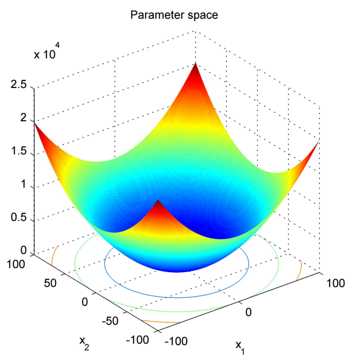
(F3)



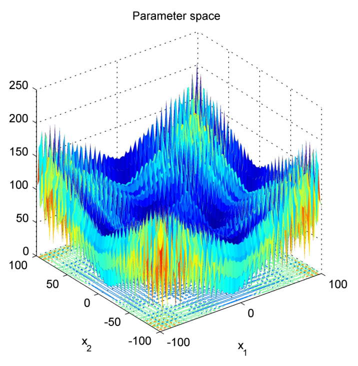
(F4)



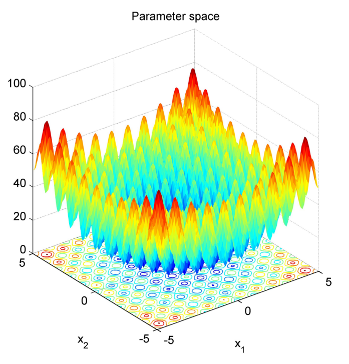
(F5)



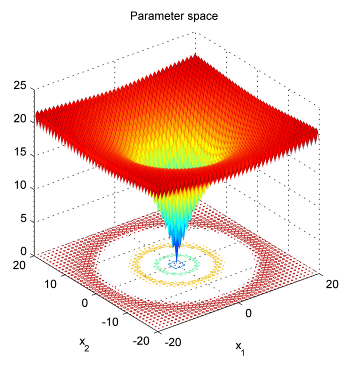
(F6)



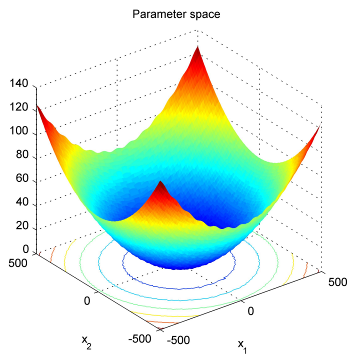
(F7)



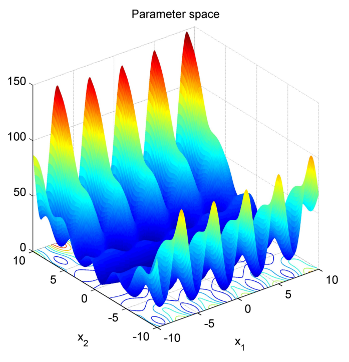
(F8)



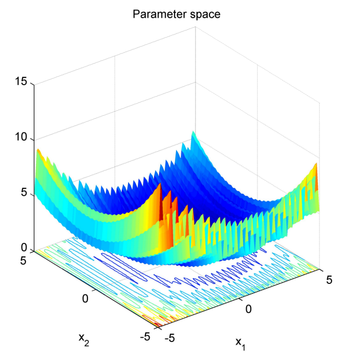
(F9)



(F10)

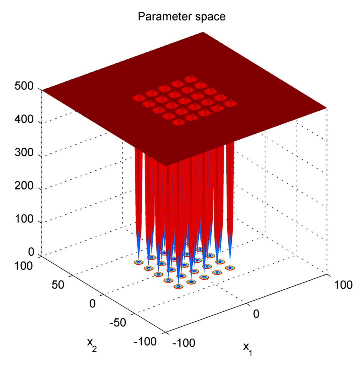


(F11)

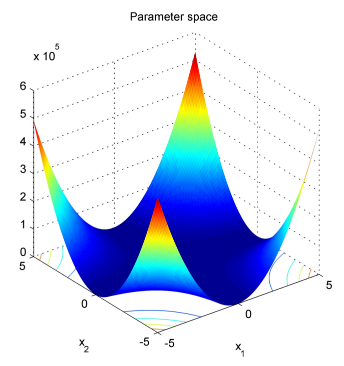


(F12)

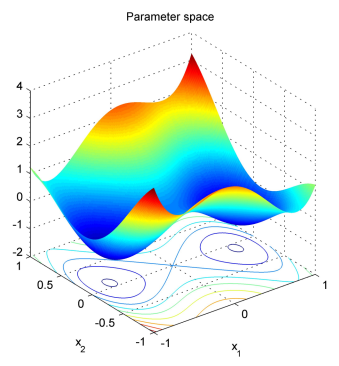
Figure 3. *Cont.*



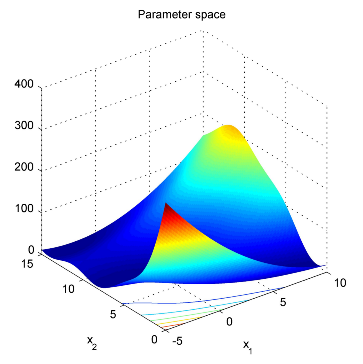
(F13)



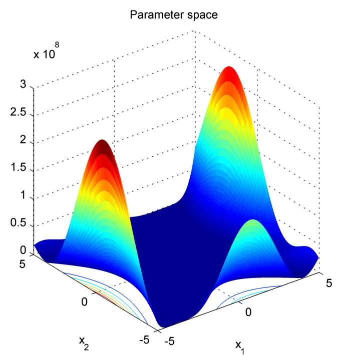
(F14)



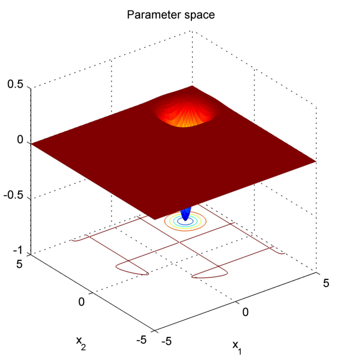
(F15)



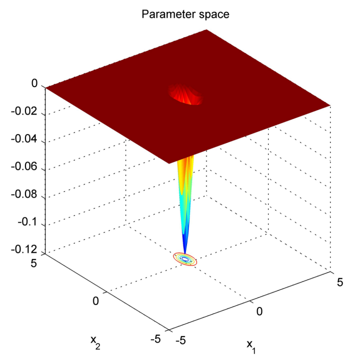
(F16)



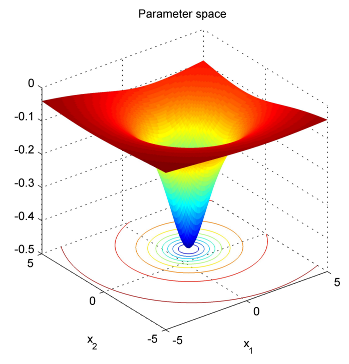
(F17)



(F18)



(F19, F20)



(F21, F22, F23)

Figure 3. 2-D versions of the 23 benchmark functions.

4.2. Simulation Experiments

To verify the optimization performance of the proposed DOGWO algorithm, it was compared with several famous and recent algorithms: BA, ABC, PSO, MFO, ALO, and GWO by using the average and standard deviation. All the algorithms were run 30 times independently on each of the benchmark functions with the population size set as 50 and the iteration number as 1000. The parameters settings of the aforementioned algorithms are given in Table 2. Experiments were conducted in MATLAB R2012b of which the runtime environment was Intel(R) Corel(TM) i7-3770 CPU, 3.5 GB memory.

The experiment results obtained by seven algorithms are shown in Table 3 where "Function" represents test function; "Best", "Worst", "Mean", and "Std." represent the best, worst, average global optimum, and standard deviation of 30 experiments, respectively. For F1–F12 of which the optimal solutions were zero, bold values indicate that the performance of DOGWO was better than other algorithms. For fixed-dimension functions in which the optimal solutions were fixed values, bold values indicate that the DOGWO was able to find the optimal solution.

The variation of fitness function value with the increase of iteration is denoted by convergence curve. Convergence curves of the partial benchmark functions are presented in Figures 4–18. The ANOVA(Analysis of Variance) test, which was developed by R. Fisher, is a useful statistic test for comparing three or more samples for statistical significance. Figures 19–32 show the ANOVA test of global optimum for several test functions.

Moreover, Figure 33 depicts the search history of 10 search agents with 500 iterations for several benchmark functions.

Table 2. The parameters setting for seven algorithms.

Algorithm	Parameter Values
BA	A = 0.25, r = 0.5, f ∈ [0, 2], the population size N = 50
ABC	Limit = 50, the population size N = 50
PSO	V_{max} = 6, ω_{max} = 0.9, ω_{min} = 0.2, c1 = c2 = 2, the population size N = 50
MFO	a ∈ [−2, −1], the population size N = 50
ALO	w∈ [2, 6] ,the population size N = 50
GWO	a ∈ [0, 2], r1, r2 ∈ rand(), the population size N = 50
DOGWO	a ∈ [0, 2], r1, r2 ∈ rand(), R ∈ rand(), the population size N = 50

Table 3. Experiment results of benchmark functions.

Function	Algorithm	Best	Worst	Mean	Std.
	BA	3.27×10^{3}	1.59×10^{4}	7.78×10^{3}	2.97×10^{3}
	ABC	0.01	0.17	0.04	0.03
	PSO	0.15	5.22	1.97	1.45
F1	MFO	2.72×10^{-6}	2.00×10^{4}	3.00×10^{3}	5.35×10^{3}
	ALO	1.27×10^{-7}	3.94×10^{-6}	8.42×10^{-7}	9.79×10^{-7}
	GWO	7.65×10^{-73}	1.49×10^{-69}	2.24×10^{-70}	3.80×10^{-70}
	DOGWO	**0**	**0**	**0**	**0**
	BA	2.92	1.61×10^{4}	9.54×10^{2}	3.33×10^{3}
	ABC	0.01	74.34	7.86	19.78
	PSO	0.34	2.76	1.20	0.56
F2	MFO	1.12×10^{-4}	60.00	32.33	16.33
	ALO	0.18	124.28	31.01	
	GWO	5.76×10^{-42}	5.76×10^{-42}	6.09×10^{-41}	6.41×10^{-41}
	DOGWO	**0**	**0**	**0**	**0**
	BA	4.86×10^{4}	6.43×10^{4}	2.88×10^{4}	1.56×10^{4}
	ABC	3.09×10^{4}	7.98×10^{4}	5.86×10^{4}	1.21×10^{4}
	PSO	2.33×10^{3}	1.59×10^{4}	7.57×10^{3}	2.74×10^{3}
F3	MFO	2.91×10^{2}	3.84×10^{4}	1.61×10^{4}	1.12×10^{4}
	ALO	75.88	6.38×10^{2}	2.87×10^{2}	1.47×10^{2}
	GWO	5.45×10^{-25}	4.12×10^{-19}	4.04×10^{-20}	9.69×10^{-20}
	DOGWO	**0**	**0**	**0**	**0**
	BA	26.23	56.24	37.58	7.30
	ABC	47.97	61.23	54.43	3.28
	PSO	7.59	31.43	20.47	5.08
F4	MFO	29.50	69.19	55.17	9.80
	ALO	3.25	15.54	8.28	2.46
	GWO	1.26×10^{-18}	6.60×10^{-17}	1.19×10^{-17}	1.14×10^{-17}
	DOGWO	**0**	**0**	**0**	**0**

Table 3. *Cont.*

Function	Algorithm	Best	Worst	Mean	Std.
F5	BA	0.53	2.84	1.54	0.57
	ABC	0.09	0.25	0.17	0.05
	PSO	7.87	1.38×10^{2}	73.13	42.16
	MFO	0.02	18.86	2.12	4.09
	ALO	0.02	0.10	0.06	0.02
	GWO	1.46×10^{-4}	1.09×10^{-2}	4.58×10^{-4}	2.45×10^{-4}
	DOGWO	$\mathbf{2.07 \times 10^{-7}}$	$\mathbf{5.73 \times 10^{-5}}$	$\mathbf{2.15 \times 10^{-5}}$	$\mathbf{1.67 \times 10^{-5}}$
F6	BA	4.22×10^{3}	1.29×10^{4}	7.84×10^{3}	2.59×10^{3}
	ABC	0.01	0.11	0.04	0.02
	PSO	4.64	21.05	8.18	3.17
	MFO	5.67×10^{-6}	1.01×10^{4}	9.97×10^{2}	3.04×10^{3}
	ALO	1.12×10^{-7}	1.93×10^{-6}	5.60×10^{-7}	5.30×10^{-7}
	GWO	7.48×10^{-6}	0.99	0.38	0.27
	DOGWO	3.92×10^{-6}	0.50	0.27	0.18
F7	BA	1.12	14.02	6.44	3.12
	ABC	16.59	30.05	24.25	2.68
	PSO	10.45	45.83	29.46	9.08
	MFO	7.01×10^{-6}	15.32	4.39	5.56
	ALO	0.93	13.53	4.68	3.06
	GWO	4.65×10^{-42}	4.87×10^{-4}	4.01×10^{-5}	1.21×10^{-4}
	DOGWO	**0**	**0**	**0**	**0**
F8	BA	1.32×10^{-9}	4.97	2.08	1.51
	ABC	2.05×10^{-6}	1.99×10^{-4}	5.86×10^{-5}	6.26×10^{-5}
	PSO	0	0	0	0
	MFO	0	0.99	0.07	0.25
	ALO	1.07×10^{-14}	0.99	0.03	0.18
	GWO	0	0	0	0
	DOGWO	**0**	**0**	**0**	**0**
F9	BA	11.61	16.68	14.48	1.35
	ABC	0.47	2.60	1.51	0.49
	PSO	0.44	3.23	1.27	0.71
	MFO	8.69×10^{-4}	19.96	14.82	8.42
	ALO	2.22×10^{-4}	3.09	1.94	0.73
	GWO	7.99×10^{-15}	1.51×10^{-14}	1.37×10^{-14}	2.21×10^{-15}
	DOGWO	$\mathbf{8.88 \times 10^{-16}}$	$\mathbf{8.88 \times 10^{-16}}$	$\mathbf{8.88 \times 10^{-16}}$	**0**
F10	BA	62.78	1.50×10^{2}	1.03×10^{2}	21.85
	ABC	0.13	0.76	0.41	0.16
	PSO	0.31	1.01	0.67	0.19
	MFO	2.92×10^{-5}	90.51	6.05	22.96
	ALO	3.95×10^{-5}	0.08	0.01	0.02
	GWO	0	1.62×10^{-2}	1.01×10^{-3}	4.01×10^{-3}
	DOGWO	**0**	**0**	**0**	**0**
F11	BA	10.35	7.02×10^{5}	9.05×10^{4}	1.89×10^{5}
	ABC	9.61×10^{2}	7.89×10^{5}	9.62×10^{4}	1.52×10^{5}
	PSO	0.69	6.05×10^{4}	2.39×10^{3}	1.10×10^{4}
	MFO	2.66×10^{-6}	2.48	0.28	0.52
	ALO	4.35	15.35	8.08	2.76
	GWO	6.21×10^{-3}	6.01×10^{-2}	3.13×10^{-2}	1.12×10^{-2}
	DOGWO	$\mathbf{2.54 \times 10^{-6}}$	$\mathbf{5.91 \times 10^{-2}}$	$\mathbf{2.10 \times 10^{-2}}$	$\mathbf{1.01 \times 10^{-2}}$
F12	BA	4.93×10^{3}	2.92×10^{7}	2.59×10^{6}	1.30×10^{5}
	ABC	2.05×10^{3}	5.84×10^{5}	1.29×10^{5}	5.39×10^{6}
	PSO	0.26	1.16×10^{6}	4.97×10^{4}	2.18×10^{5}
	MFO	2.65×10^{-5}	3.61	0.36	0.97
	ALO	2.36×10^{-5}	9.82×10^{-2}	1.87×10^{-2}	0.19
	GWO	9.86×10^{-2}	0.85	0.34	0.18
	DOGWO	$\mathbf{1.35 \times 10^{-5}}$	0.50	0.23	0.12

Table 3. *Cont.*

Function	Algorithm	Best	Worst	Mean	Std.
F13	BA	1.992	22.90	11.13	6.28
	ABC	0.998	0.998	0.998	5.14×10^{-6}
	PSO	0.998	3.968	1.92	1.10
	MFO	0.998	5.93	1.59	1.18
	ALO	0.998	1.99	1.16	0.38
	GWO	0.998	12.67	3.73	4.33
	DOGWO	**0.998**	2.98	1.19	0.60
F14	BA	3.07×10^{-4}	0.10	1.37×10^{-2}	1.91×10^{-2}
	ABC	9.39×10^{-4}	1.20×10^{-3}	1.10×10^{-3}	6.98×10^{-5}
	PSO	8.69×10^{-4}	1.90×10^{-3}	1.00×10^{-3}	1.70×10^{-4}
	MFO	3.09×10^{-4}	1.66×10^{-3}	9.66×10^{-4}	4.04×10^{-4}
	ALO	3.08×10^{-4}	2.04×10^{-2}	3.33×10^{-3}	6.78×10^{-3}
	GWO	3.07×10^{-4}	2.04×10^{-2}	2.30×10^{-3}	6.10×10^{-3}
	DOGWO	$\mathbf{3.07 \times 10^{-4}}$	$\mathbf{3.07 \times 10^{-4}}$	$\mathbf{3.07 \times 10^{-4}}$	$\mathbf{7.54 \times 10^{-9}}$
F15	BA	−1.0316	−1.0316	−1.0316	1.44×10^{-5}
	ABC	−1.0316	−1.0316	−1.0316	1.25×10^{-6}
	PSO	−1.0316	−1.0315	−1.0316	3.37×10^{-5}
	MFO	−1.0316	−1.0316	−1.0316	6.78×10^{-16}
	ALO	−1.0316	−1.0316	−1.0316	5.19×10^{-14}
	GWO	−1.0316	−1.0316	−1.0316	7.42×10^{-6}
	DOGWO	**−1.0316**	**−1.0316**	**−1.0316**	3.34×10^{-9}
F16	BA	0.3979	0.3979	0.3979	3.64×10^{-5}
	ABC	0.3979	0.3979	0.3979	8.37×10^{-8}
	PSO	0.3979	0.4136	0.3996	2.90×10^{-3}
	MFO	0.3979	0.3979	0.3979	0
	ALO	0.3979	0.3979	0.3979	3.05×10^{-14}
	GWO	0.3979	0.3979	0.3979	3.65×10^{-7}
	DOGWO	**0.3979**	**0.3979**	**0.3979**	4.36×10^{-8}
F17	BA	3	3	3	1.18×10^{-8}
	ABC	3	3	3	8.24×10^{-11}
	PSO	3	3.0003	3	7.22×10^{-5}
	MFO	3	3	3	2.68×10^{-13}
	ALO	3	3	3	1.41×10^{-15}
	GWO	3	3	3	3.61×10^{-6}
	DOGWO	**3**	**3**	**3**	1.41×10^{-7}
F18	BA	−1	0	−0.2333	0.4302
	ABC	−1	−1	−1	1.21×10^{-10}
	PSO	−1	−0.9982	−0.9995	4.71×10^{-4}
	MFO	−1	−1	−1	0
	ALO	−1	−1	−1	7.11×10^{-12}
	GWO	−1	−1	−1	1.23×10^{-7}
	DOGWO	**−1**	**−1**	**−1**	1.49×10^{-7}
F19	BA	−3.86	−3.86	−3.86	1.65×10^{-8}
	ABC	−3.86	−3.86	−3.86	1.33×10^{-15}
	PSO	−3.86	−3.82	−3.85	9.70×10^{-3}
	MFO	−3.86	−3.86	−3.86	2.71×10^{-15}
	ALO	−3.86	−3.86	−3.86	1.08×10^{-14}
	GWO	−3.86	−3.86	−3.86	2.75×10^{-3}
	DOGWO	**−3.86**	**−3.86**	**−3.86**	2.97×10^{-3}

Table 3. *Cont.*

Function	Algorithm	Best	Worst	Mean	Std.
F20	BA	−3.32	−3.20	−3.27	5.92×10^{-2}
	ABC	−3.32	−3.32	−3.32	8.59×10^{-7}
	PSO	−3.18	−3.19	−2.99	0.14
	MFO	−3.32	−3.20	−3.27	5.40×10^{-2}
	ALO	−3.32	−3.14	−3.23	6.03×10^{-2}
	GWO	−3.32	−3.09	−3.24	8.09×10^{-2}
	DOGWO	**−3.32**	**−3.21**	**−3.31**	4.58×10^{-2}
F21	BA	−10.1532	−10.1532	−10.1532	3.006
	ABC	−10.1532	−2.6305	−5.1363	4.52×10^{-13}
	PSO	−10.1532	−2.3215	−4.7358	1.6195
	MFO	−10.1532	−2.6305	−7.9587	2.7942
	ALO	−10.1532	−2.6305	−6.5294	2.9342
	GWO	−10.1531	−2.6828	−9.3972	1.9975
	DOGWO	**−10.1532**	**−10.1532**	**−10.1532**	5.81×10^{-7}
F22	BA	−10.4029	−1.8376	−5.7487	3.21
	ABC	−10.4029	−10.4029	−10.4029	6.35×10^{-14}
	PSO	−9.0894	−2.1988	−5.0820	1.57
	MFO	−10.4029	−2.7519	−7.3236	3.42
	ALO	−10.4029	−3.7243	−8.6734	2.71
	GWO	−10.4029	−5.0877	−10.0482	1.35
	DOGWO	**−10.4029**	**−10.4029**	**−10.4029**	1.74×10^{-6}
F23	BA	−10.5364	−1.6766	−5.8869	3.66
	ABC	−10.5364	−10.5364	−10.5364	2.41×10^{-13}
	PSO	−9.5666	−2.3740	−5.4016	1.89
	MFO	−10.5364	−2.4273	−9.3062	2.81
	ALO	−10.5364	−2.4273	−8.2891	3.29
	GWO	−10.5364	−10.5356	−10.5361	1.93×10^{-4}
	DOGWO	**−10.5364**	**−10.5364**	**−10.5364**	8.12×10^{-7}

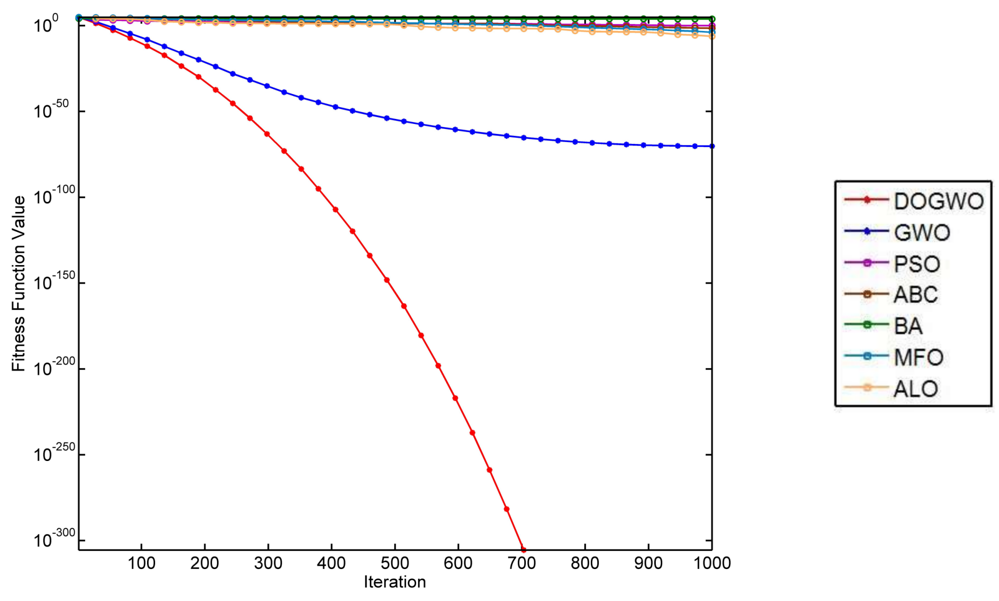


Figure 4. Convergence curves for F1.

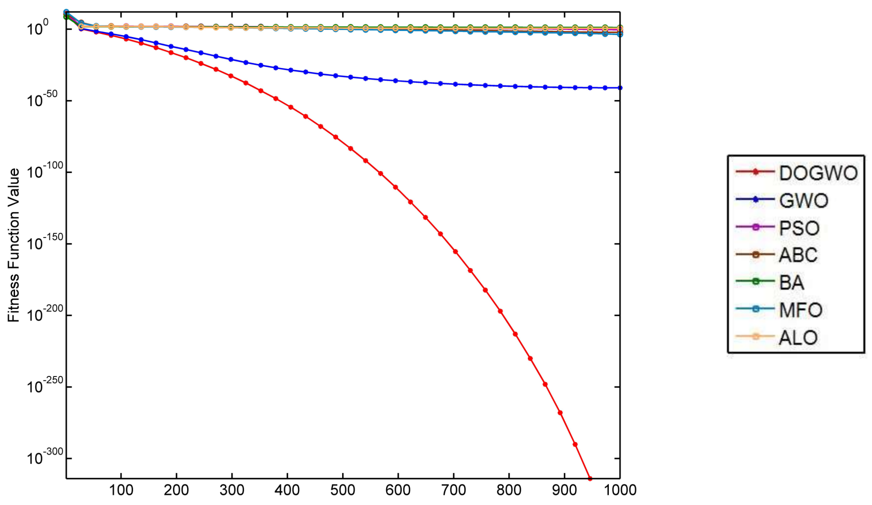


Figure 5. Convergence curves for F2.

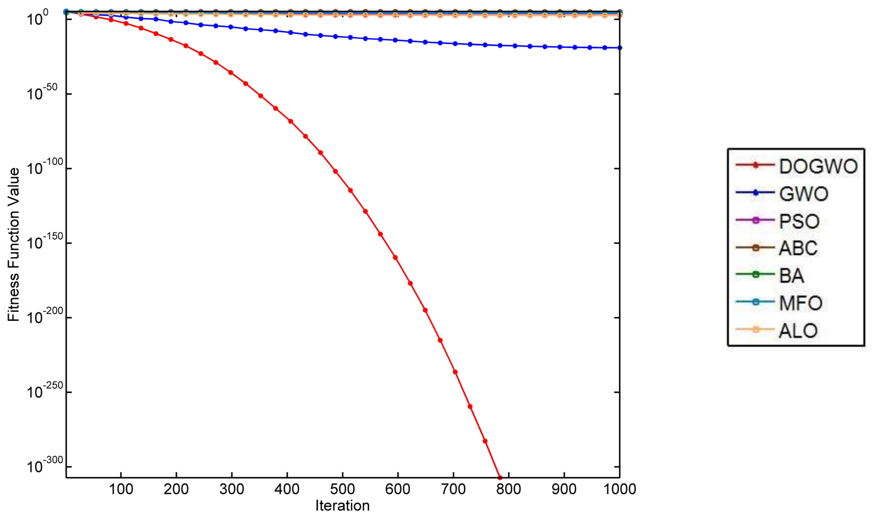


Figure 6. Convergence curves for F3.

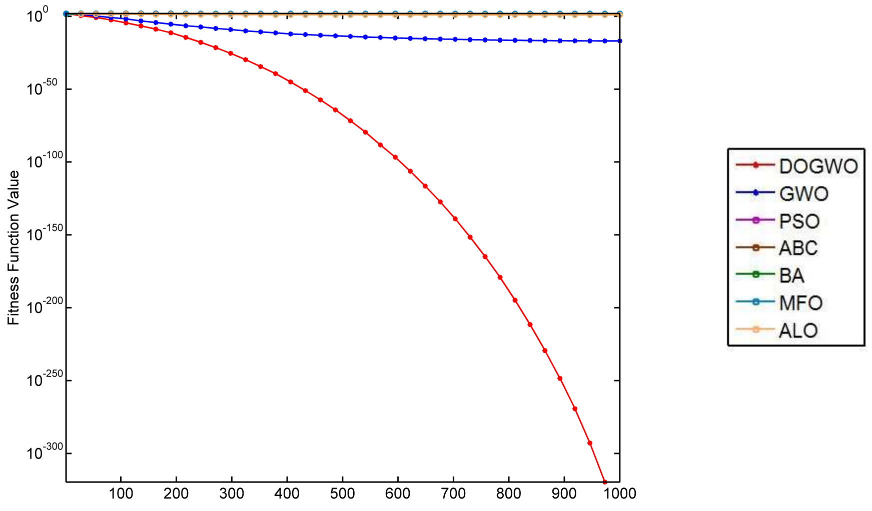


Figure 7. Convergence curves for F4.

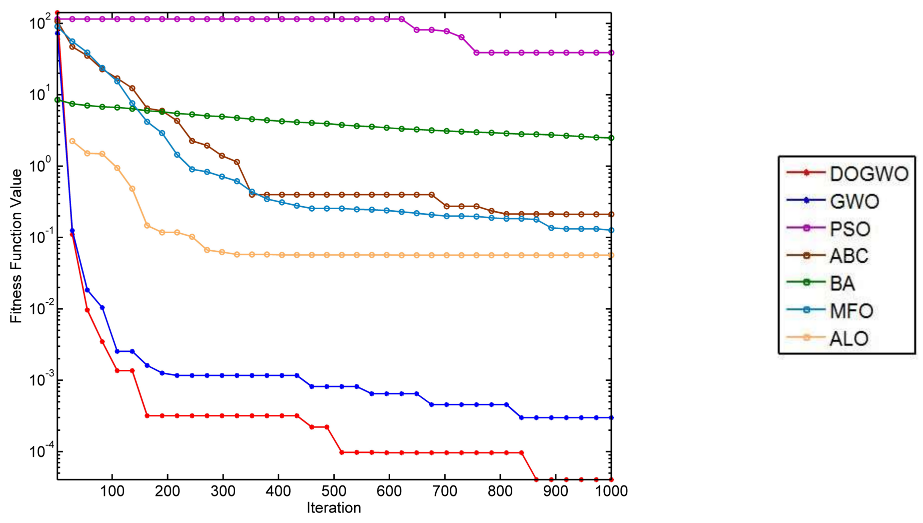


Figure 8. Convergence curves for F5.

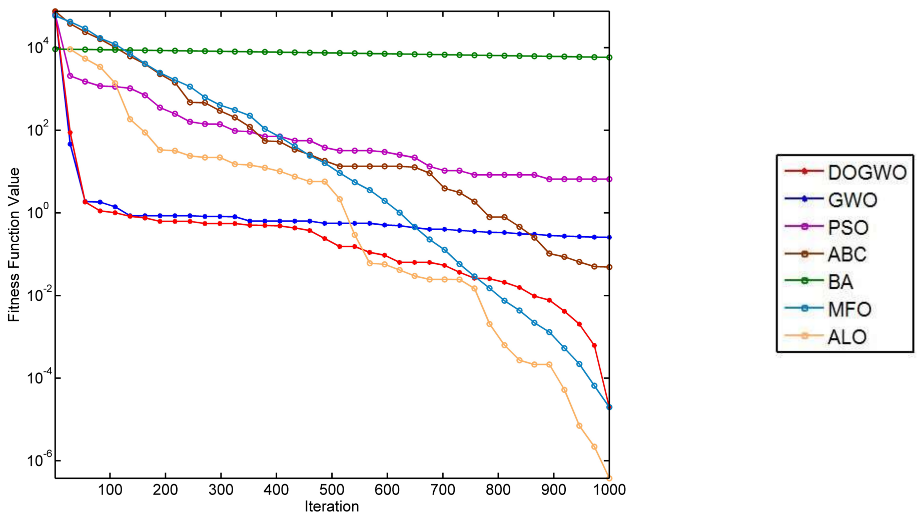


Figure 9. Convergence curves for F6.

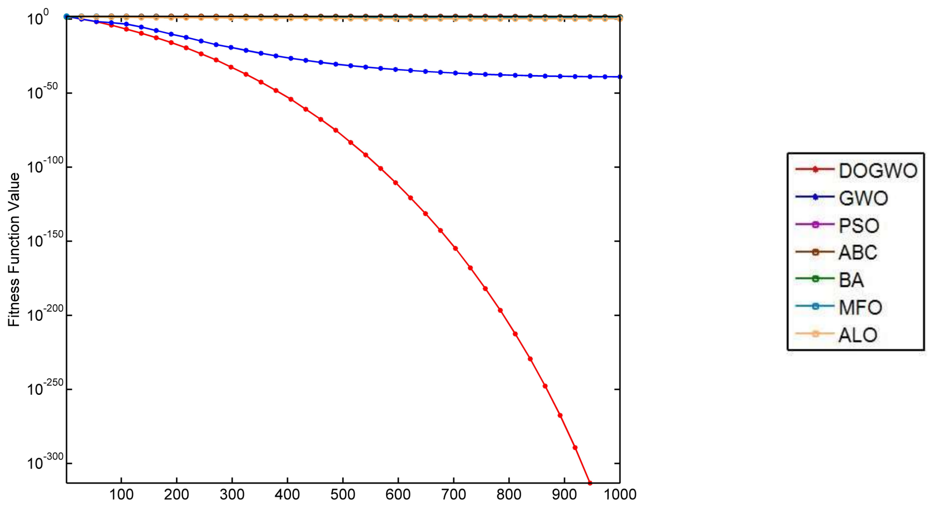


Figure 10. Convergence curves for F7.

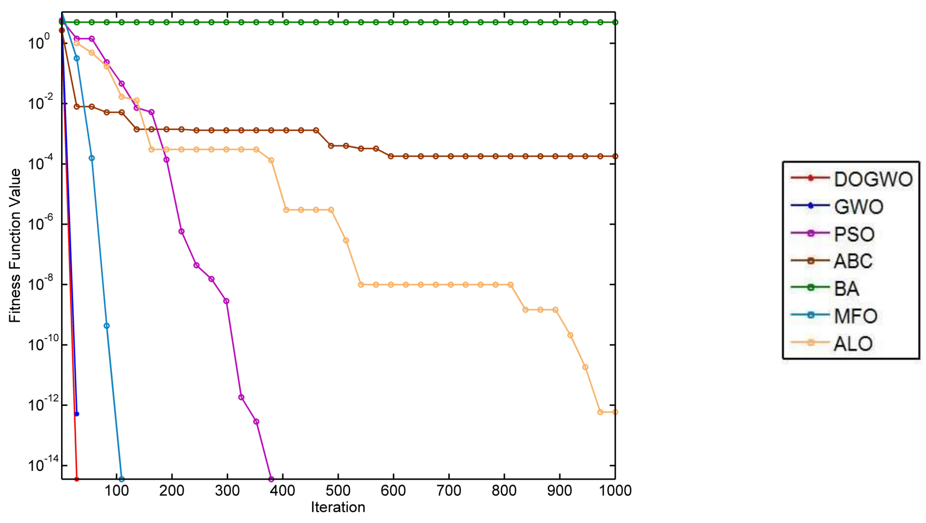


Figure 11. Convergence curves for F8.

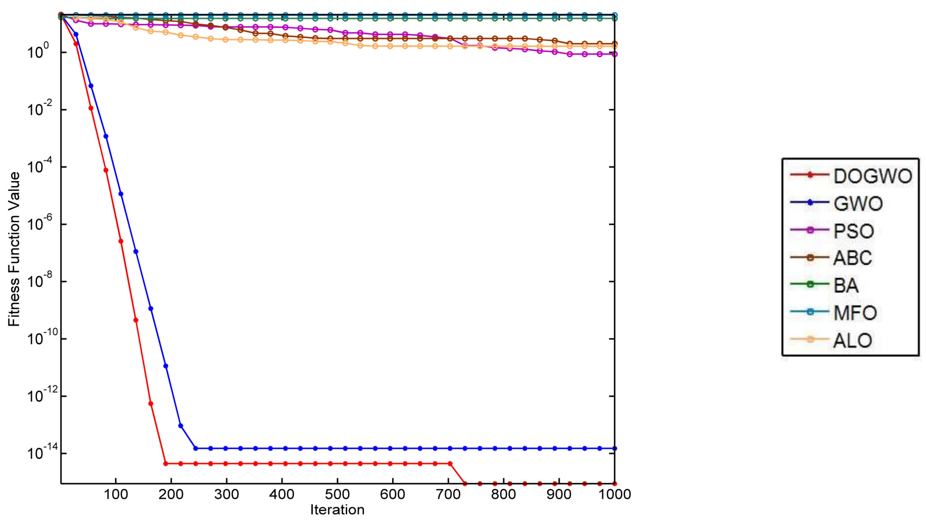


Figure 12. Convergence curves for F9.

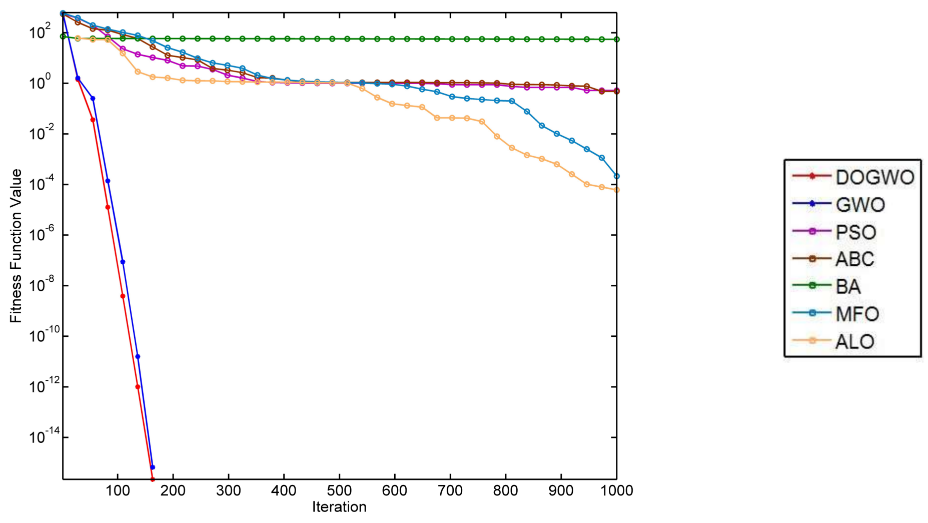


Figure 13. Convergence curves for F10.

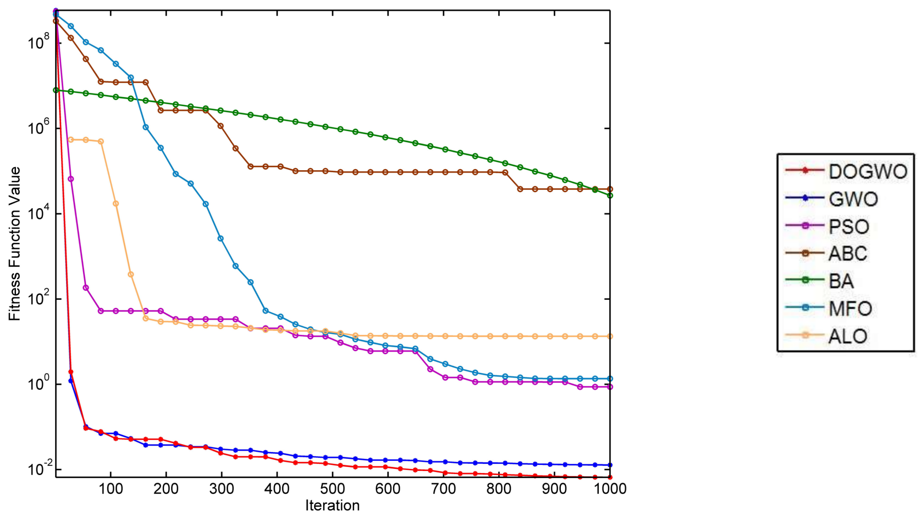


Figure 14. Convergence curves for F11.

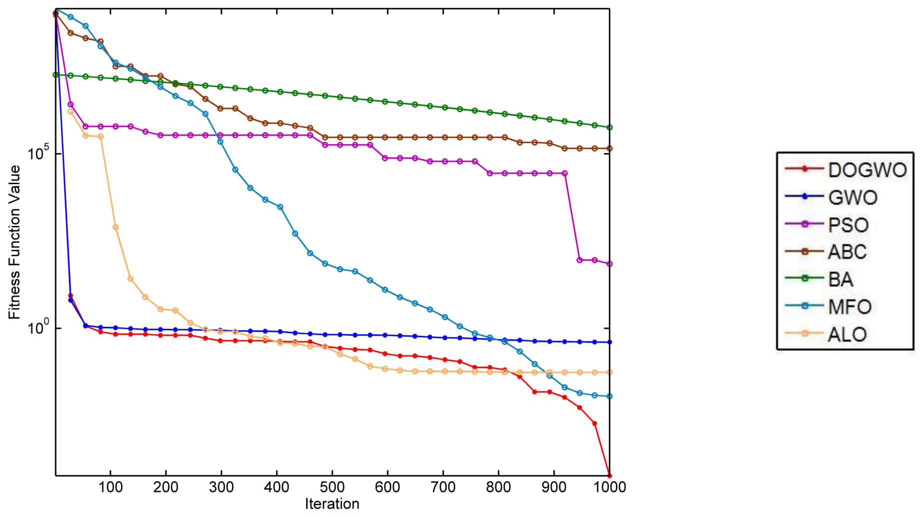


Figure 15. Convergence curves for F12.

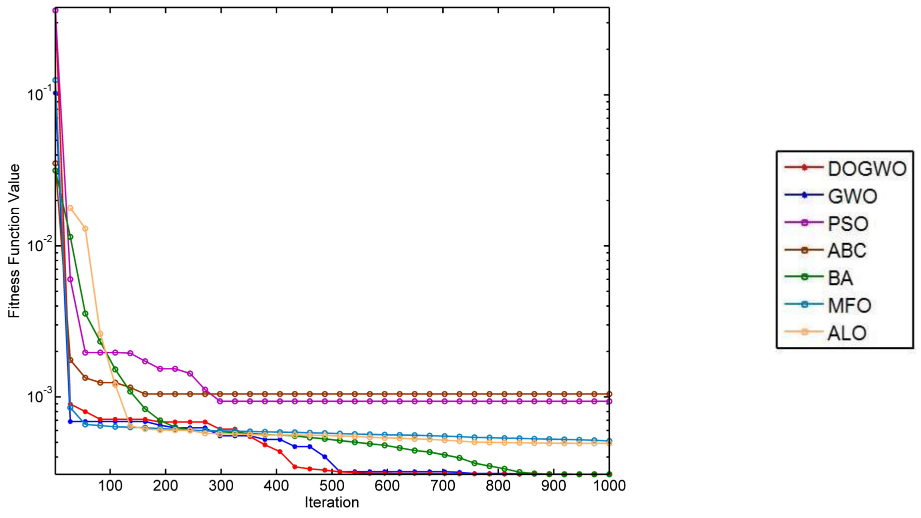


Figure 16. Convergence curves for F14.

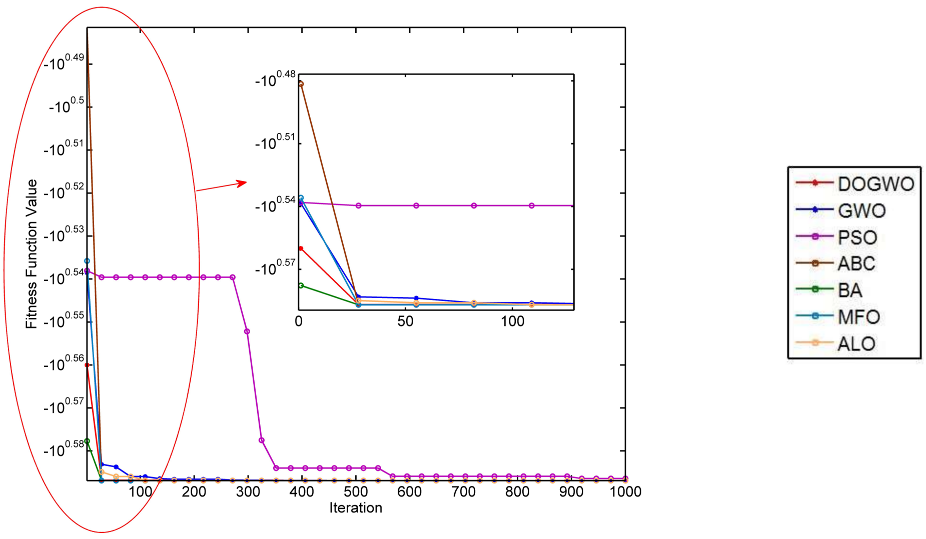


Figure 17. Convergence curves for F19–F20.

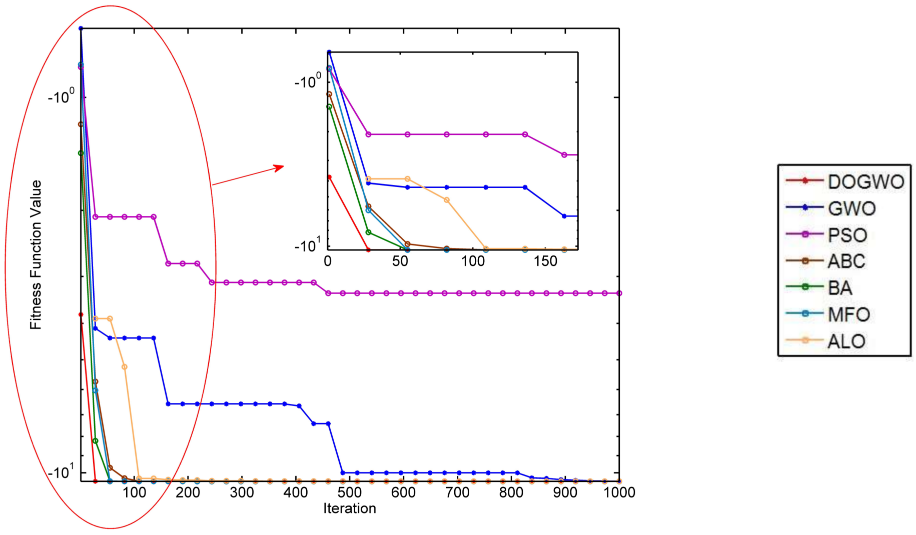


Figure 18. Convergence curves for F21–F23.

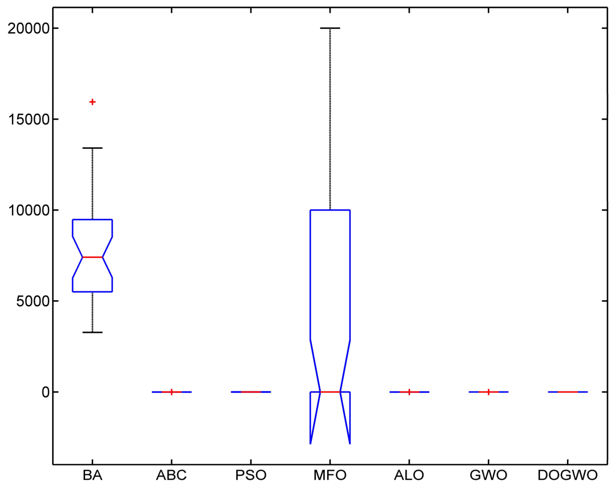


Figure 19. ANOVA test of global minimum for F1.

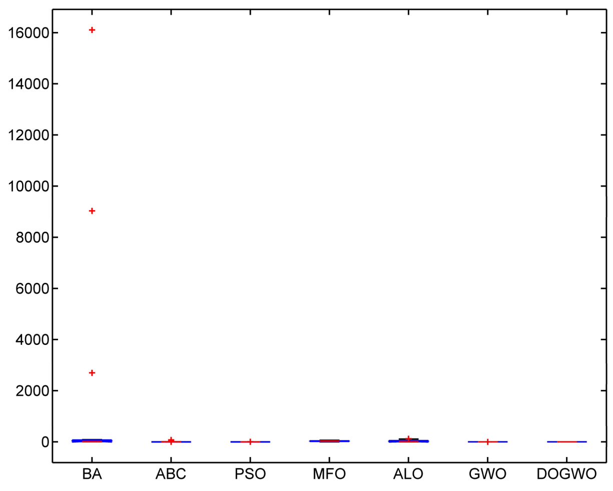


Figure 20. ANOVA test of global minimum for F2.

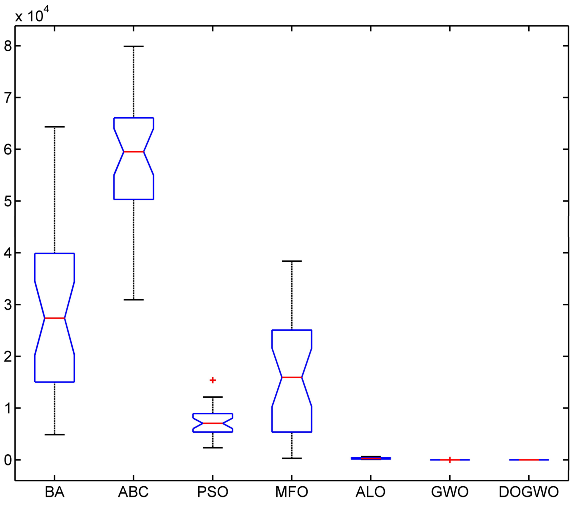


Figure 21. ANOVA test of global minimum for F3.

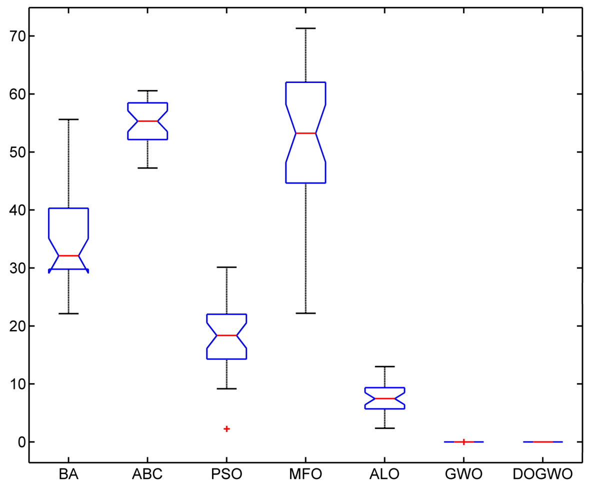


Figure 22. ANOVA test of global minimum for F4.

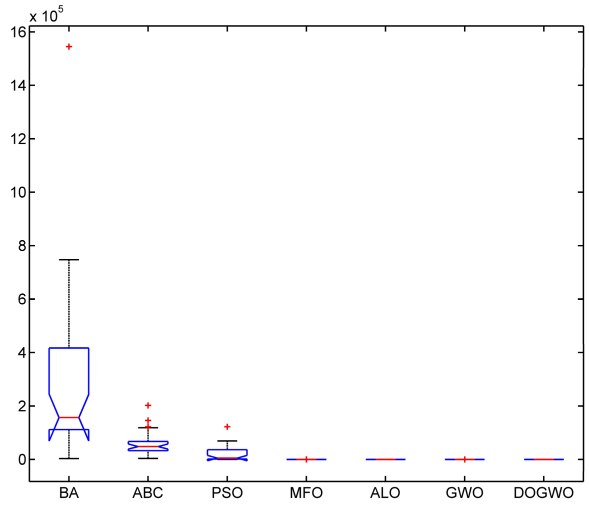


Figure 23. ANOVA test of global minimum for F5.

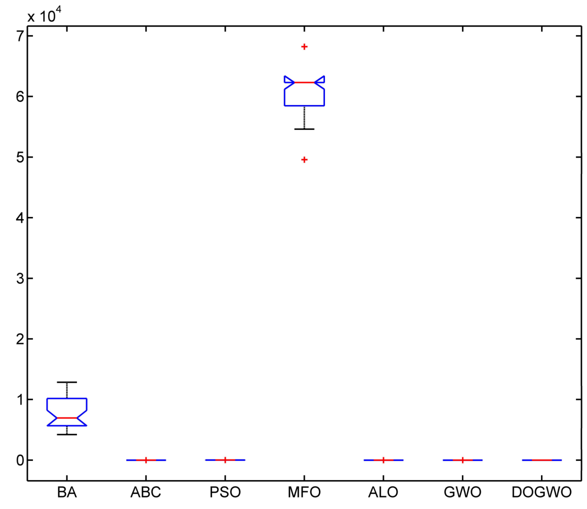


Figure 24. ANOVA test of global minimum for F6.

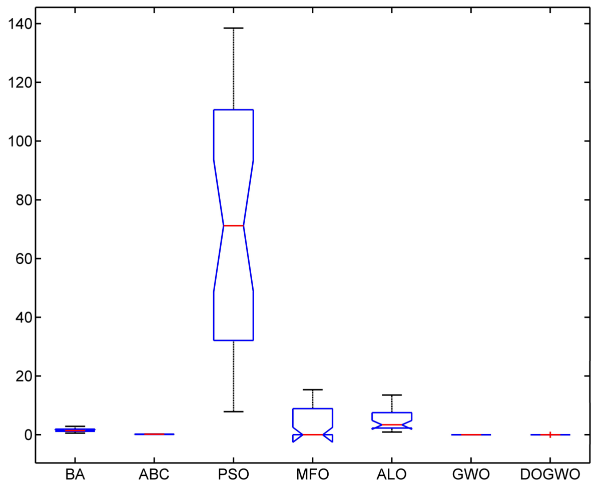


Figure 25. ANOVA test of global minimum for F7.

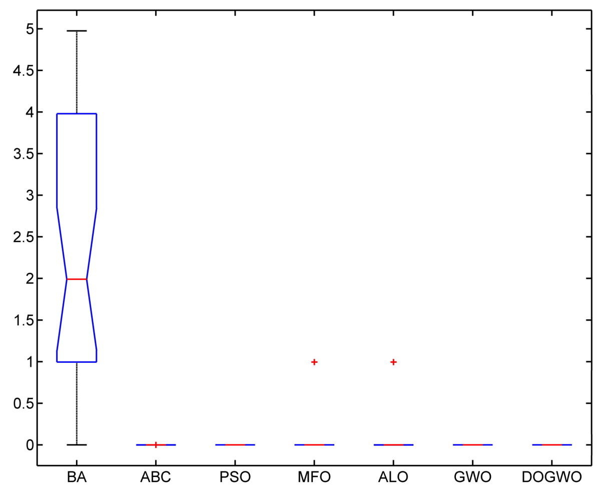


Figure 26. ANOVA test of global minimum for F8.

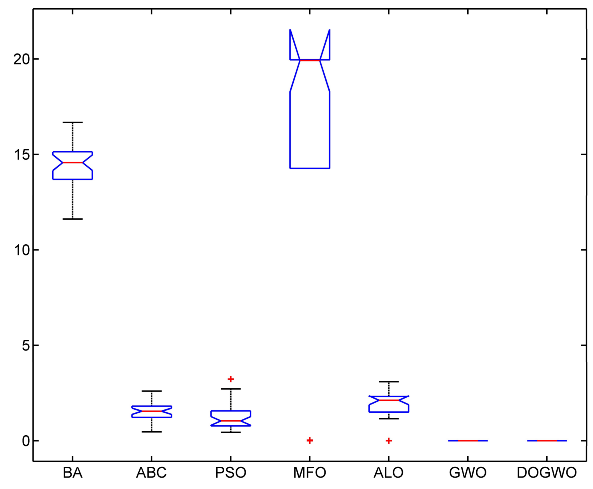


Figure 27. ANOVA test of global minimum for F9.

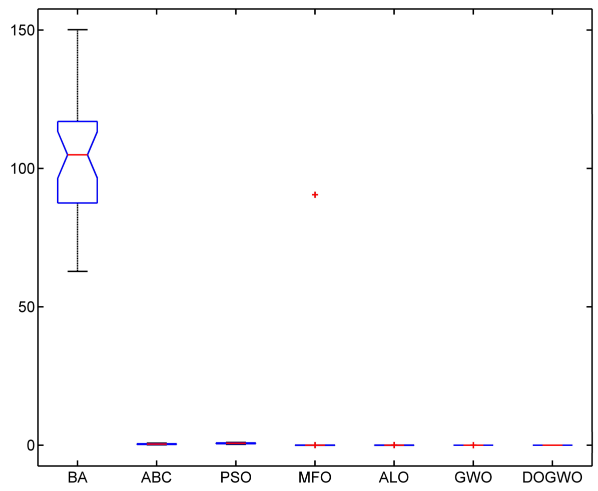


Figure 28. ANOVA test of global minimum for F10.

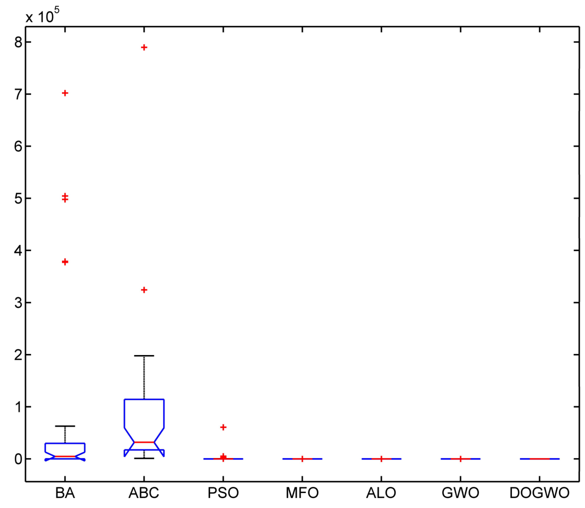


Figure 29. ANOVA test of global minimum for F11.

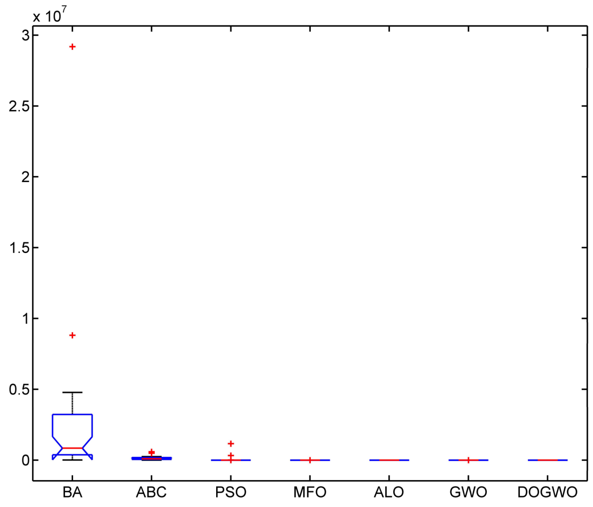


Figure 30. ANOVA test of global minimum for F12.

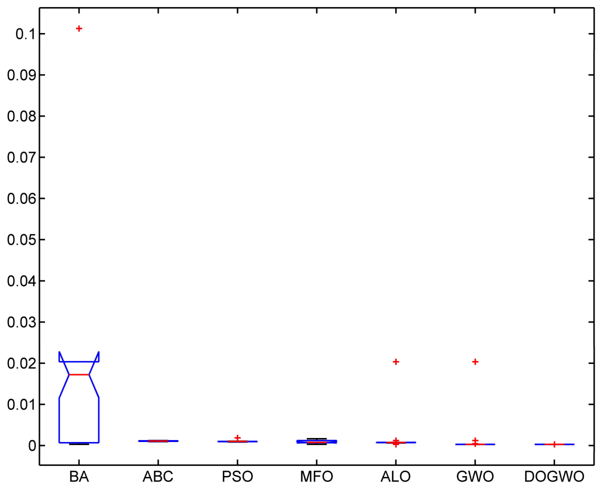


Figure 31. ANOVA test of global minimum for F14.

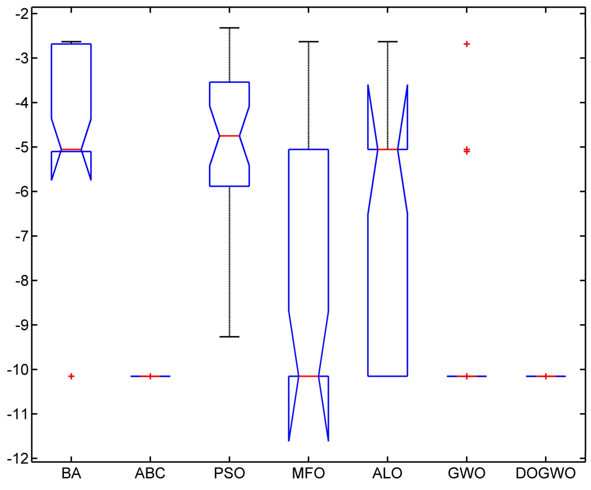
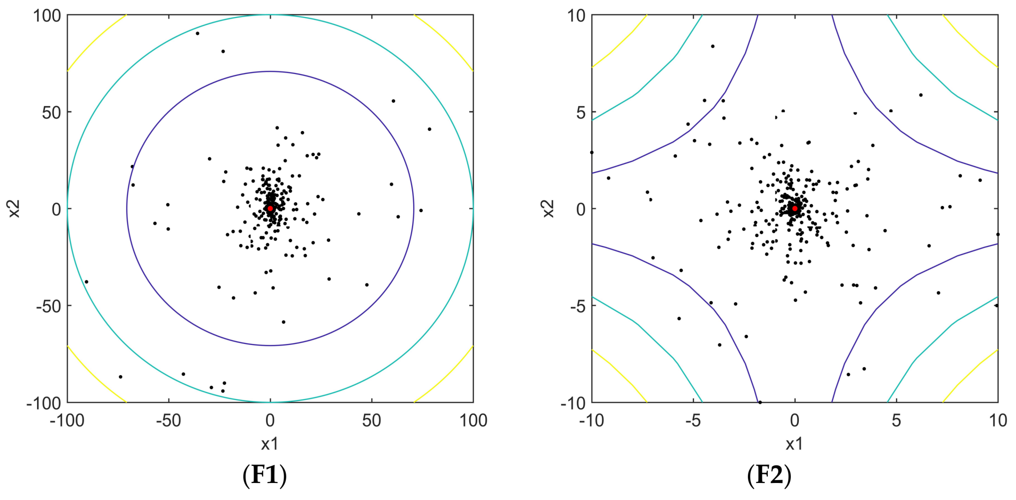


Figure 32. ANOVA test of global minimum for F21.



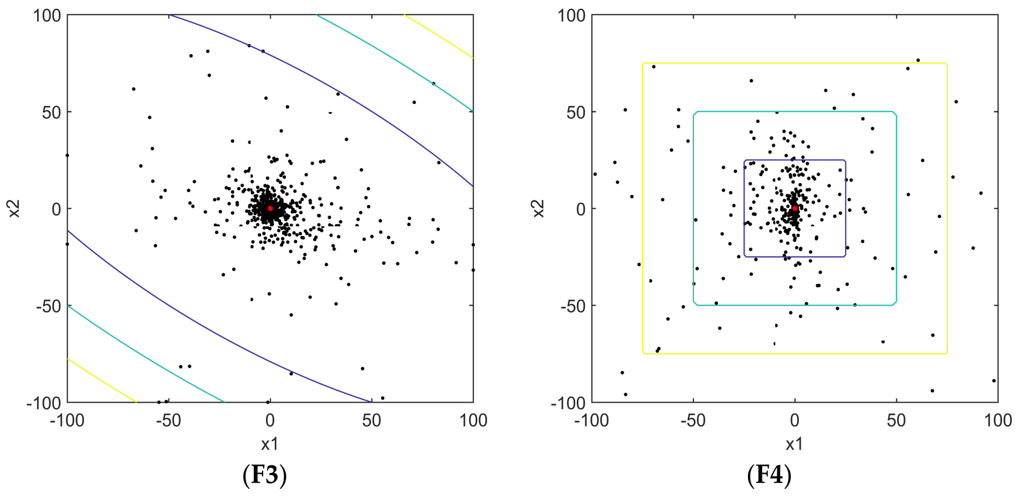


Figure 33. *Cont*.

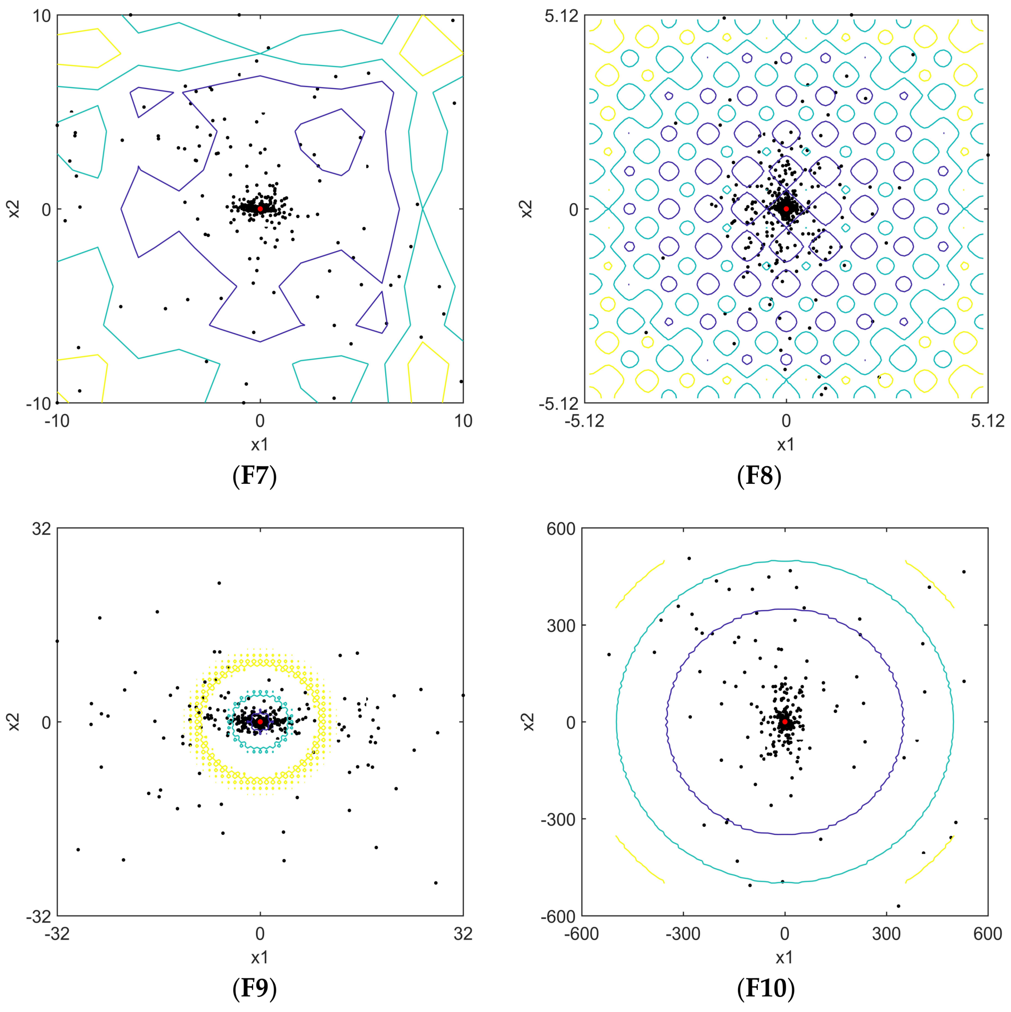


Figure 33. Search history of benchmark functions where N = 10, Iterations = 500.

4.3. Analysis and Discussion

Table 3 shows the mean, best, worst, and standard deviation fitness function values obtained by seven algorithms. From Table 3, the results of the proposed DOGWO are very competitive. It can be observed that for the 23 benchmark functions, DOGWO was superior in performance in the worst, best, mean optimal values, and standard deviation as compared to the original GWO. This indicates that the DGOBL strategy is very effective to increase the population diversity which, consequently, can enhance the performance of the GWO. Moreover, the DOGWO provided a better performance on unimodal benchmark functions F1, F2, F3, F4 as well as the multimodal benchmark functions F7, F8, F10 in comparison to other algorithms. When the DOGWO converged to the global optimum accurately, the standard deviations of these functions were also zeros. For F5 and F9, the precision of best optimal value, worst optimal value, mean optimal value, and standard deviation of DOGWO were collectively better than other algorithms. In addition, for fixed-dimension multimodal functions F14, F15, F16, F17, F18, F19, F21, F22, and F23, DOGWO could find the optimum with small standard deviations. This shows that the proposed DOGWO algorithm performed well in both exploitation and exploration. Its exploitation ability helped to converge to the optimum and the exploration ability assisted with local optima avoidance. The DOGWO also had very competitive optimization ability with high calculation precision and strong robustness for these test functions. It should be mentioned that the enhanced performance of DOGWO took advantages of the dynamic generalized opposition-based learning strategy, as the DGOBL strategy enhanced the population diversity, assisted the algorithm to explore more promising regions of the search space, and helped with local optima avoidance. For F6 and F12, ALO achieved a better calculation value than DOGWO; ABC showed better performance for F13. However, DOGWO presented a promising performance for the majority of the benchmark functions. Hence, a conclusion can be easily drawn that the DOGWO has a great potential for solving diverse optimization problems from the results in Table 3.

Figures 4–18 show the convergence curves of F1–F12, F14, F19, and F23. Red and blue asterisked dotted curves were plotted by DOGWO and GWO. Brown, green, and cyan-dotted curves were drawn by ABC, BA, and PSO, respectively. From these figures, it was demonstrated that DOGWO converged rapidly towards the optimum and exploited the optimum accurately for most of benchmark functions. It can also be concluded from the figures that DOGWO had a faster convergence rate and a better calculation precision in contrast to other algorithms. Moreover, Figures 19–32 depict the box plot of ANOVA tests of global optimum for F1–F12, F14, and F21. It can be easily determined that the fluctuate of global optimum of DOGWO was much smaller than other algorithms for most test functions, and the number of outlier was less than other algorithms. This suggests that the proposed DOGWO exhibited strong stability and robustness.

Furthermore, another test was conducted to draw the search history of search agents by which we could easily perceive the search process of DOGWO. Note that the search population number was 10 and the figures were drawn after 500 iterations. It can be observed from Figure 33 that the search agents of DOGWO tended to extensively search for promising regions of the search space and exploit the best solution. We found that most of candidate search agents were concentrated in the optimum area and other search agents were evenly dispersed in various regions of the search space. Hence, it can be stated that DOGWO had a strong ability of searching for the global optimum and avoiding the local optimum.

5. Conclusions and Future Works

This paper presents a novel GWO algorithm known as DOGWO through the use of a dynamic generalized opposition-based learning strategy (DGOBL). The DGOBL may enhance the likelihood of finding better solutions by transforming the fixed search space to a dynamic search space. The strategy also enhances the diversity of a search population which can accelerate the convergence speed, improve the calculation precision, and avoid local optima to some extent. From the results of the 23 benchmark functions, the proposed DOGWO outperformed other optimization algorithms mentioned in this paper on most benchmark functions and was comparable with partial functions. DOGWO exhibited a fast convergence speed, high precision, and a relatively high robustness and stability.

In the future, there remain questions that need to be addressed. To further test the performance of the proposed algorithm, its application to a well-known NP-hard problem, namely the resource constrained project scheduling problem (RCPSP), could be investigated. Moreover, the proposed algorithm could be applied to various practical engineering design problems, such as optimal task scheduling in cloud environments.

Acknowledgments: This work is supported by the National Science Foundation of China under Grants No. 61274025 and the Program for Youth Talents of Institute of Acoustics, Chinese Academy of Sciences No: QNYC201622. Special thanks to reviewers and editors for their constructive suggestions and work.

Author Contributions: Y.X. designed the algorithm, performed the experiments, and made the analysis; D.W. contribute materials and analysis tools; L.W. provided suggestions about both the algorithm and the experiment.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Blum, C.; Aguilera, M.J.B.; Roli, A.; Sampels, M. *Hybrid Metaheuristics, an Emerging Approach to Optimization*; Springer: Berlin, Germany, 2008.
2. Raidl, G.R.; Puchinger, J. *Combining (Integer) Linear Programming Techniques and Metaheuristics for Combinatorial Optimization*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 31–62.

3. Blum, C.; Cotta, C.; Fernández, A.J.; Gallardo, J.E.; Mastrolilli, M. *Hybridizations of Metaheuristics with Branch & Bound Derivates*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 85–116.
4. D'Andreagiovanni, F. On Improving the Capacity of Solving Large-scale Wireless Network Design Problems by Genetic Algorithms. In *Applications of Evolutionary Computation. EvoApplications. Lecture Notes in Computer Science*; Di Chio, C., Ed.; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6625, pp. 11–20.
5. D'Andreagiovanni, F.; Krolikowski, J.; Pulaj, J. A fast hybrid primal heuristic for multiband robust capacitated network design with multiple time periods. *Appl. Soft. Comput.* **2015**, *26*, 497–507. [CrossRef]
6. Egea, J.A.; Banga, J.R. Extended ant colony optimization for non-convex mixed integer nonlinear programming. *Comput. Oper. Res.* **2009**, *36*, 2217–2229.
7. Bianchi, L.; Dorigo, M.; Gambardella, L.M.; Gutjahr, W.J. A survey on optimization metaheuristics for stochastic combinatorial optimization. *Nat. Comput.* **2009**, *8*, 239–287. [CrossRef]
8. Cornuéjols, G. Valid inequalities for mixed integer linear programs. *Math. Program.* **2008**, *112*, 3–44. [CrossRef]
9. Murty, K.G. *Nonlinear Programming Theory and Algorithms: Nonlinear Programming Theory and Algorithms*, 3rd ed.; Wiley: New York, NY, USA, 1979.
10. Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*; Addison-Wesley Publishing Company: Boston, MA, USA, 1989; pp. 2104–2116.
11. Simon, D. Biogeography-Based Optimization. *IEEE Trans. Evolut. Comput.* **2008**, *12*, 702–713. [CrossRef]
12. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
13. Połap, D.; Woz´niak, M. Polar Bear Optimization Algorithm: Meta-Heuristic with Fast Population Movement and Dynamic Birth and Death Mechanism. *Symmetry* **2017**, *9*, 203. [CrossRef]
14. Bertsimas, D.; Tsitsiklis, J. Simulated Annealing. *Stat. Sci.* **1993**, *8*, 10–15. [CrossRef]
15. Rashedi, E.; Nezamabadi, P.H.; Saryazdi, S. GSA: A Gravitational Search Algorithm. *Intell. Inf. Manag.* **2012**, *4*, 390–395. [CrossRef]
16. Farahmandian, M.; Hatamlou, A. Solving optimization problem using black hole algorithm. *J. Comput. Sci. Technol.* **2015**, *4*, 68–74. [CrossRef]
17. Kaveh, A.; Khayatazad, M. A new meta-heuristic method: Ray Optimization. *Comput. Struct.* **2012**, *112–113*, 283–294. [CrossRef]
18. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
19. Dorigo, M.; Birattari, M.; Stutzle, T. Ant Colony Optimization. *IEEE Comput. Intell. Mag.* **2007**, *1*, 28–39. [CrossRef]
20. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471. [CrossRef]
21. Yang, X.S. Firefly Algorithms for Multimodal Optimization. *Mathematics* **2010**, *5792*, 169–178.
22. Yang, X.S. A New Metaheuristic Bat-Inspired Algorithm. *Comput. Knowl. Technol.* **2010**, *284*, 65–74.
23. Yang, X.S.; Deb, S. Cuckoo Search via Levy Flights. In Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, 9–11 December 2009; pp. 210–214.
24. Cuevas, E.; Cienfuegos, M.; Zaldívar, D. A swarm optimization algorithm inspired in the behavior of the social-spider. *Expert Syst. Appl.* **2014**, *40*, 6374–6384. [CrossRef]
25. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
26. Mirjalili, S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput. Appl.* **2016**, *27*, 1053–1073. [CrossRef]
27. Mirjalili, S. The Ant Lion Optimizer. *Adv. Eng. Softw.* **2015**, *83*, 80–98. [CrossRef]
28. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl.-Based Syst.* **2015**, *89*, 228–249. [CrossRef]
29. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
30. Shayeghi, H.; Asefi, S.; Younesi, A. Tuning and comparing different power system stabilizers using different performance indices applying GWO algorithm. In Proceedings of the International Comprehensive Competition Conference on Engineering Sciences, Iran, Anzali, 8 September 2016.
31. Mohanty, S.; Subudhi, B.; Ray, P.K. A Grey Wolf-Assisted Perturb & Observe MPPT Algorithm for a PV System. *IEEE Trans. Energy Conv.* **2017**, *32*, 340–347.

32. Hameed, I.A.; Bye, R.T.; Osen, O.L. Grey wolf optimizer (GWO) for automated offshore crane design. In Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece, 6–9 December 2017.
33. Siavash, M.; Pfeifer, C.; Rahiminejad, A. Reconfiguration of Smart Distribution Network in the Presence of Renewable DG's Using GWO Algorithm. *IOP Conf. Ser. Earth Environ. Sci.* **2017**, *83*, 012003. [CrossRef]
34. Emary, E.; Zawbaa, H.M.; Grosan, C. Experienced Grey Wolf Optimizer through Reinforcement Learning and Neural Networks. *IEEE Trans. Neural Netw. Learn.* **2018**, *29*, 681–694. [CrossRef] [PubMed]
35. Zawbaa, H.M.; Emary, E.; Grosan, C.; Snasel, V. Large-dimensionality small-instance set feature selection: A hybrid bioinspired heuristic approach. *Swarm. Evol. Comput.* **2018**. [CrossRef]
36. Faris, H.; Aljarah, I.; Al-Betar, M.A. Grey wolf optimizer: A review of recent variants and applications. *Neural Comput. Appl.* **2017**, *22*, 1–23. [CrossRef]
37. Rodríguez, L.; Castillo, O.; Soria, J. A Fuzzy Hierarchical Operator in the Grey Wolf Optimizer Algorithm. *Appl. Soft Comput.* **2017**, *57*, 315–328. [CrossRef]
38. Emary, E.; Zawbaa, H.M.; Hassanien, A.E. Binary Grey Wolf Optimization Approaches for Feature Selection. *Neurocomputing* **2016**, *172*, 371–381. [CrossRef]
39. Emary, E.; Zawbaa, H.M. Impact of chaos functions on modern swarm optimizers. *PLoS ONE* **2016**, *11*, e0158738. [CrossRef] [PubMed]
40. Kohli, M.; Arora, S. Chaotic grey wolf optimization algorithm for constrained optimization problems. *J. Comput. Des. Eng.* **2017**, 1–15. [CrossRef]
41. Malik, M.R.S.; Mohideen, E.R.; Ali, L. Weighted distance Grey wolf optimizer for global optimization problems. In Proceedings of the 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Kanazawa, Japan, 26–28 June 2017; pp. 1–6.
42. Heidari, A.A.; Pahlavani, P. An efficient modified grey wolf optimizer with Lévy flight for optimization tasks. *Appl. Soft Comput.* **2017**, *60*, 115–134. [CrossRef]
43. Mittal, N.; Sohi, B.S.; Sohi, B.S. Modified Grey Wolf Optimizer for Global Engineering Optimization. *Appl. Comput. Intell. Soft Comput.* **2016**, *4598*, 1–16. [CrossRef]
44. Muro, C.; Escobedo, R.; Spector, L. Wolf-pack (*Canis lupus*) hunting strategies emerge from simple rules in computational simulations. *Behav. Process.* **2011**, *88*, 192–197. [CrossRef] [PubMed]
45. Tizhoosh, H.R. Opposition-based learning: A new scheme for machine intelligence. In Proceedings of the International Conference on Computation Intelligence on Modeling Control Automation and International Conference on Intelligent Agents, Web Technologies Internet Commerce, Vienna, Austria, 28–30 November 2005; pp. 695–701.
46. Rahnamayan, S.; Tizhoosh, H.R.; Salama, M.M.A. Opposition-based differential evolution algorithms. In Proceedings of the IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, 16–21 July 2006; pp. 2010–2017.
47. Rahnamayan, S.; Tizhoosh, H.R.; Salama, M.M.A. Opposition-based differential evolution for optimization of noisy problems. In Proceedings of the IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, 16–21 July 2006; pp. 1865–1872.
48. Wang, H.; Li, H.; Liu, Y.; Li, C.; Zeng, S. Opposition based particle swarm algorithm with Cauchy mutation. In Proceedings of the IEEE Congress on Evolutionary Computation, Singapore, 25–28 September 2007; pp. 4750–4756.
49. Rahnamayan, S.; Tizhoosh, H.R.; Salama, M.M.A. Opposition-based differential evolution. *IEEE Trans. Evol. Comput.* **2008**, *2*, 64–79. [CrossRef]
50. Haiping, M.; Xieyong, R.; Baogen, J. Oppositional ant colony optimization algorithm and its application to fault monitoring. In Proceedings of the 29th Chinese Control Conference (CCC), Beijing, China, 29–31 July 2010; pp. 3895–3903.
51. Lin, Z.Y.; Wang, L.L. A new opposition-based compact genetic algorithm with fluctuation. *J. Comput. Inf. Syst.* **2010**, *6*, 897–904.
52. Shaw, B.; Mukherjee, V.; Ghoshal, S.P. A novel opposition-based gravitational search algorithm for combined economic and emission dispatch problems of power systems. *Int. J. Electr. Power Energy Syst.* **2012**, *35*, 21–33. [CrossRef]

53. Wang, S.W.; Ding, L.X.; Xie, C.W.; Guo, Z.L.; Hu, Y.R. A hybrid differential evolution with elite opposition-based learning. *J. Wuhan Univ. (Nat. Sci. Ed.)* **2013**, *59*, 111–116.
54. Zhao, R.X.; Luo, Q.F.; Zhou, Y.Q. Elite opposition-based social spider optimization algorithm for global function optimization. *Algorithms* **2017**, *10*, 9. [CrossRef]
55. Wang, H.; Wu, Z.; Liu, Y. Space transformation search: A new evolutionary technique. In Proceedings of the First ACM/SIGEVO Summit on Genetic and Evolutionary Computation Conference, Shanghai, China, 12–14 June 2009; pp. 537–544.
56. Wang, H.; Wu, Z.; Rahnamayan, S. Enhancing particle swarm optimization using generalized opposition-based learning. *Inf. Sci.* **2011**, *181*, 4699–4714. [CrossRef]
57. Wang, H.; Rahnamay, S.; Wu, Z. Parallel differential evolution with self-adapting control parameters and generalized opposition-based learning for solving high-dimensional optimization problems. *J. Parallel Distrib. Comput.* **2013**, *73*, 62–73. [CrossRef]