

Article

A Novel Complex-Valued Encoding Grey Wolf Optimization Algorithm

Qifang Luo ^{1,2}, Sen Zhang ¹, Zhiming Li ¹ and Yongquan Zhou ^{1,2,*}

Received: 6 November 2015; Accepted: 10 December 2015; Published: 30 December 2015

Academic Editor: Javier Del Ser Lorente

¹ College of Information Science and Engineering, Guangxi University for Nationalities, Nanning 530006, China; luoqifang@mail.gxun.cn (Q.L.); zhangsen98k@163.com (S.Z.); chimingle@163.com (Z.L.)

² Key Laboratory of Guangxi High Schools Complex System and Computational Intelligence, Nanning 530006, China

* Correspondence: zhouyongquan@mail.gxun.cn; Tel.: +86-771-3260264

Abstract: Grey wolf optimization (GWO) is one of the recently proposed heuristic algorithms imitating the leadership hierarchy and hunting mechanism of grey wolves in nature. The aim of these algorithms is to perform global optimization. This paper presents a modified GWO algorithm based on complex-valued encoding; namely the complex-valued encoding grey wolf optimization (CGWO). We use CGWO to test 16 unconstrained benchmark functions with seven different scales and infinite impulse response (IIR) model identification. Compared to the real-valued GWO algorithm and other optimization algorithms; the CGWO performs significantly better in terms of accuracy; robustness; and convergence speed.

Keywords: complex-valued encoding; grey wolf optimization; diploid; test functions; IIR model identification

1. Introduction

Over the last few decades, various heuristic optimizations have been developed to solve complex computational problems. Some of the most popular are: particle swarm optimization (PSO) [1], genetic algorithm (GA) [2], differential evolution (DE) [3,4], simulated annealing (SA) [5], ant colony optimization (ACO) [6], artificial bee colony optimization (ABC) [7], gravitational search algorithm (GSA) [8,9], charged system search (CSS) [10], magnetic optimization algorithm (MOA) [11], biogeography-based optimization algorithm (BBO) [12,13], and teaching learning-based optimization (TLBO) [14]. The swarm intelligence optimization algorithm can flexibly and effectively deal with different problems that conventional optimization techniques cannot solve. For this reason, optimization techniques have been widely used in many domains of study. In addition, it has been demonstrated that there is no heuristic optimization that can solve all optimization problems [15]. In other words, existing algorithms give satisfactory results in solving some problems, but not all problems. Therefore, several new heuristic algorithms are proposed every year, and research in this field is active.

The real-value grey wolf optimization (GWO) is a new population-based meta-heuristic approach initially proposed by Mirjalili *et al.* (2014) [16]. Due to the fact that the GWO algorithm is simple, flexible and efficient, it can be successfully applied to practical applications [17–19]. There are many modified versions of the GWO algorithm. For example, some people study the parameters of the GWO algorithm or combine the GWO algorithm with other heuristic optimization algorithms. However, these methods all use binary or decimal encoders to position the grey wolf, the individual gene contains information to be limited.

The complex-valued encoding method has been applied in expressing neural network weights [20] and representing individual genes for evolutionary algorithms [21]. It uses a diploid to express genes and greatly expand the information capacity of the individual. Starting from the individual encoding method, this article studies the improvement of the performance of the grey wolf with the complex-valued method. The value of the independent variables for an objective function is determined by modules and the sign of the independent variables is determined by angles. For the GWO algorithm, the grey wolf individual is divided into two parts, namely: a real part gene and an imaginary gene. Additionally, the two parts represent a variable being able to enhance the information of a grey wolf individual and a diverse population, avoiding the local optimum. Based on the above advantages, we provide a new method for the GWO algorithm to solve practical problems.

The rest of the paper is organized as follows: Section 2 presents a brief introduction to GWO. Section 3 discusses the basic principles of a complex-valued version of GWO. The experimental results of test functions and infinite impulse response (IIR) system identification problem are shown in Sections 4 and 5 respectively. Finally, Section 6 concludes the work and suggests some directions for future study.

2. Grey Wolf Optimization (GWO)

The GWO algorithm simulates the hunting and social leadership of grey wolves in nature [16]. The algorithm is simple, robust, and has been used in various complex problems. In the GWO algorithm, the colony of grey wolves is divided into four groups: alpha (α), beta (β), delta (δ), and omega (ω). In every iteration, the first three best candidate solutions are named α , β , and δ . The rest of the grey wolves are considered as ω , and are guided by α , β , and δ to find the better solutions. The mathematical model of the ω wolves' encircling process is as follows [16]:

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_p(t) - \vec{X}(t) \right| \quad (1)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (2)$$

where t indicates the current iteration, $\vec{C} = 2 \cdot \vec{r}_2$, $\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}$, \vec{X}_p is the position vector of the prey, \vec{X} indicates the position vector of a grey wolf, \vec{a} is gradually decreased from 2 to 0, and r_1, r_2 are random numbers over range $[0, 1]$.

It should be noted here that, during optimization, the ω wolves update their positions around α , β , and δ . Therefore, ω wolves are able to reposition with respect to α , β , and δ . The mathematical model of ω wolves readjusting the positions is as follows [16]:

$$\vec{D}_\alpha = \left| \vec{C}_1 \cdot \vec{X}_\alpha - \vec{X} \right|, \vec{D}_\beta = \left| \vec{C}_2 \cdot \vec{X}_\beta - \vec{X} \right|, \vec{D}_\delta = \left| \vec{C}_3 \cdot \vec{X}_\delta - \vec{X} \right| \quad (3)$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha), \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta), \vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta) \quad (4)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (5)$$

where \vec{X}_α is the position of the alpha, \vec{X}_β is the position of the beta, \vec{X}_δ is the position of the delta, $\vec{C}_1, \vec{C}_2, \vec{C}_3$ and $\vec{A}_1, \vec{A}_2, \vec{A}_3$ are all random vectors, \vec{X} is the position of the current solution, and t indicates the number of iterations.

As mentioned before, ω wolves update their positions based on positions of α , β , and δ . There are two vectors: \vec{A} and \vec{C} that are random and adaptive vectors which help the algorithm to escape from local optima. When $|A| > 1$, half of the iterations are devoted to exploration. The range of C is $2 \leq C \leq 0$. The vector C also improves exploration when $C > 1$. In contrast, the other half of

the iterations are dedicated to exploitation when $|A| < 1$, and the exploitation is also emphasized when $C < 1$. Note here that A decreases linearly over the course of iterations in order to emphasize exploitation, and C is provided with a random value at all times to emphasize exploration not only during initial iterations but also during final iterations. The pseudo code of GWO (Algorithm 1) is presented as follows:

Algorithm 1. GWO algorithm

1. Initialize the grey wolf population $X_i = (i = 1, 2, \dots, n)$
 2. Initialize a , A , and C
 3. **For** all X_i **do**
 4. Calculate fitness $F(X_i)$
 5. **End for**
 6. Get the first three best wolves as X_α , X_β , and X_δ
 7. **While** ($t < \text{Max number of iterations}$)
 8. **For** each search agent
 9. Update the position of the current search agent by Equation (5)
 10. **End for**
 11. Update a , A , and C
 12. **For** all X_i **do**
 13. Calculate fitness $F(X_i)$
 14. **End for**
 15. Update X_α , X_β , and X_δ
 16. $t = t + 1$
 17. **End while**
 18. Return X_α
-

3. Complex-Valued Encoding Grey Wolf Optimization Algorithm (CGWO)

3.1. The Complex-Valued Encoding Method

In order to contain the M -variable function optimization problem within the M complex, corresponding to the M complex, the grey wolf location is recorded as follows:

$$x_p = R_p + iI_p \quad p = 1, 2, \dots, M \quad (6)$$

The genes of grey wolves can be expressed as a diploid, which is recorded as (R_p, I_p) . Where R_p indicates the real part of the variable, I_p indicates the imaginary part of the variable. Therefore, the i th grey wolf can be seen in Table 1.

Table 1. Grey wolf chromosome model.

Gene ₁	Gene ₂	Gene _i	Gene _M
(R_{p1}, I_{p1})	(R_{p2}, I_{p2})	...	(R_{pM}, I_{pM})

3.1.1. Initializing the Complex-Valued Encoding Population

The variable interval of the function is set to $[A_L, B_L]$, $L = 1, 2, \dots, M$. Generate M modulus and M phase angle randomly, and they form a complex-valued M . The relationship of the modulus and the phase is provided as follows:

$$\rho_L = \left[0, \frac{B_L - A_L}{2} \right], \quad \theta_L = [-2\pi, 2\pi], \quad L = 1, 2, \dots, M \quad (7)$$

$$R_L + iI_L = \rho_L(\cos\theta_L + i\sin\theta_L), \quad L = 1, 2, \dots, M \quad (8)$$

So, we obtain M real and imaginary parts. The real and imaginary parts are updated according to Section 3.1.2.

3.1.2. The Updating Method of CGWO

Update the Real Parts

$$\vec{D}_{R\alpha} = \left| \vec{C}_1 \cdot \vec{X}_{R\alpha} - \vec{X}_R \right|, \vec{D}_{R\beta} = \left| \vec{C}_2 \cdot \vec{X}_{R\beta} - \vec{X}_R \right|, \vec{D}_{R\delta} = \left| \vec{C}_3 \cdot \vec{X}_{R\delta} - \vec{X}_R \right| \quad (9)$$

$$\vec{X}_{R1} = \vec{X}_{R\alpha} - \vec{A}_1 \cdot (\vec{D}_{R\alpha}), \vec{X}_{R2} = \vec{X}_{R\beta} - \vec{A}_2 \cdot (\vec{D}_{R\beta}), \vec{X}_{R3} = \vec{X}_{R\delta} - \vec{A}_3 \cdot (\vec{D}_{R\delta}) \quad (10)$$

$$\vec{X}_R(t+1) = \frac{\vec{X}_{R1} + \vec{X}_{R2} + \vec{X}_{R3}}{3} \quad (11)$$

Update the Imaginary Parts

$$\vec{D}_{I\alpha} = \left| \vec{C}_1 \cdot \vec{X}_{I\alpha} - \vec{X}_I \right|, \vec{D}_{I\beta} = \left| \vec{C}_2 \cdot \vec{X}_{I\beta} - \vec{X}_I \right|, \vec{D}_{I\delta} = \left| \vec{C}_3 \cdot \vec{X}_{I\delta} - \vec{X}_I \right| \quad (12)$$

$$\vec{X}_{I1} = \vec{X}_{I\alpha} - \vec{A}_1 \cdot (\vec{D}_{I\alpha}), \vec{X}_{I2} = \vec{X}_{I\beta} - \vec{A}_2 \cdot (\vec{D}_{I\beta}), \vec{X}_{I3} = \vec{X}_{I\delta} - \vec{A}_3 \cdot (\vec{D}_{I\delta}) \quad (13)$$

$$\vec{X}_I(t+1) = \frac{\vec{X}_{I1} + \vec{X}_{I2} + \vec{X}_{I3}}{3} \quad (14)$$

where \vec{D}_R and \vec{D}_L indicate the real and imaginary of the approximate distance between the current solution, alpha, beta, and delta respectively. \vec{X}_R and \vec{X}_L indicate the real and imaginary parts of the final position of the current solution. It should be noted here that \vec{A} and \vec{C} are coefficient vectors, so it is not necessary for them to be converted to real and imaginary.

3.1.3. The Calculation Method of Fitness Value

The complex domain is composed of real and imaginary parts, so the complex number must be changed into a real number when solving the fitness function value. The real value of an objective function is determined by its modulus, and the sign is determined by its phase angle. The concrete practices are as follows:

$$\rho_n = \sqrt{X_{R_n}^2 + X_{I_n}^2}, \quad n = 1, 2, \dots, M \quad (15)$$

$$RV_n = \rho_n \operatorname{sgn}(\sin(\frac{X_{I_n}}{\rho_n})) + \frac{B_L + A_L}{2}, \quad n = 1, 2, \dots, M \quad (16)$$

where ρ_n indicates the n th dimension module, $X_{R_n}^2, X_{I_n}^2$ indicate the real part and imaginary part of the n th dimension, respectively, and RV_n is converted into a real variable.

3.2. CGWO Algorithm

Based on the above analysis, complex-valued encoding, which can be considered as a valid global optimization strategy, is applied to the grey wolf optimizer. The real and imaginary parts of complex numbers are updated independently due to two-dimensional properties of complex numbers. The strategy greatly expands the amount of information contained in the individual gene and enhances the diversity of individual populations. In addition, to improve the local search ability of the algorithm, the differential evolution strategy “DE/best/2” [22] is introduced. Under the circumstances, CGWO has the potential to balance global and local searches. In the following section, various benchmark functions are employed to explore the effectiveness of CGWO. The pseudo code of CGWO (Algorithm 2) is as follows:

Algorithm 2. CGWO algorithm

```

1. Initialize the grey wolf population: Let  $\rho_L = \left[0, \frac{B_L - A_L}{2}\right]$ ,  $\theta_L = [-2\pi, 2\pi]$ ,  $L = 1, 2, \dots, M$ 
2. Get the real and imaginary part of the complex number by Equation (8)
3. Convert to real variables by Equations (15) and (16)
4. For all  $X_i$  do
5.     Calculate fitness  $F(X_i)$ 
6. End for
7. Get the first three best wolves as  $X_\alpha$ ,  $X_\beta$ , and  $X_\delta$ 
8. While ( $t < \text{Max number of iterations}$ )
9.     For each search agent
10.        Update the real part of the current search agent by Equations (9–11)
11.        Update the imaginary part of the current search agent by Equations (12–14)
12.    End for
13.    Modify the real and imaginary part by “DE/best/2”
14.    Select randomly  $r_2 \neq r_2 \neq r_3 \neq r_4 \neq i$ 
15.     $\vec{X}_{new} = \vec{X}_\alpha + F * (\vec{X}_{r1} - \vec{X}_{r2}) + F * (\vec{X}_{r3} - \vec{X}_{r4})$ 
16.    Update  $a$ ,  $A$ , and  $C$ 
17.    Convert to real variables by Equations (15) and (16)
18.    For all  $X_i$  do
19.        Calculate fitness  $F(X_i)$ 
20.    End for
21.    Update  $X_\alpha$ ,  $X_\beta$ , and  $X_\delta$ 
22.     $t = t + 1$ 
23. End while
24. Return  $X_\alpha$ 

```

4. Experimental Results and Discussion

4.1. Simulation Platform

All algorithms are tested in Matlab R2012a (7.14), and experiments are executed on AMD Athlon (tm) II X4 640 Processor 3.00 GHz PC with 3G RAM. Operating system is Windows 7.

4.2. Benchmark Functions

To evaluate the performance of the proposed method, termed CGWO, 16 standard benchmark functions are used to demonstrate the performance of the algorithm [23–25]. They can be divided into two groups: unimodal and multimodal functions. Table 1 lists the functions, where Dim shows dimension of the function, Range is the boundary of search space of the function, and f_{min} is the minimum of the function.

We compare ABC [7], GGSA [9], CGWO, GWO [16] on benchmark functions to verify the effectiveness of these algorithms dealing with problems of different scales. Moreover, in order to prove the superiority of CGWO over GWO, we derive the functions from complex-valued encoding rather than the differential evolution strategy “DE/best/2”. CGWO is also compared to the basic GWO with a differential evolution strategy called differential evolution GWO (DGWO). The five algorithms have some parameters that are shown in Table 2.

Table 2. Benchmark functions.

Benchmark Functions	Range	Dim	f_{\min}
<i>Unimodal functions</i>			
F_1 : Sphere function	[−100,100]	10,30,50,100,200,300,400	0
F_2 : Schwefel's problem 2.22	[−10,10]	10,30,50,100,200,300,400	0
F_3 : Schwefel's problem 2.21	[−100,100]	10,30,50,100,200,300,400	0
F_4 : Generalized Rosenbrock's function	[−30,30]	10,30,50,100,200,300,400	0
F_5 : Quartic function <i>i.e.</i> , Nieme	[−1.28,1.28]	10,30,50,100,200,300,400	0
F_6 : Shifted sphere function	[−100,100]	10,30,50,100,200,300,400	−450
F_7 : Shifted Schwefel's	[−100,100]	10,30,50,100,200,300,400	−450
F_8 : Generalized Rosenbrock's function	[−100,100]	10,30,50,100,200,300,400	−450
<i>Multimodal functions</i>			
F_9 : Generalized Rastrigin's function	[−5.12,5.12]	10,30,50,100,200,300,400	0
F_{10} : Ackley's function	[−32,32]	10,30,50,100,200,300,400	0
F_{11} : Generalized Griewank function	[−600,600]	10,30,50,100,200,300,400	0
F_{12} : Generalized Penalized 1 function	[−50,50]	10,30,50,100,200,300,400	0
F_{13} : Alpine function	[−10,10]	10,30,50,100,200,300,400	0
F_{14} : Shifted Rosenbrock's function	[−100,100]	10,30,50,100,200,300,400	390
F_{15} : Shifted Rastrigin's function	[−5,5]	10,30,50,100,200,300,400	−330
F_{16} : Expanded extended Griewank's plus Rosenbrock's function	[−3,1]	10,30,50,100,200,300,400	−130

The experimental initial parameters of the five algorithms are provided in Table 3, the experimental results are provided in Tables 4 and 5. The results are averaged over 20 consecutive experiments. The best results are demonstrated in bold type. The average (AVE) and standard deviation (STD) of the best solution obtained in the last iteration are reported in the form of AVE \pm STD. Note that the Matlab code of the GGSA algorithm is given in [26].

To improve the performance evaluation of evolutionary algorithms, statistical tests should be conducted [27]. In order to determine whether the results of CGWO differ from the best results of the other algorithms in a statistical method, a nonparametric test, which is known as Wilcoxon's rank-sum test [28–31], is performed at 5% significance level. The calculated p values in Wilcoxon's rank-sum test comparing CGWO and the other algorithms over all the benchmark functions are given in Tables 6–12. Usually, p values < 0.05 can be considered as sufficient evidence against the null hypothesis.

Table 3. Initial parameters of algorithms (where T indicates the maximum number of interactions; t is the current iteration).

Algorithm	Parameter	Value
CGWO	Number of grey wolves $\overrightarrow{\alpha}$	50 (dim = 10, 30, 50, 100, 200, 300, 400) Linearly decreased from 2 to 0
	Scaling factor (F)	0.1
	Max iteration	500
	Stopping criteria	Max iteration
DGWO	Number of grey wolves $\overrightarrow{\alpha}$	50 (dim = 10, 30, 50, 100, 200, 300, 400) Linearly decreased from 2 to 0
	Scaling factor (F)	0.1
	Max iteration	500
	Stopping criteria	Max iteration
GWO	Number of grey wolves $\overrightarrow{\alpha}$	50 (dim = 10, 30, 50, 100, 200, 300, 400) Linearly decreased from 2 to 0
	Max iteration	500
	Stopping criteria	Max iteration
GGSA	Number of particles c'_1 c'_2 G_0 α	50 (dim = 10, 30, 50, 100, 200, 300, 400) $(-2t^3/T^3) + 2$ $(2t^3/T^3)$ 1 20
	Max iteration	500
	Stopping criteria	Max iteration
	Number of bees Limit	50 (dim = 10, 30, 50, 100, 200, 300, 400) dim
	Max iteration	500
	Stopping criteria	Max iteration
ABC		

Table 6. *p*-values produced by Wilcoxon rank-sum test comparing CGWO (complex-valued encoding grey wolf optimization) vs. DGWO (differential evolution GWO), GWO, GGSA(gbest-guided gravitational search algorithm), and ABC (artificial bee colony optimization)minimization results of the multimodal benchmark functions at Dim = 10 ($p \geq 0.05$).

Dim = 10	F₁	F₂	F₃	F₄	F₅	F₆	F₇	F₈
CGWO vs. DGWO	2.4426E−08	8.0065E−09	6.2771E−08	6.7956E−08	6.7956E−08	6.7956E−08	1.2346E−07	1.6571E−07
CGWO vs. GWO	2.4426E−08	8.0065E−09	6.2771E−08	6.7956E−08	2.1393E−03	<u>3.5070E−01</u>	4.1658E−05	1.1045E−05
CGWO vs. GGSA	2.4426E−08	8.0065E−09	6.2771E−08	6.7956E−08	7.9479E−07	8.0065E−09	6.7956E−08	6.7956E−08
CGWO vs. ABC	2.4426E−08	8.0065E−09	6.2771E−08	6.7956E−08	6.7956E−08	9.1728E−08	6.7956E−08	9.1728E−08
Dim = 10	F₉	F₁₀	F₁₁	F₁₂	F₁₃	F₁₄	F₁₅	F₁₆
CGWO vs. DGWO	8.0065E−09	2.4037E−08	1.5149E−08	6.7956E−08	8.0065E−09	2.5629E−07	6.3761E−08	1.0646E−07
CGWO vs. GWO	<u>1.6259E−01</u>	1.7005E−07	8.2529E−04	6.9166E−07	8.0065E−09	7.5774E−06	1.1613E−07	4.9864E−02
CGWO vs. GGSA	7.3831E−09	2.4037E−08	1.5149E−08	6.7956E−08	8.0065E−09	3.7499E−04	1.5605E−07	<u>3.2348E−01</u>
CGWO vs. ABC	8.0065E−09	2.4037E−08	1.5149E−08	6.7956E−08	8.0065E−09	1.9177E−07	7.4151E−08	7.1135E−03

Table 7. *p*-values produced by Wilcoxon rank-sum test comparing CGWO vs. DGWO, GWO, GGSA, and ABC at Dim = 30 ($p \geq 0.05$ have been underlined).

Dim = 30	F₁	F₂	F₃	F₄	F₅	F₆	F₇	F₈
CGWO vs. DGWO	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	2.0616E−06	7.8980E−08
CGWO vs. GWO	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	<u>5.1153E−03</u>	6.7956E−08	2.5629E−07	2.3025E−05
CGWO vs. GGSA	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08
CGWO vs. ABC	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08
Dim = 30	F₉	F₁₀	F₁₁	F₁₂	F₁₃	F₁₄	F₁₅	F₁₆
CGWO vs. DGWO	1.5149E−08	4.7371E−08	1.1267E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08
CGWO vs. GWO	<u>3.6853E−05</u>	3.8179E−08	1.8196E−02	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	<u>4.7348E−01</u>
CGWO vs. GGSA	1.4970E−08	4.7371E−08	1.1267E−08	1.8074E−05	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08
CGWO vs. ABC	1.5149E−08	4.7371E−08	1.1267E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	7.8980E−08

Table 8. *p*-values produced by Wilcoxon rank-sum test comparing CGWO vs. DGWO, GWO, GGSA, and ABC at Dim = 50 ($p \geq 0.05$ have been underlined).

Dim = 50	F₁	F₂	F₃	F₄	F₅	F₆	F₇	F₈
CGWO vs. DGWO	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	1.3486E−03	6.7956E−08
CGWO vs. GWO	6.7956E−08	6.7956E−08	6.7956E−08	3.0566E−03	3.7051E−05	6.7956E−08	7.9479E−07	3.9388E−07
CGWO vs. GGSA	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08
CGWO vs. ABC	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08
Dim = 50	F₉	F₁₀	F₁₁	F₁₂	F₁₃	F₁₄	F₁₅	F₁₆
CGWO vs. DGWO	1.1267E−08	5.6939E−08	1.1267E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08
CGWO vs. GWO	1.1168E−08	5.6775E−08	<u>1.6493E−01</u>	6.7956E−08	6.7956E−08	1.0646E−07	6.7956E−08	1.0646E−07
CGWO vs. GGSA	1.1267E−08	5.6611E−08	1.1267E−08	6.7956E−08	6.7956E−08	6.7956E−08	1.6571E−07	5.2269E−07
CGWO vs. ABC	1.1267E−08	5.6939E−08	1.1267E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08

Table 9. *p*-values produced by Wilcoxon rank-sum test comparing CGWO vs. DGWO, GWO, GGSA, and ABC at Dim = 100 ($p \geq 0.05$ have been underlined).

Dim = 100	F₁	F₂	F₃	F₄	F₅	F₆	F₇	F₈
CGWO vs. DGWO	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	1.0373E−04	6.7956E−08
CGWO vs. GWO	6.7956E−08	6.7956E−08	2.0616E−06	<u>2.7329E−01</u>	5.1658E−06	5.6290E−04	6.7956E−08	6.7956E−08
CGWO vs. GGSA	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08
CGWO vs. ABC	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08
Dim = 100	F₉	F₁₀	F₁₁	F₁₂	F₁₃	F₁₄	F₁₅	F₁₆
CGWO vs. DGWO	5.6042E−08	6.5041E−08	8.0065E−09	6.7956E−08	6.7956E−08	4.6792E−02	6.7956E−08	6.7956E−08
CGWO vs. GWO	1.1910E−07	6.5041E−08	7.9480E−09	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	9.2780E−05
CGWO vs. GGSA	5.6042E−08	6.5041E−08	8.0065E−09	6.7956E−08	6.7956E−08	9.1266E−07	6.7956E−08	6.7956E−08
CGWO vs. ABC	5.6042E−08	6.5041E−08	8.0065E−09	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	<u>1.1355E−01</u>

Table 10. *p*-values produced by Wilcoxon rank-sum test comparing CGWO vs. DGWO, GWO, GGSA, and ABC at Dim = 200 (*p* ≥ 0.05 have been underlined).

Dim = 200	F₁	F₂	F₃	F₄	F₅	F₆	F₇	F₈
CGWO vs. DGWO	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	1.1045E−05
CGWO vs. GWO	6.7956E−08	6.7956E−08	6.7956E−08	<u>4.0936E−01</u>	1.4149E−05	6.7956E−08	7.8980E−08	<u>9.8921E−01</u>
CGWO vs. GGSA	6.7956E−08	6.7956E−08	6.7956E−08	<u>6.7956E−08</u>	6.7956E−08	7.8980E−08	2.9598E−07	6.7956E−08
CGWO vs. ABC	6.7956E−08	6.7956E−08	6.7956E−08	<u>6.7956E−08</u>	6.7956E−08	6.7956E−08	6.7956E−08	5.8736E−06
Dim = 200	F₉	F₁₀	F₁₁	F₁₂	F₁₃	F₁₄	F₁₅	F₁₆
CGWO vs. DGWO	6.6532E−08	6.7956E−08	1.9544E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08
CGWO vs. GWO	1.3953E−05	6.7956E−08	2.7851E−07	6.7956E−08	6.7956E−08	6.7956E−08	3.9874E−06	6.7956E−08
CGWO vs. GGSA	1.0292E−06	6.7956E−08	1.9544E−08	6.7956E−08	6.7956E−08	6.6737E−06	5.1153E−03	7.8980E−08
CGWO vs. ABC	6.6532E−08	6.7956E−08	1.9544E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08

Table 11. *p*-values produced by Wilcoxon rank-sum test comparing CGWO vs. DGWO, GWO, GGSA, and ABC at Dim = 300 (*p* ≥ 0.05 have been underlined).

Dim = 300	F₁	F₂	F₃	F₄	F₅	F₆	F₇	F₈
CGWO vs. DGWO	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	1.0646E−07
CGWO vs. GWO	6.7956E−08	6.7956E−08	6.7956E−08	<u>6.7956E−08</u>	<u>8.3923E−01</u>	6.7956E−08	6.7956E−08	7.8980E−08
CGWO vs. GGSA	6.7956E−08	6.7956E−08	6.7956E−08	<u>6.7956E−08</u>	6.7956E−08	6.7956E−08	7.8980E−08	5.2269E−07
CGWO vs. ABC	6.7956E−08	6.7956E−08	6.7956E−08	<u>6.7956E−08</u>	6.7956E−08	6.7956E−08	6.7956E−08	9.1728E−08
Dim = 300	F₉	F₁₀	F₁₁	F₁₂	F₁₃	F₁₄	F₁₅	F₁₆
CGWO vs. DGWO	6.7956E−08	6.7956E−08	8.0065E−09	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08
CGWO vs. GWO	1.6098E−04	6.7956E−08	8.0065E−09	6.7956E−08	6.7956E−08	6.7956E−08	9.1728E−08	6.7956E−08
CGWO vs. GGSA	1.5997E−05	6.7956E−08	8.0065E−09	6.7956E−08	6.7956E−08	6.7956E−08	9.7480E−06	6.7956E−08
CGWO vs. ABC	6.7956E−08	6.7956E−08	8.0065E−09	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08	6.7956E−08

Table 12. *p*-values produced by Wilcoxon rank-sum test comparing CGWO vs. DGWO, GWO, GGSA, and ABC at Dim = 400 (*p* ≥ 0.05 have been underlined).

Dim = 400	F₁	F₂	F₃	F₄	F₅	F₆	F₇	F₈
CGWO vs. DGWO	6.7956E–08	6.7956E–08	6.7956E–08	6.7956E–08	6.7956E–08	6.7956E–08	6.7956E–08	9.1266E–07
CGWO vs. GWO	6.7956E–08	6.7956E–08	6.7956E–08	1.2346E–07	<u>7.3527E–01</u>	6.7956E–08	9.1728E–08	<u>7.2045E–02</u>
CGWO vs. GGSA	6.7956E–08	6.7956E–08	<u>7.3527E–01</u>	6.7956E–08	6.7956E–08	6.7956E–08	1.9177E–07	2.3025E–05
CGWO vs. ABC	6.7956E–08	6.7956E–08	6.7956E–08	6.7956E–08	6.7956E–08	6.7956E–08	6.7956E–08	3.9388E–07
Dim = 400	F₉	F₁₀	F₁₁	F₁₂	F₁₃	F₁₄	F₁₅	F₁₆
CGWO vs. DGWO	1.4309E–07	6.7956E–08	3.3727E–08	6.7956E–08	6.7956E–08	6.7956E–08	6.7956E–08	6.7956E–08
CGWO vs. GWO	7.1135E–03	6.7956E–08	4.4078E–07	6.7956E–08	6.7956E–08	6.9166E–07	6.7956E–08	6.7956E–08
CGWO vs. GGSA	1.6098E–04	6.7956E–08	3.3727E–08	6.7956E–08	6.7956E–08	7.5788E–04	6.7956E–08	6.7956E–08
CGWO vs. ABC	6.7956E–08	6.7956E–08	3.3727E–08	6.7956E–08	6.7956E–08	6.7956E–08	6.7956E–08	6.7956E–08

4.3. Unimodal Benchmark Functions

The unimodal benchmarks have only one global solution without any local optima for them. Consequently, they are useful in examining the convergence capability of heuristic optimization. The results of unimodal benchmark functions are shown in Table 3. As shown in this table, for 30-D unimodal benchmark functions, CGWO has the best results for all benchmark functions. For 10-D, 50-D, 100-D, 200-D, 300-D, and 400-D, this table shows that the CGWO algorithm outperforms the other algorithms on the majority of the unimodal benchmark test functions. Therefore, the proposed algorithm has high performance in finding the global solution of unimodal benchmark test functions. According to the p values of $F_1 \sim F_8$ with different dimensions in Tables 5–11 CGWO achieves significant improvement across the majority of dimensions compared to the other algorithms. Hence, this proves that CGWO has better performance than the other algorithms in forging for a global optimum solution of unimodal benchmark functions.

Figure 1 shows the averaged convergence curves of unimodal benchmark functions with 30-D over 20 independent runs. As shown from these curves, the CGWO has the fastest convergence rate. Figure 2 indicates the anova tests of the global minimum on unimodal functions with 30-D, which shows that CGWO is the most robust method. For $F_1 \sim F_8$ functions, the other algorithms that we compared (namely, DGWO, GWO, GGSA, ABC) were only robust for a few functions and were not able to be robust for all functions. On the basis of Tables 3 and 5–11 and Figures 1 and 3 it can be concluded that the CGWO tends to find the best solution in unimodal functions with a very fast convergence rate and a more stable performance.

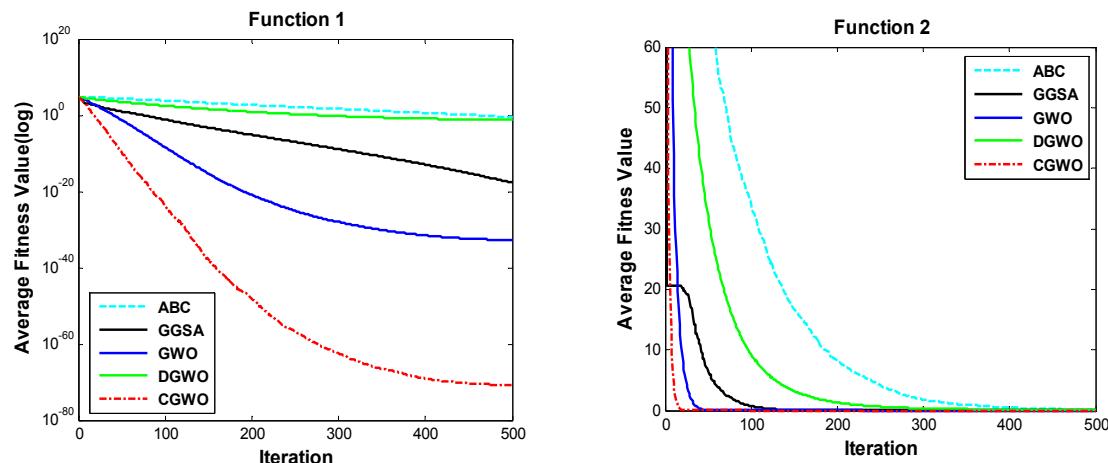


Figure 1. Cont

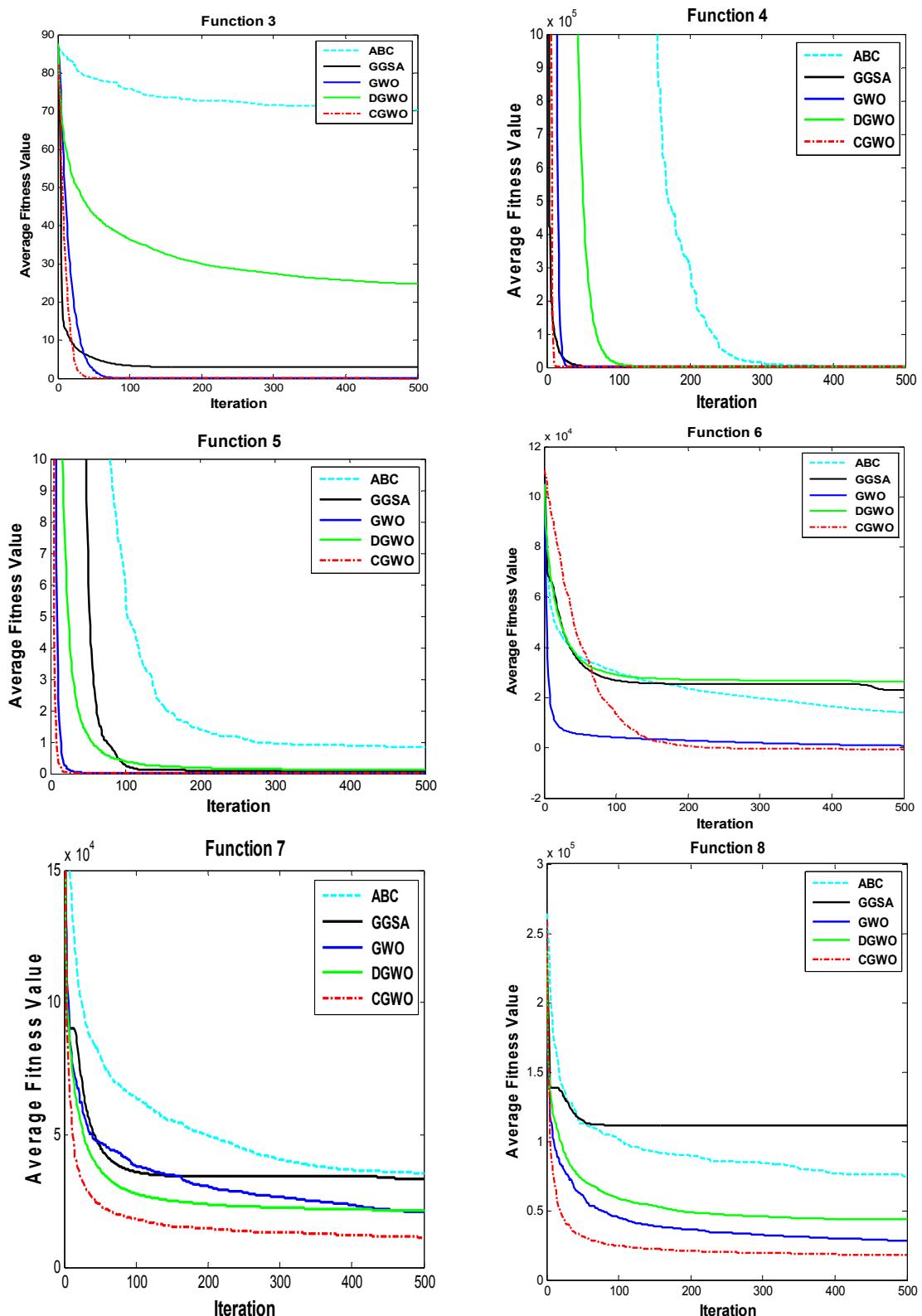


Figure 1. Convergence behavior of ABC, GGSA, GWO, DGWO, and CGWO on unimodal functions with Dim = 30.

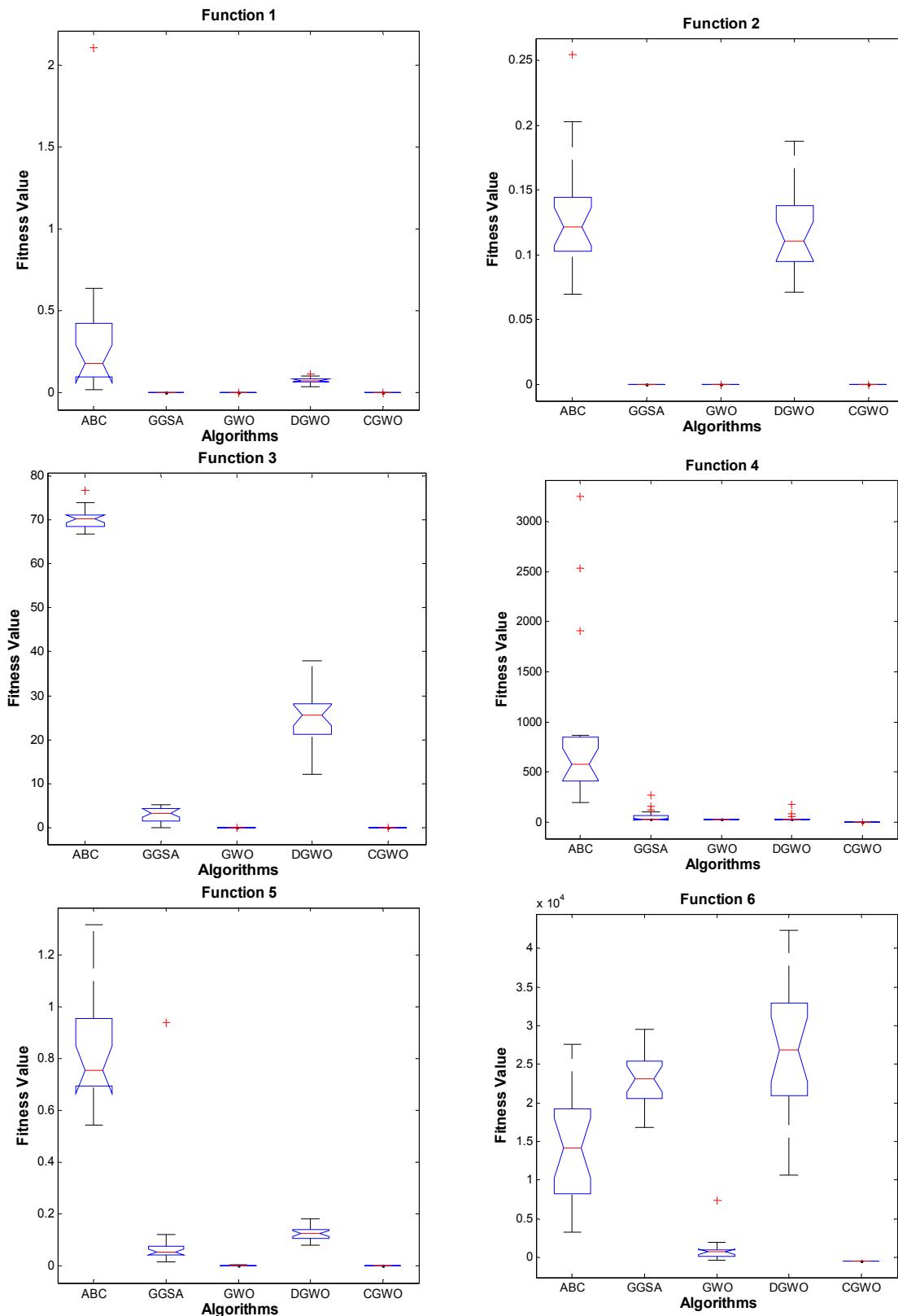


Figure 2. Cont

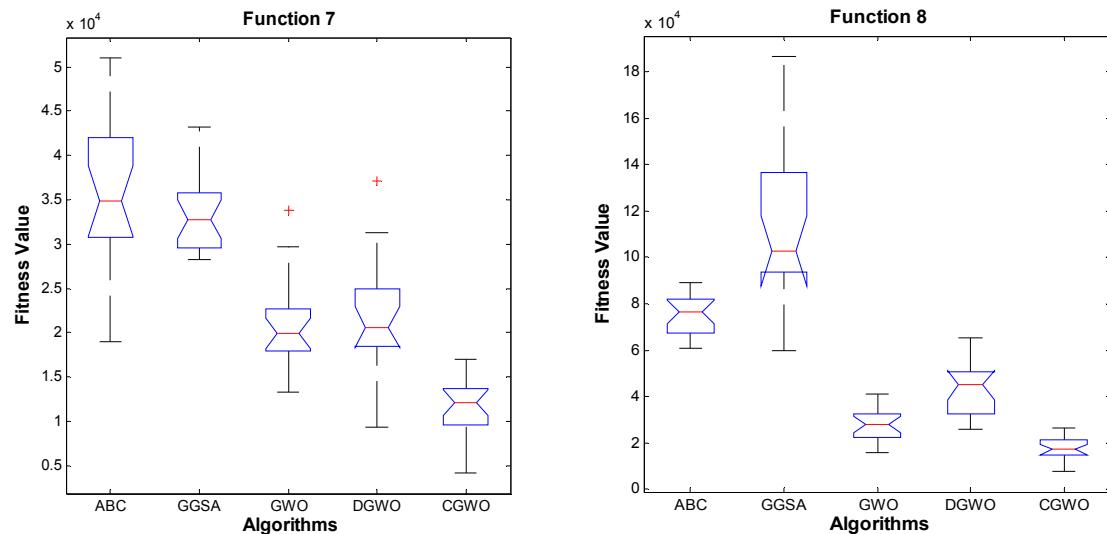


Figure 2. ANOVA tests of ABC, GGSA, GWO, DGWO, and CGWO on unimodal functions with Dim = 30.

4.4. Multimodal Benchmark Functions

The results of the multimodal benchmark functions are provided in Table 4. Note that multimodal functions have many local minima with the number increasing exponentially with dimension. Therefore, they are very suitable to test the exploration capability of algorithms to avoid local minima. As the results of the mean and STD show, CGWO performs better than the other algorithms on 30-D, 50-D multimodal benchmark functions. For F_9 on 200-D, 300-D, 400-D, the GWO has better value than the CGWO. For F_{12} on 10-D, the GGSA has best result. For F_{14} on 100-D, the GWO is slightly better than the CGWO. For F_{16} on 100-D, CGWO and GGSA have very close results. This table also shows that CGWO provides better results than the other algorithms across the majority of dimensions on 10-D, 100-D, 200-D, 300-D, and 400-D. Moreover, the p values of $F_9 \sim F_{16}$ with different dimensions reported in Tables 5–11 are less than 0.05 across the majority of dimensions, which is strong evidence against null hypothesis. So, this evidence demonstrates that the results of CGWO are statistically significant and do not occur by coincidence.

Figure 3 shows the averaged convergence curves of multimodal benchmark functions with 30-D over 20 independent runs. As shown in Figure 2, the convergence rate of CGWO in all cases is better than the other algorithms. Figure 4 depicts the anova tests between CGWO and the other algorithms on multimodal functions with 30-D. As shown, the more stable performance belongs to CGWO. According to Tables 4–11 and Figures 3 and 4 it can be stated that CGWO is capable of avoiding local minima with good convergence speed and stable property when dealing with multimodal benchmark functions.

According to this comprehensive comparative study and discussion, we state that this algorithm has merit compared to the other algorithms. The next section inspects the performance of the CGWO in solving the IIR model identification problem.

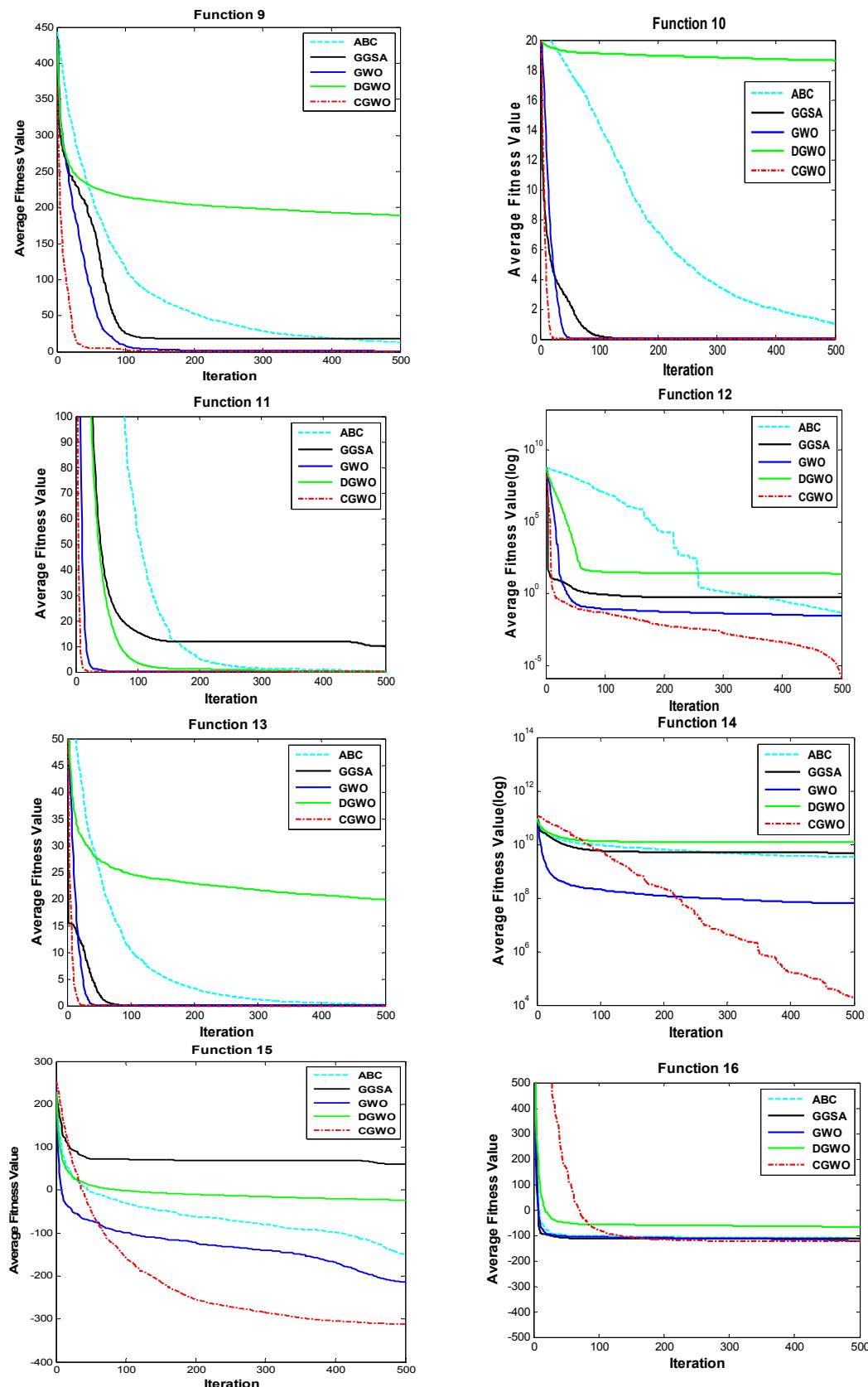


Figure 3. Convergence behavior of ABC, GGSA, GWO, DGWO, and CGWO on multimodal functions with Dim = 30.

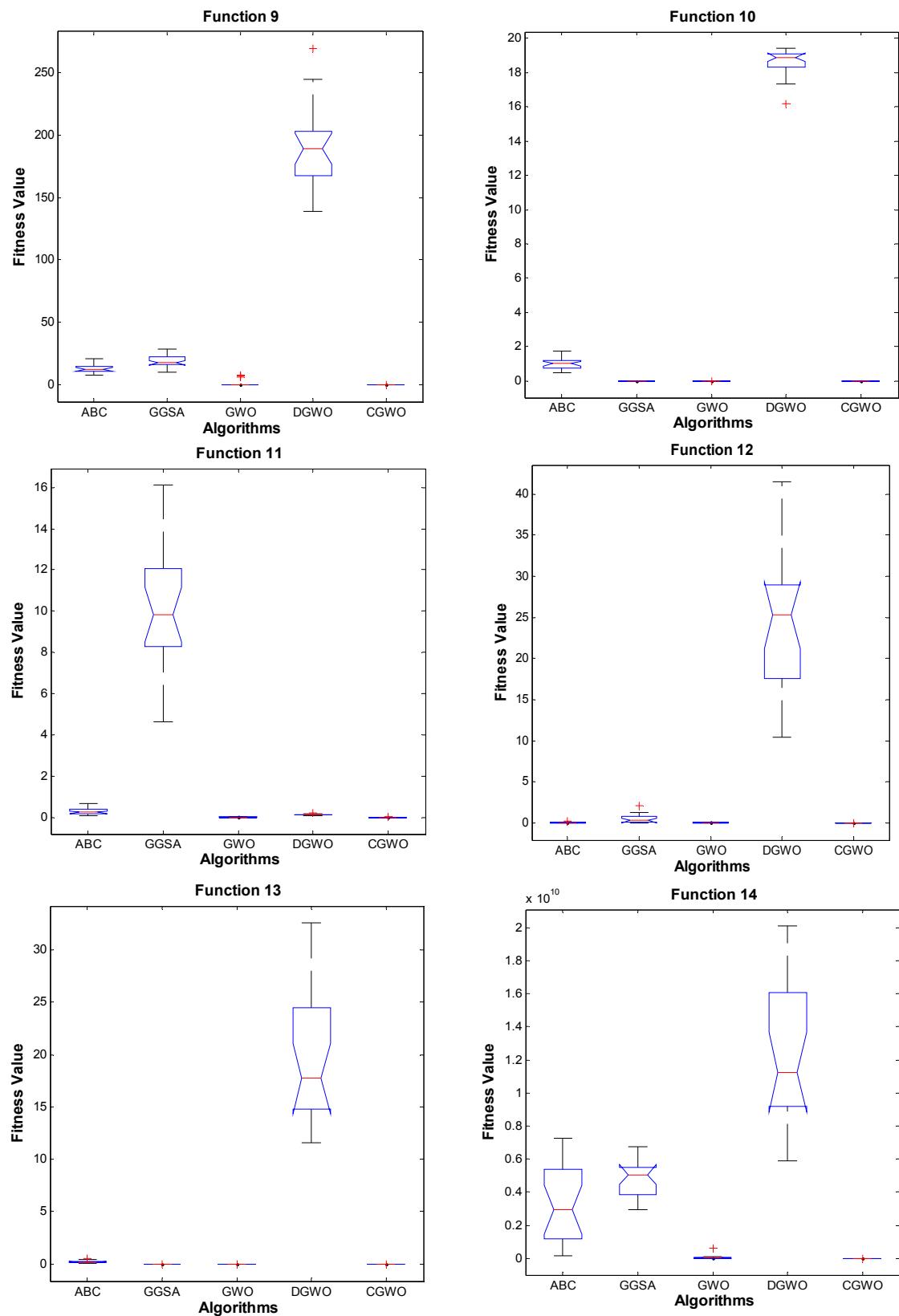


Figure 4. Cont

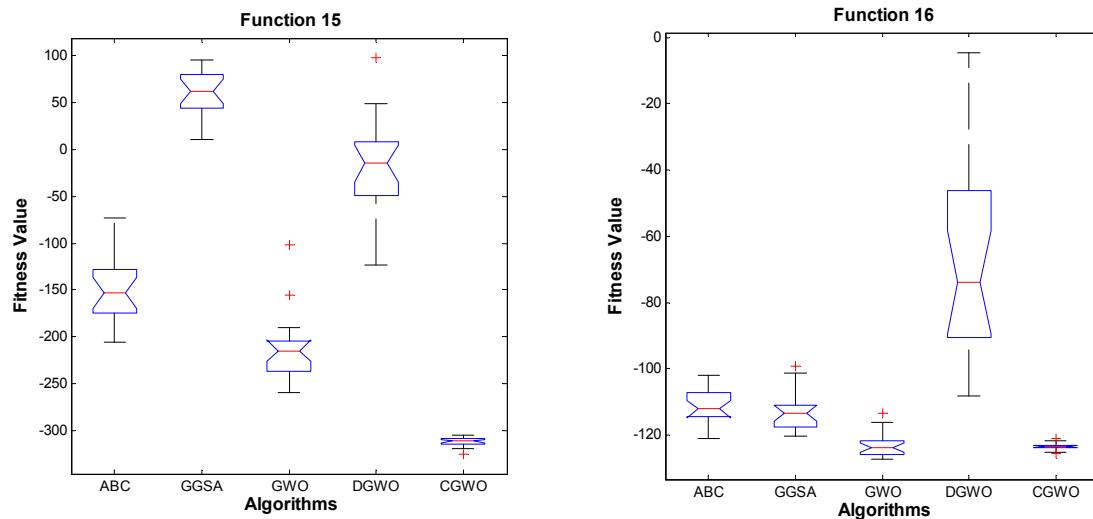


Figure 4. ANOVA tests of ABC, GGSA, GWO, DGWO, and CGWO on multimodal functions with Dim = 30.

5. IIR Model Identification

System identification is an effective method to obtain the mathematical model of an unknown system by analyzing its input and output values. In system identification, the use of infinite impulse response (IIR) has been widely studied since many problems encountered in signal processing can be characterized as a system identification problem (Figure 5). In this case, an optimization algorithm tries to iteratively adjust the parameters until the error between the output of the candidate model and the actual output of the plant is minimized. The use of IIR models for identification is preferred over their equivalent FIR (finite impulse response) models since the former produce more accurate models of physical plants for real world applications [32]. Moreover, IIR models are able to use fewer model parameters to meet the performance specifications. The fitness function, which is used in the article, is calculated as follows [33].

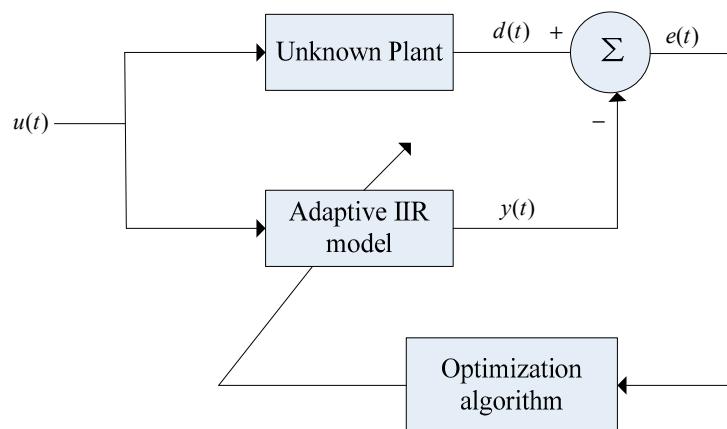


Figure 5. Block diagram of IIR system identification.

In an IIR system, the transfer function is provided as follows:

$$\frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_m z^{-m}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}} \quad (17)$$

where m and n are the number of numerator and denominator coefficients of the transfer function, respectively, a_i ($i \in [1, \dots, n]$) is the pole of the IIR model, b_j ($j \in [1, \dots, m]$) is the zero parameters of the IIR model. Equation (17) is expressed in another form as follows:

$$y(t) = \sum_{i=1}^n a_i \cdot y(t-i) + \sum_{j=0}^m b_j \cdot x(t-j) \quad (18)$$

where $u(t)$ indicates the t th input of the system, $y(t)$ indicates the t th output of the system. Hence, the set of unknown parameters that simulates the IIR system is indicated by $\theta = \{a_1, \dots, a_n, b_0, \dots, b_m\}$. The number of unknown parameters of θ is $(n + m + 1)$. Correspondingly, the search space S of IIR feasible values for θ is $\Re^{(n+m+1)}$.

According to Figure 5, the output of the plant is $d(t)$, while the output of the IIR filter is $y(t)$. The error $e(t)$ indicates the difference between the actual system and its model yields. Here the error is $e(t) = d(t) - y(t)$. Therefore, the problem of IIR model identification can be solved as a minimization problem of function. Additionally, the function can be molded as follows:

$$f(\theta) = \frac{1}{w} \sum_{t=1}^w (d(t) - y(t))^2 \quad (19)$$

where W indicates the number of the samples used in the simulation, and the aim is to minimize the function $f(\theta)$ through adjusting θ . When the error function $f(\theta)$ reaches its minimum value, the optimal model θ^* or solution is obtained. The θ^* is shown as follows:

$$\theta^* = \operatorname{argmin}(f(\theta)), \quad \theta \in S \quad (20)$$

IIR Model Identification Result

The result is reported considering a superior-order plant through a high-order IIR model. The unknown plant H_p and the IIR model H_M hold the following transfer function [27]:

$$H_p(z^{-1}) = \frac{1 - 0.4z^{-2} - 0.65z^{-4} + 0.26z^{-6}}{1 - 0.77z^{-2} - 0.8498z^{-4} + 0.6486z^{-6}} \quad (21)$$

$$H_M(z^{-1}) = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + b_3z^{-3} + b_4z^{-4}}{1 + a_1z^{-1} + a_2z^{-2} + a_3z^{-3} + a_4z^{-4}} \quad (22)$$

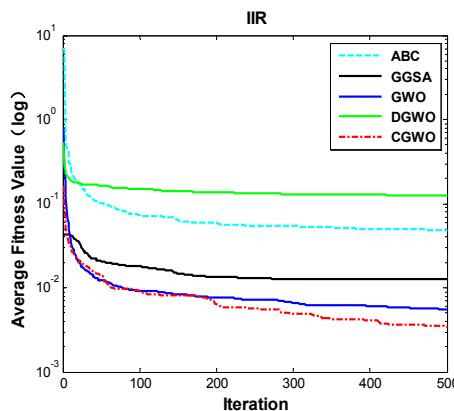
Since the plant is a sixth-order system and the IIR model a fourth-order system, the error surface $f(\theta)$ is multimodal. A white sequence with 100 samples has been used as input. The result of the experience over 20 consecutive experiments is shown in Tables 5 and 6. Table 13 shows the best parameter values (ABP). Table 14 indicates the average $f(\theta)$ value (AVE) and the standard deviation (STD). Figure 6 provides a comparison between CGWO and the other algorithms for IIR system identification.

Table 13. The best parameter values (ABP).

Algorithms	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9
CGWO	1.1093E−02	−4.9144E−01	−2.9101E−03	−3.3551E−01	9.8932E−01	1.3089E−02	−1.2463E−01	−1.5870E−02	−7.3745E−02
DGWO	1.2382E−01	5.5593E−02	−4.9268E−04	1.0155E−01	2.1844E−01	−1.6315E−01	−8.9793E−02	−1.1818E−01	6.1028E−02
GWO	−2.8883E−02	−2.6764E−01	−2.3851E−02	−2.4399E−01	9.1861E−01	−2.9241E−03	4.8354E−02	−2.7404E−02	4.9129E−02
GGSA	4.6698E−01	−1.1299E−01	5.7827E−02	1.8673E−01	−3.6432E−02	1.0900E−01	−8.4527E−02	−8.4070E−02	3.0088E−01
ABC	9.9479E−03	4.4119E−02	5.9096E−02	−2.2442E−01	8.3829E−01	−1.1020E−01	2.8357E−01	−4.1144E−02	1.1184E−01

Table 14. The average $f(\theta)$ value (AVE) and the standard deviation (STD).

Algorithms	AVE	STD
CGWO	3.4013E−03	2.1644E−03
DGWO	1.2513E−01	7.1059E−02
GWO	5.5317E−03	4.9962E−03
GGSA	1.2537E−02	2.4702E−03
ABC	4.7898E−02	1.2004E−02

**Figure 6.** Comparison between CGWO and the other algorithms for IIR system identification.

According to the AVE and STD indexes in Table 6, the CGWO algorithm finds better results than the other algorithms. The results show that CGWO provides better precision (AVE value) and robustness (STD). As shown in Figure 6, the CGWO has the fastest convergence rate.

In conclusion, the results show that the proposed method successfully outperforms the other algorithms in a majority of benchmark functions. In addition, IIR model identification shows that CGWO is able to obtain very competitive results. Therefore, as shown in the results of this comparative study, the proposed method has merit in the field of evolutionary algorithms and optimization.

6. Conclusions

In this work, the idea of complex-valued encoding is introduced into GWO, and a complex-valued version of GWO called CGWO was proposed based on complex-valued encoding. The proposed functions justified its excellent property on 16 benchmark functions in terms of improved avoidance of local minima and increased convergence speed. Furthermore, a complex system identification problem been employed to test the performance of the proposed method. The results verified the superior performance of CGWO on IIR model identification when compared to the other algorithms because of its improved ability. For future studies, it is recommended to apply the CGWO to solve more real-world engineering problems.

Acknowledgments: This work is supported by National Science Foundation of China under Grants No. 61463007; 6153008. Project of Guangxi University for Nationalities Science Foundation under Grant No. 2012MDZD037.

Author Contributions: Qifang Luo designed algorithm; Sen Zhang and Zhiming Li performed the experiments; Yongquan Zhou wrote the paper.

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

References

- Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, WA, USA, 27 November–1 December 1995; pp. 1942–1948.
- Holland, J.H. Genetic algorithms. *Sci. Am.* **1992**, *267*, 66–72. [[CrossRef](#)]
- Price, K.; Storn, R. Differential evolution. *Dr. Dobb's J.* **1997**, *22*, 18–20.
- Price, K.V.; Storn, R.M.; Lampinen, J.A. *Differential Evolution: A Practical Approach to Global Optimization*; Springer: New York, NY, USA, 2005.
- Kirkpatrick, S.; Gelati, C.D.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [[CrossRef](#)] [[PubMed](#)]
- Dorigo, M.; Maniezzo, V.; Colomi, A. The ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. B* **1996**, *26*, 29–41. [[CrossRef](#)] [[PubMed](#)]
- 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](#)]
- Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. GSA: A gravitational search algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [[CrossRef](#)]
- Mirjalili, S.; Lewis, A. Adaptive gbest-guided gravitational search algorithm. *Neural Comput. Appl.* **2014**, *25*, 1569–1584. [[CrossRef](#)]
- Kaveh, A.; Talatahari, S. A novel heuristic optimization method: Charged system search. *Acta Mech.* **2010**, *213*, 267–289. [[CrossRef](#)]
- Tayarani, N.M.H.; Akbarzadeh, T.M.R. Magnetic optimization algorithms a new synthesis. In Proceedings of the IEEE Congress on Evolutionary Computation, Hong Kong, China, 1–6 June 2008; pp. 2659–2664.
- Simon, D. Biogeography-based optimization. *IEEE Trans. Evol. Comput.* **2008**, *12*, 702–713. [[CrossRef](#)]
- Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Let a biogeographybased optimizer train your multi-layer perceptron. *Inf. Sci.* **2014**, *269*, 188–209. [[CrossRef](#)]
- Rao, R.V.; Savsani, V.J.; Vakharia, D.P. Teaching–learning-based optimization: A novelmethod for constrained mechanical design optimization problems. *Comput.-Aided Des.* **2011**, *43*, 303–315. [[CrossRef](#)]

15. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [[CrossRef](#)]
16. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
17. Sulaiman, M.H.; Mustaffa, Z.; Mohamed, M.R.; Aliman, O. Using the gray wolf optimizer for solving optimal reactive power dispatch problem. *Appl. Soft Comput.* **2015**, *32*, 286–292. [[CrossRef](#)]
18. Song, X.H.; Tang, L.; Zhao, S.T.; Zhang, X.Q.; Li, L.; Huang, J.Q.; Cai, W. Grey Wolf Optimizer for parameter estimation in surface waves. *Soil Dyn. Earthq. Eng.* **2015**, *75*, 147–157. [[CrossRef](#)]
19. Komaki, G.M.; Kayvanfar, V. Grey Wolf Optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time. *J. Comput. Sci.* **2015**, *8*, 109–120. [[CrossRef](#)]
20. Casasent, D.; Natarajan, S. A classifier neural network with complex-valued weights and square-law nonlinearities. *Neural Netw.* **1995**, *8*, 989–998. [[CrossRef](#)]
21. Chen, D.-B.; Li, H.-J.; Li, Z. Particle swarm optimization based on complex-valued encoding and application in function optimization. *Comput. Eng. Appl.* **2009**, *45*, 59–61.
22. Das, S.; Suganthan, P.N. Differential evolution: A survey of the state-of-the-art. *IEEE Trans. Evol. Comput.* **2011**, *15*, 4–31. [[CrossRef](#)]
23. Yang, X.S. Appendix a: Test problems in optimization. In *Engineering Optimization*; Yang, X.S., Ed.; John: Hoboken, NJ, USA, 2010; pp. 261–266.
24. Tang, K.; Yao, X.; Suganthan, P.N.; MacNish, C.; Chen, Y.P.; Chen, C.M.; Yang, Z. *Benchmark Functions for the CEC'2008 Special Session and Competition on Large Scale Global Optimization*; Technical Report; Nature Inspired Computation and Applications Laboratory, University of Science and Technology of China: Hefei, China, 2007; pp. 153–177.
25. Suganthan, P.N.; Hansen, N.; Liang, J.J.; Deb, K.; Chen, Y.; Auger, A.; Tiwari, S. *Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization*; Technical Report; Nanyang Technological University: Singapore, 2005; Volume 2005005.
26. The Matlab Code of the GGSA Algorithm. Available online: <http://www.alimirjalili.com/Projects.html> (accessed on 10 October 2014).
27. Derrac, J.; García, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18. [[CrossRef](#)]
28. White, M.S.; Flockton, S.J. Adaptive recursive filtering using evolutionary algorithms. In *Evolutionary Algorithms in Engineering Applications*; Dasgupta, D., Michalewicz, Z., Eds.; Springer: Berlin, Germany, 1997; pp. 361–376.
29. García, S.; Molina, D.; Lozano, M.; Herrera, F. A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization. *J. Heuristics* **2009**, *15*, 617–644. [[CrossRef](#)]
30. Wilcoxon, F. *Individual Comparisons by Ranking Methods*; Biometrics Bulletin; International Biometric Society: Washington, DC, USA, 1945; pp. 80–83.
31. Alcalá-Fdez, J.; Sánchez, L.; García, S.; del Jesus, M.J.; Ventura, S.; Garrell, J.M.; Otero, J.; Romero, C.; Bacardit, J.; Rivas, V.M.; et al. KEEL: A Software Tool to Assess Evolutionary Algorithms to Data Mining Problems. *Soft Comput.* **2009**, *13*, 307–318. [[CrossRef](#)]
32. Kukrer, O. Analysis of the dynamics of a memoryless nonlinear gradient IIR adaptive notch filter. *Signal Process.* **2011**, *91*, 2379–2394. [[CrossRef](#)]
33. Cuevas, E.; Gálvez, J.; Hinojosa, S.; Avalos, O.; Zaldívar, D.; Pérez-Cisneros, M. A Comparison of Evolutionary Computation Techniques for IIR Model Identification. *J. Appl. Math.* **2014**, *2014*, 768516. [[CrossRef](#)]

