Hindawi
Journal of Optimization
Volume 2018, Article ID 6495362, 18 pages
https://doi.org/10.1155/2018/6495362

Research Article

A New Hybridization Approach between the Fireworks Algorithm and Grey Wolf Optimizer Algorithm

Juan Barraza, Luis Rodríguez, Oscar Castillo, Patricia Melin, and Fevrier Valdez

Tijuana Institute of Technology, Calzada Tecnologico s/n, 22379 Tijuana, BC, Mexico

Correspondence should be addressed to Patricia Melin; pmelin@tectijuana.mx

Received 30 January 2018; Accepted 3 April 2018; Published 27 May 2018

Academic Editor: S. Mehdi Vahidipour

The main aim of this paper is to present a new hybridization approach for combining two powerful metaheuristics, one inspired by physics and the other one based on bioinspired phenomena. The first metaheuristic is based on physics laws and imitates the explosion of the fireworks and is called Fireworks Algorithm; the second metaheuristic is based on the behavior of the grey wolf and belongs to swarm intelligence methods, and this method is called the Grey Wolf Optimizer algorithm. For this work we studied and analyzed the advantages of the two methods and we propose to enhance the weakness of both methods, respectively, with the goal of obtaining a new hybridization between the Fireworks Algorithm (FWA) and the Grey Wolf Optimizer (GWO), which is denoted as FWA-GWO, and that is presented in more detail in this work. In addition, we are presenting simulation results on a set of problems that were tested in this paper with three different metaheuristics (FWA, GWO, and FWA-GWO) and these problems form a set of 22 benchmark functions in total. Finally, a statistical study with the goal of comparing the three different algorithms through a hypothesis test (Z-test) is presented for supporting the conclusions of this work.

1. Introduction

In recent years, the global world of computer science is creating an interesting environment [1] for research, especially, with algorithms that aim at solving different optimization problems [2], which means maximizing or minimizing depending on the goal and the requirements of the particular problem [3].

At present time, a family of recent algorithms having great impact is the so-called bioinspired algorithms [4]; the reason is because they offer an interesting way of simulating the combination among the natural phenomena, mathematics, and computation. Combination is a key word among these algorithms; for example, in genetic algorithms [5], a combination is performed with two operators, such as crossover and mutation.

Keeping on with the combination concept, another related concept is hybridization [6], and we can understand, for hybridization, processes in which discrete structures, which can exist separately, are combined to generate new structures, objects, and methods.

In recent works, it has been demonstrated that the hybridization of bioinspired algorithms has yielded very good results, and in this regard the main contribution in this work is the proposed hybridization between the Fireworks Algorithm (FWA) [7] and the Grey Wolf Optimizer (GWO) [8] taking advantage of their best features and combining them for obtaining a better overall performance for solving problems, in a new hybrid algorithm. The above-mentioned hybridization is described and explained in more detail in the following sections of this paper.

The remainder of the paper is organized as follows: we start with basic concepts in Section 2, which is organized in two parts, FWA in the first part and GWO in the second part. In Section 3, we present the proposed FWA-GWO method, then the simulation results are presented in Section 4, and finally, we conclude the paper, by mentioning possible future work in Section 5.

2. Basic Concepts

In this section, the basic concepts about FWA and GWO are presented to provide the reader with a background for understanding the proposed hybrid approach (FWA-GWO).

2.1. Fireworks Algorithm (FWA). The FWA is a metaheuristic method based on the explosion of fireworks behavior [9]. Every firework performs an explosion process generating a number of sparks, which are placed on a local space around the corresponding firework to represent possible solutions in a search space [10, 11].

2.1.1. Number of Sparks. Equations (1) are used to calculate the number of sparks [12, 13].

$$\text{Minimize} \quad \mathbf{f}(\mathbf{x_i}) \in \mathbf{R}, \mathbf{x_i}\_\text{min} \le \mathbf{x_i} \le \mathbf{x_i}\_\text{max}$$
$$\mathbf{s_i} = \mathbf{m}\frac{\mathbf{y}_{\max} - \mathbf{f}(\mathbf{x_i}) + \boldsymbol{\varepsilon}}{\sum_{\mathbf{i=1}}^{\mathbf{n}} (\mathbf{y}_{\max} - \mathbf{f}(\mathbf{x_i})) + \boldsymbol{\varepsilon}}$$
$$\widehat{\mathbf{S_i}} = \begin{cases} \text{round}(\mathbf{a}\cdot\mathbf{m}) & \text{if } \mathbf{S_i} < \mathbf{am} \\ \text{round}(\mathbf{b}\cdot\mathbf{m}) & \text{if } \mathbf{S_i} > \mathbf{bm},\ \mathbf{a} < \mathbf{b} < \mathbf{1}, \\ \text{round}(\mathbf{S_i}) & \text{otherwise.} \end{cases} \tag{1}$$

2.1.2. Explosion Amplitude. The explosion amplitude [14] is calculated for each firework by the following equation:

$$A_i = \widehat{A} \cdot \frac{f(x_i) - y_{\min} + \epsilon}{\sum_{i=1}^{n} (f(x_i) - y_{\min}) + \epsilon}. \tag{2}$$

2.1.3. Generating Sparks. Random dimensions are calculated with the following equation knowing that when a firework explodes, its sparks will take different random directions.

$$\mathbf{z} = \text{round}(\mathbf{d} \cdot \text{rand}(\mathbf{0}, \mathbf{1})). \tag{3}$$

2.1.4. Selection of Locations. After the explosion of the fireworks, to maintain the diversity of the sparks, $n-1$ places are selected with respect to the location of the others. To calculate the distance among $\mathbf{x_i}$ and the other places, the following equations are used:

$$\mathbf{R}(\mathbf{x_i}) = \sum_{\mathbf{j}\in\mathbf{K}} \mathbf{d}(\mathbf{x_i}, \mathbf{x_j}) = \sum_{\mathbf{j}\in\mathbf{K}} \left\| \mathbf{x_i} - \mathbf{x_j} \right\|. \tag{4}$$

$$\mathbf{p}(\mathbf{x_i}) = \frac{\mathbf{R}(\mathbf{x_i})}{\sum_{\mathbf{j}\in\mathbf{K}} \mathbf{R}(\mathbf{x_j})} \tag{5}$$

Figure 1 shows the general flowchart of the FWA, where the original author of the method [7] described each equation that was presented above and that we can find in more detail in the original paper [11] and other variants presented in [15–17].

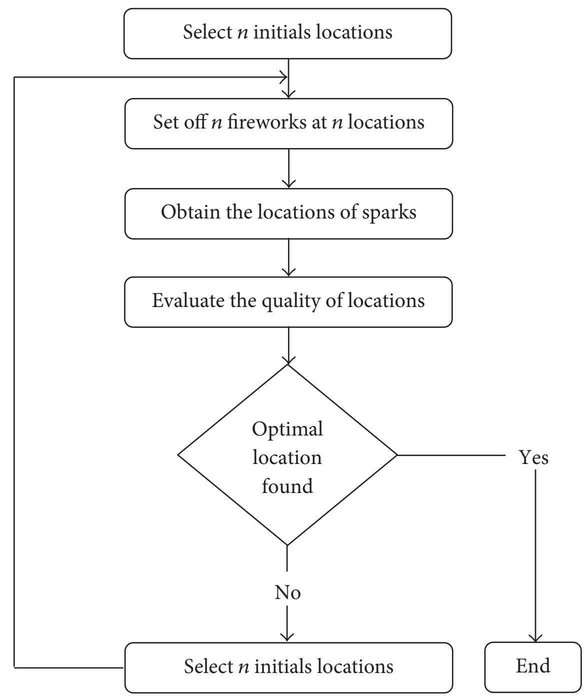

Figure 1: General flowchart of the FWA.

2.2. Grey Wolf Optimizer. The Grey Wolf Optimizer (GWO) algorithm [18] is a metaheuristic created by Seyedali Mirjalili in 2014. In this algorithm the author takes advantage of the main features that the grey wolf has in nature based on the research of Muro et al. [19] about hunting strategies of the Canis lupus or grey wolf. When the original author designed the algorithm, the following features were highlighted: the hierarchy into the pack [20] and the hunting mechanism.

Basically, the optimization process is guided by the three best solutions in the pack or in all the population, so in order to recognize these better solutions, we assume that the best solution is called the alpha (α) wolf, then the second and third best solutions in the optimization are called beta (β) and delta (δ) wolves, respectively, and finally the rest of the candidate solutions are called omega (ω) wolves.

In addition it is important to mention that the hunting mechanism behavior of the grey wolf and its main phases are described as follows: first, pursue and approach the prey, after that encircle and harass the prey until it stops moving, and finally attack the prey. In the algorithm this behavior is simulated by

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_p(t) - \vec{X}(t) \right|. \tag{6}$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A}\vec{D} \tag{7}$$

In this case, (6) represents a distance between the best solution ($\vec{X}_p(t)$) and a random motion ($\vec{C}$) that is represented in (9) and is a random value in a range $[0, 2]$ minus the solution that we have evaluated ($\vec{X}(t)$).

The next position of the current solution is presented in (7) that is a subtraction of the best solution and the distance that we have obtained in (6) multiplied by a weight (A), which is assigned to this distance and this is described in

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}. \tag{8}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \tag{9}$$

A and C coefficients represent the ways for the exploration and exploitation in the algorithm to occur [21], both in direct and in indirect ways, and we can find extensive information of these coefficients in the original paper, where the method was originally proposed [22].

Equations (10) and (11) are the same as (6) and (7), respectively, but in this case the best solution is represented by the leaders of the pack and in this case alpha, beta, and delta, as we mentioned above.

$$\begin{aligned}\vec{D}_\alpha &= \left|\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}\right|,\\ \vec{D}_\beta &= \left|\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}\right|,\\ \vec{D}_\delta &= \left|\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}\right|,\end{aligned} \tag{10}$$

$$\begin{aligned}\vec{X}_1 &= \vec{X}_\alpha - \vec{A}_1 \cdot \left(\vec{D}_\alpha\right),\\ \vec{X}_2 &= \vec{X}_\beta - \vec{A}_2 \cdot \left(\vec{D}_\beta\right),\\ \vec{X}_3 &= \vec{X}_\delta - \vec{A}_3 \cdot \left(\vec{D}_\delta\right),\end{aligned} \tag{11}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}. \tag{12}$$

Finally, in (12) we can find the next position of the solution that we are evaluating, which is basically an average based on the three best wolves in the pack, so in these equations that we described above we can find the main inspiration for the GWO algorithm, which is the hierarchical pyramid of leadership and the hunting mechanism. In Figure 2 we can find the flow chart of the algorithm.

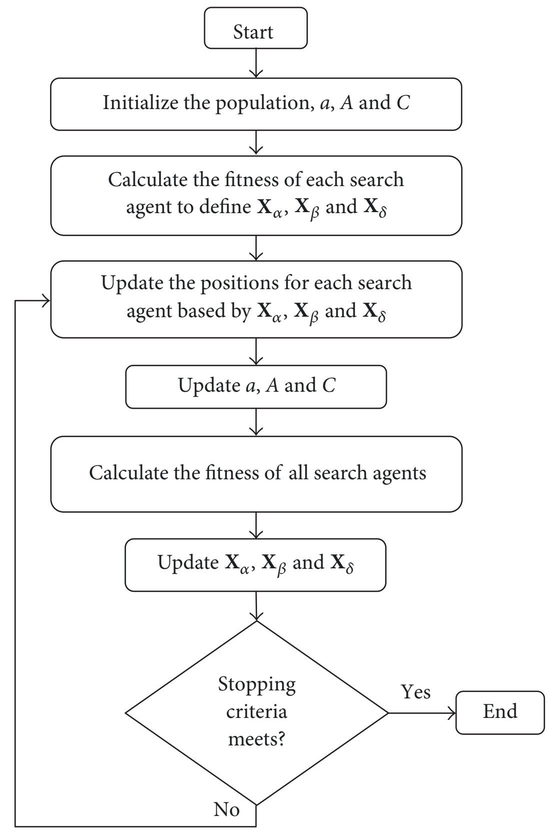

Figure 2: General flowchart of the GWO.

3. Proposed FWA-GWO

The main goal of the hybridization between both methods [6] is to take advantage of their main features and in this paper we present the following features of each method that we are using for achieving hybridization:

FWA:

(i) Initialization
(ii) Explosion amplitude
(iii) Locations

GWO:

(i) Hierarchical pyramid
(ii) Interaction among the population
(iii) Next location of population

Figure 3 shows a general flow chart of the proposed hybridization between the FWA and GWO algorithms and the red blocks represent the considered features of the FWA and the blue blocks the GWO features that are used in the proposed hybrid approach.

In order to explain the performance of the FWA-GWO, we are presenting below the details of each block in the flowchart that we illustrated in Figure 3.

3.1. Select n Initial Packs and m Wolves. We start to select a number n of locations as in the FWA, but in this case we are representing the number of the packs in the population. In addition we need to select the number of wolves for each pack represented by m. It is important to mention that the conventional FWA works with a number of function evaluations and the GWO works with the iterations as stopping criteria, so in (13) we show the relation between both types of stopping criteria.

$$T = \frac{O}{n * m}, \tag{13}$$

where T is the total number of iterations; O is the number of function evaluations; n is the number of packs; and m is the number of wolves for each pack.

For example if we have 4 packs and 6 wolves for each pack, we have 24 possible solutions for each iteration in the algorithm; therefore, for 15,000 function evaluations the FWA-GWO algorithm will execute 625 iterations [23].

3.2. Set n Initial Packs with m Wolves for the Search Space. We add a way to select n initial locations in this algorithm, which means that, depending on the number of packs, the corresponding number of partitions is found. For example, if we have a search space with a lower bound and upper bound defined in a range $[-100, 100]$,

$$\mathrm{TS} = \mathrm{ub} - \mathrm{lb}, \tag{14}$$

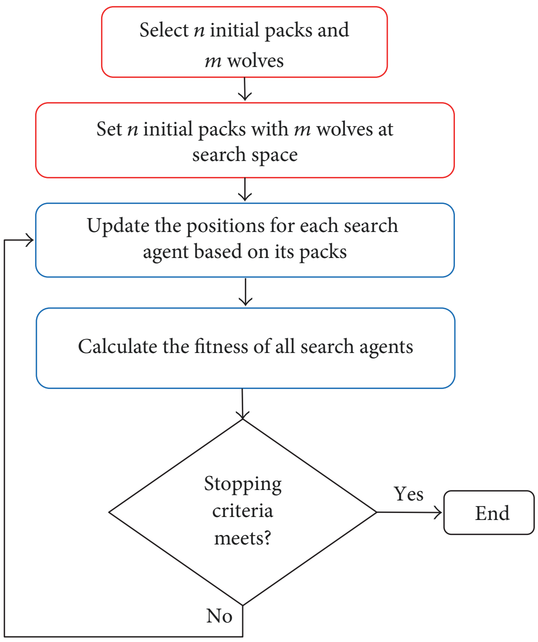

Figure 3: General flowchart of the FWA-GWO.

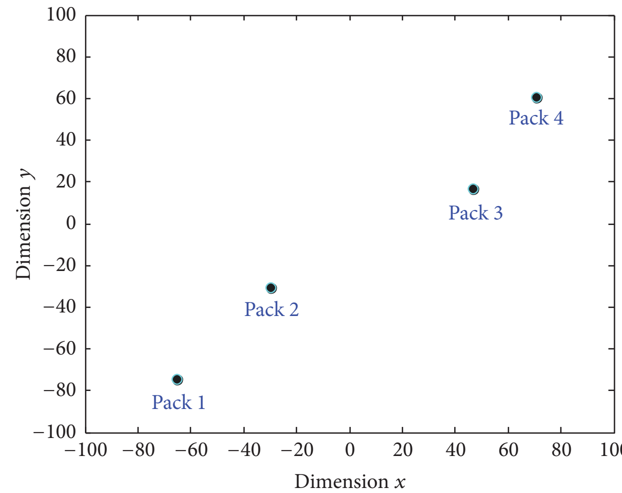

Figure 4: Example of initialization for each pack in FWA-GWO.

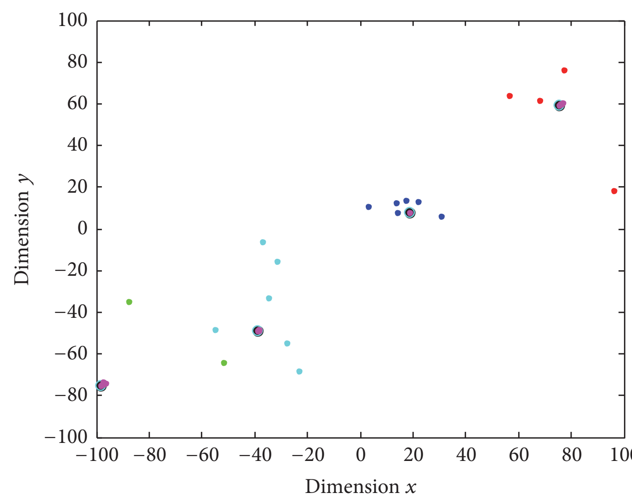

Figure 5: Initialization of the population in FWA-GWO.

where TS is a total search space and ub and lb represent the upper and lower bounds, respectively.

We need subspaces where each pack performs the initialization of its subpopulation and this behavior is represented in

$$SS = \frac{TS}{n}. \quad (15)$$

SS is the distance that should exist between each range for each pack. These ranges for each pack are calculated with a general linear function, as shown as follows:

$$f(n) = an + b, \quad (16)$$

where $a = SS$, n is the number of packs, and b is a calculated constant value. In this case, $f(1) = \text{lb}$ of the search space, such that the search ranges for each pack are $[f(1), f(2)]$, $[f(2), f(3)] \cdots [f(n), f(n+1)]$. We can represent this as follows: $f(n+1) = \text{ub}$.

In Figure 4, we can find an example of this initialization method with 4 packs and a search space in a range of of $[-100, 100]$ with 2 dimensions.

An advantage of this algorithm is the initialization of the population in subspaces of the search space; FWA-GWO partitions the search space for the function to be evaluated based on the number of packs, with the aim of covering most of the search space and assuring achieving the exploration. Figure 5 shows the initialization of the population based on the leader.

3.3. Update the Positions for Each Agent Based on Its Packs. As we mentioned above, this part simulates the update of the wolves based on the leaders of each pack, and to understand better its performance, we illustrate in a graphical way a representation of this behavior.

Figure 6 shows an example of how to update the positions in FWA-GWO; is a bidimensional plot with a function in a range of $[-100, 100]$ and with an optimum equal to zero. Also in Figure 6, we can find in a graphical way the convergence of the all search agents in the algorithm. In addition, we can mention that blue, red, aqua, and green points represent the members of the 4 packs for this example, and the magenta points represent the initialization of each pack.

On the other hand, to keep the equilibrium of this algorithm (FWA-GWO), we decided to partition the search process in 3 phases; 100 percent of the total number of iterations was divided into 3 parts. We consider the first phase in an interval of 0% to 33.3% of the iterations and we consider this as the exploration phase, the second phase takes an interval from 33.3% to 66.6%, in this phase sometimes the FWA-GWO algorithm is exploring and other times exploiting, and finally, we consider the third phase as the exploitation and take the remaining part of the percentage (66.6% to 99.9%) of the total number of iterations.

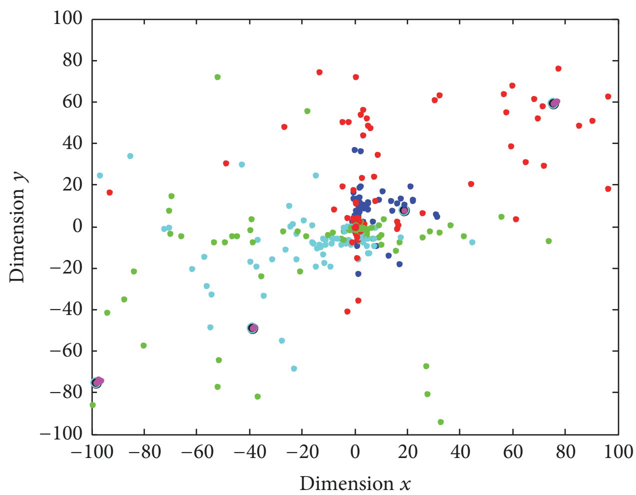

Figure 6: Convergence of search agents in FWA-GWO.

Table 1: Relationship between phases and number of packs in the FWA-GWO.

Phase	Number of packs						
1	2	3	4	5	6	7	8
2	1	1	2	2	3	3	4
3	1	1	1	1	1	1	1

It is worth mentioning that the number of packs is not constant, while the FWA-GWO is working. In other words, the objective of having three phases in this algorithm is to reduce the number of packs according to the particular phase in this algorithm. In addition, in the first phase (exploration) the algorithm will work with the number of packs and wolves that we selected at the beginning; in the second phase, the numbers of packs is reduced at half with the help of (17) and finally the FWA-GWO algorithm always finishes the optimization with only one big pack.

$$\underline{n} = \frac{n}{2}. \tag{17}$$

Table 1 shows examples of the relationship between the phases and number of packs, respectively, based on the number of packs that were initialized.

In addition, we are presenting this information (Table 1) in a graphical way to explain the relationship in more detail. Figure 7 represents phase 1 where $n = 4$ and $m = 6$.

In Figure 8 we present the second phase, so the new value of $n = 2$ is based on (17) and the new value for m is 12, because we divided the population into the number of packs (n). In addition we can mention that the total population is the same as that was originally initialized.

Finally, in Figure 9 we can find only one big pack as we mentioned above. In addition we can mention that this is an example of how the population is distributed into the 3 phases that the algorithm has and finally we assume that each pack has their corresponding leaders (alpha, beta, and delta).

3.4. Mathematical Model of FWA-GWO. For the hybridization between FWA and GWO, we modified the equations in GWO to calculate the distance for updating the next position of the current solution, and we introduced the amplitude explosion of the FWA into the distance for updating the next position in GWO (coefficient C in (6)). Equation (18) shows the changes that we mentioned above.

$$\mathbf{A_n} = \begin{cases} \mathbf{n = 1}, & \mathbf{A_1 = 0.5} \\ \mathbf{n = 2}, & \mathbf{A_1 = 1},\ \mathbf{A_2 = 2} \\ \mathbf{n \geq 3}, & \mathbf{A_n} = \widehat{\mathbf{A}} \cdot \dfrac{\mathbf{f(x_n) - y_{min}} + \epsilon}{\sum_{\mathbf{i=1}}^{\mathbf{n}} \left(\mathbf{f(x_n) - y_{min}}\right) + \epsilon}, \end{cases} \tag{18}$$

where $\mathbf{A_n}$ represents the amplitude explosions of each pack and $\mathbf{n}$ is the number of packs. When the number packs is 1 then the amplitude is 0.5, when the number of packs is 2 then the amplitude of the packs is 1 and 2, respectively [8]. When the number of packs is equal or greater than 3, we can then use the formula of the amplitude explosion of FWA.

The reason why the parameters are 0.5, 1, and 2 when the number of packs is one or two is because we are trying to maintain the parameter values in a range of $[0, 2]$, as the parameter C in (6) in GWO is the one that controls the exploration and exploitation in the algorithm.

Equation (19) shows how we can normalize the parameter of the explosion amplitude between 0 and 2 when the number of packs is greater than 2.

$$\widehat{A_n} = \frac{2 * A_n}{\max(A_n)}, \tag{19}$$

where $\widehat{A_n}$ represents the explosion amplitude normalized for each pack and $\max(A_n)$ is the maximum value of all amplitudes.

The distance between the best wolf with a random motion and the current wolf of each pack is calculated in the following way:

$$\overrightarrow{D_n} = \left| \widehat{A_n} \cdot \overrightarrow{X_{pn}}(t) - \overrightarrow{X_n}(t) \right|, \tag{20}$$

where $\overrightarrow{D_n}$ is the obtained distance of each pack, $\overrightarrow{X_{pn}}(t)$ is the leader of the pack, and $\overrightarrow{X_n}(t)$ is the current wolf of each pack.

The weight for each distance of the omega wolves and the leaders of the pack that we described above is defined as follows:

$$\overrightarrow{E_n} = 2a \cdot \overrightarrow{r_{1n}}. \tag{21}$$

In (21), $\overrightarrow{E_n}$ represents the weight for (22), a is a value that decreases during iterations in a range of $[2, 0]$, and $\overrightarrow{r_{1n}}$ is a random value between 0 and 1 for each pack.

Equation (22) allows updating the next position of the current wolf and the mathematical model is defined as follows:

$$\overrightarrow{X_n}(t+1) = \overrightarrow{X_{pn}}(t) - \overrightarrow{E_n}\overrightarrow{D_n}, \tag{22}$$

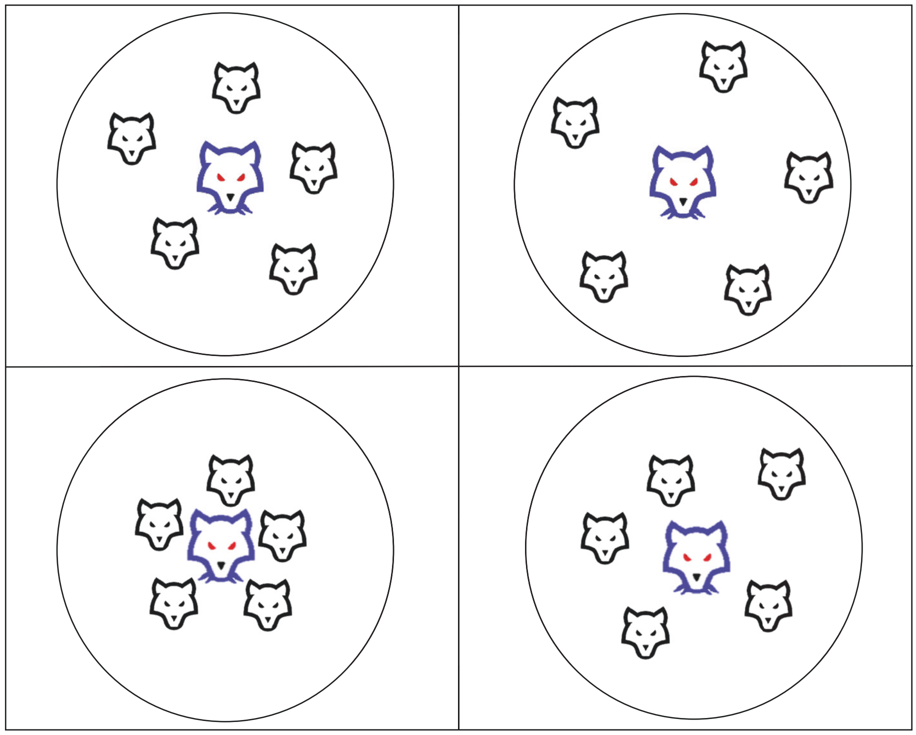

Figure 7: Example of phase 1 in FWA-GWO.

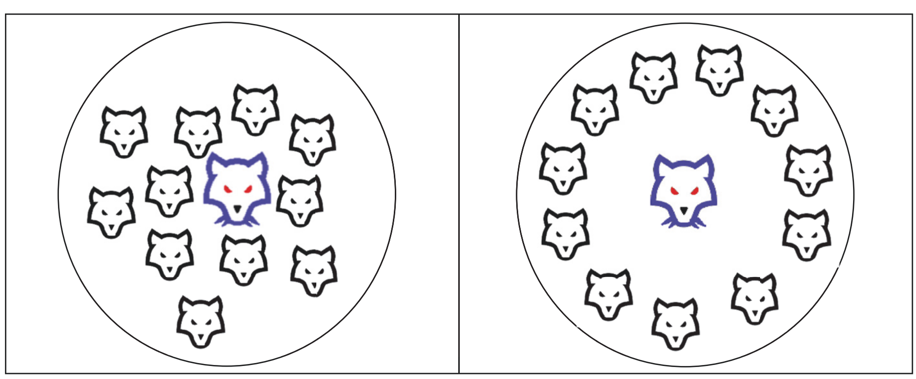

Figure 8: Example of phase 2 in FWA-GWO.

where $\overrightarrow{X_n}(t+1)$ represents the next position, $\overrightarrow{X_{pn}}(t)$ is the best wolf of each pack, and $\overrightarrow{D_n}$ and $\overrightarrow{E_n}$ are described in (20) and (21), respectively.

To apply randomness in the method, (23) is used with the aim that each leader obtains different moves for each omega wolf.

$$\overrightarrow{C_n} = \widehat{A}_n \cdot \overrightarrow{r_{2n}}. \quad (23)$$

Equations (24) and (25) are the same as (10) and (11) in GWO, but the difference is that now the leaders are represented for each pack.

$$\overrightarrow{D_{\alpha n}} = \left\| \overrightarrow{C_{1n}} \cdot \overrightarrow{X_{\alpha n}} - \overrightarrow{X_n} \right\|,$$
$$\overrightarrow{D_\beta} = \left\| \overrightarrow{C_{2n}} \cdot \overrightarrow{X_{\beta n}} - \overrightarrow{X_n} \right\|, \quad (24)$$
$$\overrightarrow{D_\delta} = \left\| \overrightarrow{C_{3n}} \cdot \overrightarrow{X_{\delta n}} - \overrightarrow{X_n} \right\|,$$

$$\overrightarrow{X_{1n}} = \overrightarrow{X_{\alpha n}} - \overrightarrow{E_{1n}} \cdot \left(\overrightarrow{D_{\alpha n}}\right),$$
$$\overrightarrow{X_{2n}} = \overrightarrow{X_{\beta n}} - \overrightarrow{E_{2n}} \cdot \left(\overrightarrow{D_{\beta n}}\right), \quad (25)$$
$$\overrightarrow{X_{3n}} = \overrightarrow{X_{\delta n}} - \overrightarrow{E_{3n}} \cdot \left(\overrightarrow{D_{\delta n}}\right),$$

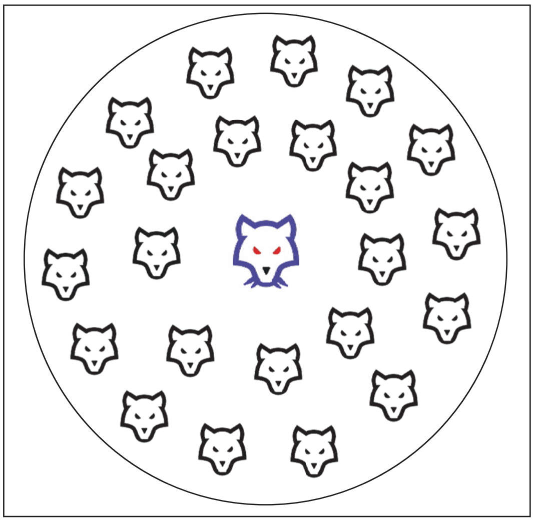

Figure 9: Example of phase 3 in FWA-GWO.

```
Initialize n and m
Initialize the grey wolf population X_ni (i = 1, 2, ..., n)
Initialize a, E_n and Â_n
Calculate the fitness of each search agent
X_nα = the best search agent of n pack
X_nβ = the second best agent of n pack
X_nδ = the third best search agent of n pack
while (t < Max number of iterations)
        for each search agent
                Update the position of the current search
                 agent by  Equation (5)
        end for
        Update a, Â_n and E_n
        Update n and m
        Calculate the fitness of all search agents
        Update X_nα, X_nβ and X_nδ
        t = t + 1
end while
return best (X_1α, X_2α, ..., X_nα)
```

Algorithm 1: Pseudocode for the FWA-GWO.

$$\overrightarrow{X_n}(t+1) = \frac{\overrightarrow{X_{1n}} + \overrightarrow{X_{2n}} + \overrightarrow{X_{3n}}}{3}. \quad (26)$$

In (26) we can find the next position of the solution that we are evaluating as the GWO works and is an average of the three best wolves, but in this case this is represented for each pack. Finally, Algorithm 1 presents the pseudocode for the FWA-GWO.

4. Simulation Results and Discussion

In this section we are presenting the benchmark functions that are used in this work.

Table 2 shows the equations of the first 13 benchmark functions [24–26] used for the tests with the algorithms (FWA, FWO, and FWA-GWO) that can be classified as unimodal and multimodal, respectively. Figure 10 shows the graphical representations of these benchmark functions in their 3D versions.

Also in this paper we used another set of 9 benchmark functions that are called fixed-dimension multimodal and we can find their corresponding equations in Table 3 with the number of dimensions that were considered, the range of the search space and the optimal value for each function, respectively. Finally, in Figure 11 we show the graphical representations of these benchmark functions in their 3D versions.

For the experiments that were performed in this paper, we are presenting three different configurations for the FWA-GWO algorithm and we can find their descriptions as follows:

(1) Version 1

- (i) 4 packs
- (ii) 6 wolves for each pack
- (iii) 625 iterations
- (iv) 15,000 function evaluations

(2) Version 2

- (i) 5 packs
- (ii) 6 wolves for each pack
- (iii) 500 iterations
- (iv) 15,000 function evaluations

(3) Version 3

- (i) 8 packs
- (ii) 5 wolves for each pack
- (iii) 375 iterations
- (iv) 15,000 function evaluations

For comparing the performance of all the algorithms, we performed hypothesis tests (Z-test) [22, 27, 28] with the following parameters:

- (i) μ_1 = New Method (FWA-GWO)
- (ii) μ_2 = FWA or GWO
- (iii) The mean of the FWA-GWO is lower than the mean of the original method (claim)
- (iv) $\mathbf{H_0}: \mu_1 \geq \mu_2$
- (v) $\mathbf{H_a}: \mu_1 < \mu_2$ (Claim)
- (vi) $\alpha = 0.05$
- (vii) $Z_0 = -1.645$

We show the hypothesis tests results in the following tables, where we are comparing FWA-GWO with FWA and FWA-GWO with GWO, respectively.

Table 2: Unimodal and multimodal Benchmark functions.

Function	Range	$f_{\min}$
$f_1(x) = \sum_{i=1}^{n} {x_i}^2$	[−100, 100]	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_1 \rvert + \prod_{i=1}^{n} \lvert x_1 \rvert$	[−100, 100]	0
$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j \right)^2$	[−100, 100]	0
$f_4(x) = \max_i \{ \lvert x_1 \rvert, 1 \le i \le n \}$	[−100, 100]	0
$f_5(x) = \sum_{i=1}^{n-1} \left[100 \left(x_{i+1} - {x_i}^2 \right)^2 + (x_1 - 1)^2 \right]$	[−100, 100]	0
$f_6(x) = \sum_{i=1}^{n} ([x_1 + 0.5])^2$	[−100, 100]	0
$f_7(x) = \frac{1}{2} \sum_{i=1}^{d} \left(x_i^4 - 16 x_i^2 + 5 x_i \right)$	[−100, 100]	−39.17
$f_8(x) = \sum_{i=1}^{d/4} \left[(x_{4i-3} + 10 x_{4i-2})^2 + 5 (x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2 x_{4i-1})^4 + 10 (x_{4i-3} - x_{4i})^4 \right]$	[−100, 100]	0
$f_9(x) = \sum_{i=1}^{d} x_i^2 + \left(\sum_{i=1}^{d} 0.5 i x_i \right)^2 + \left(\sum_{i=1}^{d} 0.5 i x_i \right)^4$	[−100, 100]	0
$f_{10}(x) = \sum_{i=1}^{n} \left[{x_i}^2 - 10 \cos (2 \pi x_i) + 10 \right]$	[−100, 100]	0
$f_{11}(x) = 20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} {x_i}^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^{n} \cos (2 \pi x_i) \right) + 20 + e$	[−100, 100]	0
$f_{12}(x) = \frac{\pi}{n} \left\{ 10 \sin (\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 \left[1 + 10 \sin^2 (\pi y_{y+1}) \right] + (y_n - 1)^2 \right\} + \sum_{i=1}^{n} u (x_i, 10, 100, 4)$ $y_1 = 1 + \frac{x_i + 1}{4}$ $u(x_1, a, k, m) = \begin{cases} k (x_i - a)^m, & x_i > a, \\ 0, & -a < x_i < a \\ k (-x_i - a)^m, & x_i < -a \end{cases}$	[−100, 100]	0
$f_{13}(x) = \frac{1}{4000} \sum_{i=1}^{n} {x_1}^2 - \prod_{i=1}^{n} \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	[−100, 100]	0

4.1. Comparison between GWO and FWA-GWO. In Tables 4–11, we are presenting a comparison and hypothesis tests between the GWO and the hybrid method for 30, 60, and 90 dimensions and the three different versions, respectively.

From Table 4 we can conclude based on the hypothesis test that, for the case of 30 dimensions, the FWA-GWO in version 1 is better in 7 of the 13 benchmark functions that are analyzed in this paper.

In Table 5 we show the results of a hypothesis test for the original GWO and the FWA-GWO in version 2 and in this case the proposed method is better in 6 of the 13 analyzed functions.

Finally, Table 6 shows the results of the hypothesis tests when we are using version 3 of the FWA-GWO, in other words when the algorithm has 8 packs and 5 wolves for each pack, then we can conclude that the proposed hybrid approach (FWA-GWO) is better in 7 of the 13 functions that were tested in this paper.

In the following figures we are illustrating in a graphical way the results of the hypothesis tests, more specifically the z-values. The comparison is between the conventional algorithms and the proposed FWA-GWO with the different versions.

In this case the blue bar represents the z value of the hypothesis test with 60 dimensions, the green bar represents the z values of the hypothesis with 90 dimensions, and finally the red bar represents the critical value according of the hypothesis test that was described above, which in this case is −1.645. Therefore, while the blue and green bars are less than −1.645, the hybrid method is better.

For 60 dimensions in the analyzed problems, the FWA-GWO in version 1 is better in 6 of the 13 analyzed benchmark functions according to Figure 12 and also we can find the z-values for 90 dimensions between the GWO and version 1 of the FWA-GWO. In this case the proposed method is better in 4 benchmark functions.

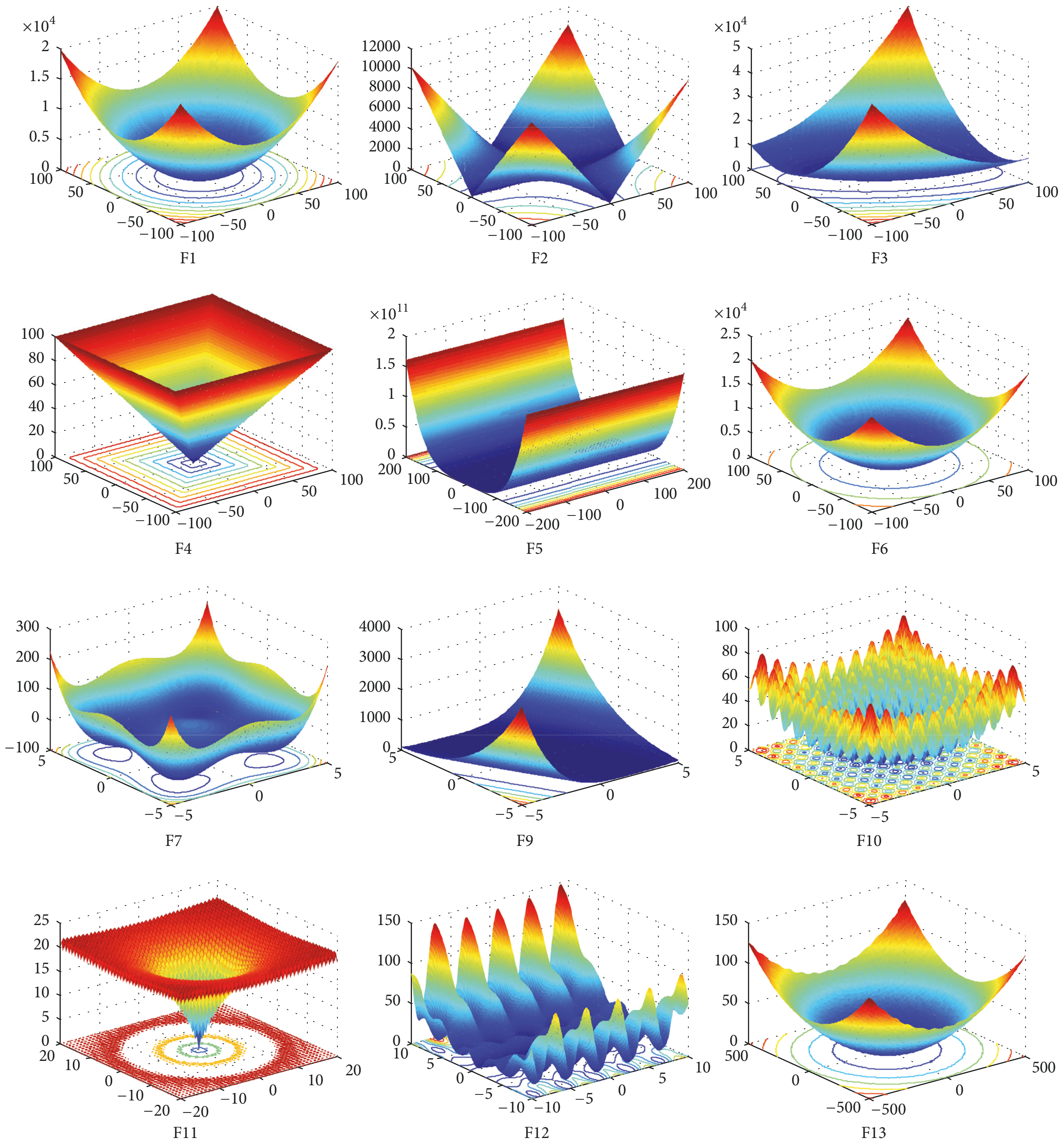

FIGURE 10: Examples of the unimodal and multimodal benchmark functions in their 3D versions.

In Figure 13 we can conclude that for 60 dimensions, the FWA-GWO in version 2 is better only in 3 of the 13 analyzed functions, respectively, and for 90 dimensions. In this case, the version 2 of the proposed method (FWA-GWO) has not a good performance because it is better than the original method only in 2 of the analyzed benchmark functions.

Finally, we are presenting in Figure 14 the z-values when the problem has 60 and 90 dimensions, respectively. According to the hypothesis tests between GWO and the FWA-GWO in version 3 we can conclude that the FWA-GWO is better only in 3 of the 13 analyzed functions, respectively, with 60 dimensions and for 90 dimensions we can find that version 3

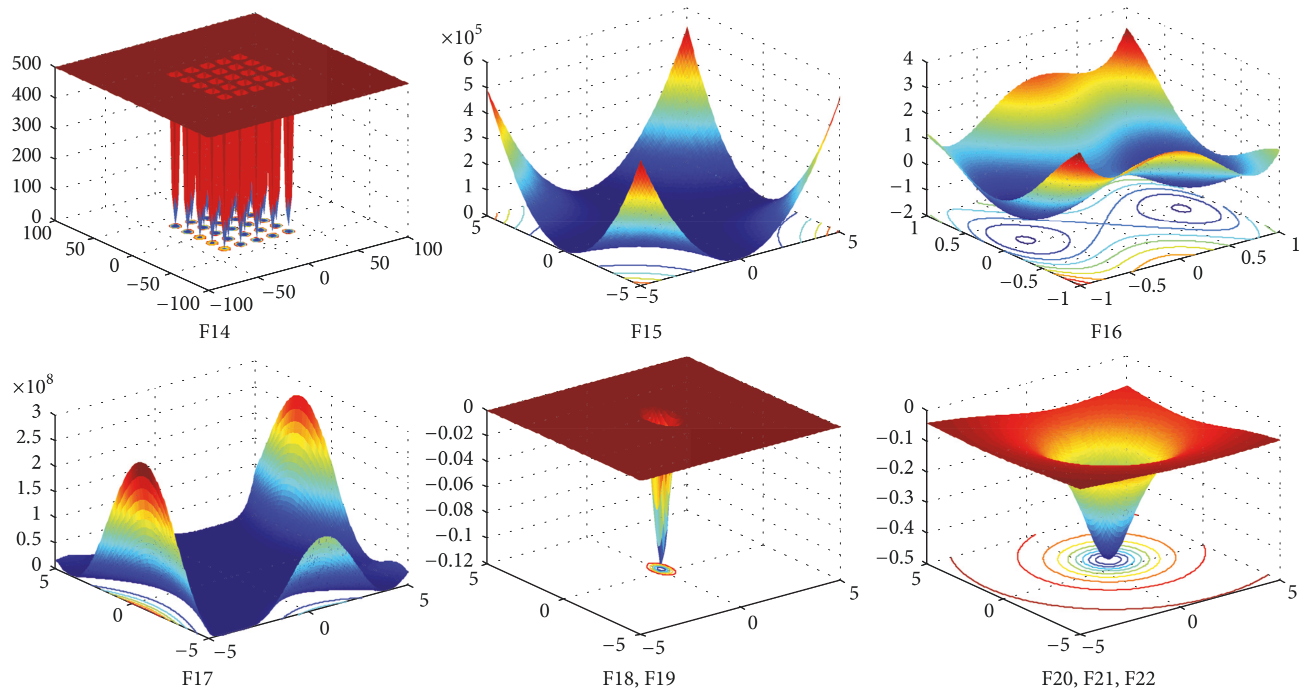

Figure 11: Examples of the fixed-dimension multimodal benchmark functions in their 3D versions.

Table 3: Fixed-dimension multimodal benchmark functions.

Function	Dim	Range	$f_{\min}$
$f_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=2}^{2}\left(x_i - a_{ij}\right)^6}\right)^{-1}$	2	$[-65, 65]$	1
$f_{15}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1\left(b_1^2 + b_i x_2\right)}{b_1^2 + b_i x_3 + x_4}\right]^2$	4	$[-5, 5]$	0.00030
$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	−1.0316
$f_{17}(x) = \left[1 + \left(x_1 + x_2 + 1\right)^2\left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right] \times \left[30 + \left(2x_1 - 3x_2\right)^2 \times \left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)\right]$	2	$[-2, 2]$	3
$f_{18}(x) = -\sum_{i=1}^{4} c_1 \exp\left(-\sum_{j=1}^{3} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	3	$[1, 3]$	−3.86
$f_{19}(x) = -\sum_{i=1}^{4} c_1 \exp\left(-\sum_{j=1}^{6} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	6	$[0, 1]$	−3.32
$f_{20}(x) = -\sum_{i=1}^{5}\left[\left(X - a_i\right)\left(X - a_i\right)^T + c_i\right]^{-1}$	4	$[0, 10]$	−10.1532
$f_{21}(x) = -\sum_{i=1}^{7}\left[\left(X - a_i\right)\left(X - a_i\right)^T + c_i\right]^{-1}$	4	$[0, 10]$	−10.4028
$f_{22}(x) = -\sum_{i=1}^{10}\left[\left(X - a_i\right)\left(X - a_i\right)^T + c_i\right]^{-1}$	4	$[0, 10]$	−10.5363

is better in 4 of the 13 analyzed functions based on the results of the hypothesis test.

4.2. Comparison between FWA and FWA-GWO. In addition we are presenting the hypothesis testing between the FWA and the FWA-GWO; Table 7 shows the results of the averages, standard deviations, and Z-values for 30 dimensions with the FWA-GWO in version 1.

We can conclude from Table 7 that the FWA-GWO is better in 5 of the 13 benchmark functions that are analyzed in this paper.

Table 8 shows the results when we used the FWA-GWO in version 2 and in this case we can find that it has a better

Table 4: Comparison between GWO and the FWA-GWO in version 1 with 30 dimensions.

			30 dimensions		
Function	GWO	STD	Hybrid V1	STD	Z-value
F1	$1.45E-27$	$2.39E-27$	**6.43E − 28**	**1.17E − 27**	−2.1364
F2	$1.28E-15$	$1.01E-15$	**1.70E − 16**	**1.63E − 16**	−7.6564
F3	**1.96E − 05**	**4.19E − 05**	$8.47E-04$	0.0025	2.3188
F4	**8.40E − 07**	**8.83E − 07**	0.0383	0.0584	4.6322
F5	**28.3085**	**8.5466**	27.6283	0.5878	−0.5614
F6	**0.7575**	**0.3307**	0.8856	0.4284	1.6727
F7	−843.5616	64.6298	**−775.8908**	**98.3576**	−4.0658
F8	13.2227	47.144	**1.7697**	**6.7631**	−1.7004
F9	**154.0219**	**272.2662**	9641.7985	11406.7060	5.8798
F10	4.6208	4.4183	**1.7310**	**2.2942**	−4.1045
F11	20.8930	0.0927	**5.8606**	**9.4934**	−11.1962
F12	**0.0617**	**0.0779**	0.0693	0.0297	0.6441
F13	0.0053	0.0103	**0.00**	**0.00**	−3.6561

Table 5: Comparison between GWO and the FWA-GWO in version 2 with 30 dimensions.

			30 dimensions		
Function	GWO	STD	Hybrid V2	STD	Z-value
F1	**1.45E − 27**	**2.39E − 27**	$1.84E-22$	$7.49E-22$	1.7398
F2	**1.28E − 15**	**1.01E − 15**	$1.17E-13$	$1.20E-13$	6.8015
F3	**1.96E − 05**	**4.19E − 05**	0.0044	0.0166	1.8497
F4	**8.40E − 07**	**8.83E − 07**	0.0409	0.0440	6.5681
F5	**28.3085**	**8.5466**	27.5222	0.5276	−0.6493
F6	0.7575	0.3307	**0.5746**	**0.3179**	−2.8192
F7	−843.5616	64.6298	**−811.8833**	**98.7647**	−1.8978
F8	13.2227	47.144	**7.55E − 05**	**1.91E − 04**	−1.9832
F9	**154.0219**	**272.2662**	15241.4387	16021.8731	6.6577
F10	4.6208	4.4183	**1.3627**	**2.0485**	−4.7306
F11	20.8930	0.0927	**1.71E − 11**	**4.23E − 11**	−1594.3538
F12	**0.0617**	**0.0779**	0.0705	0.0316	0.7432
F13	0.0053	0.0103	**0**	**0**	−3.6561

Table 6: Comparison between GWO and the FWA-GWO in version 3 with 30 dimensions.

			30 dimensions		
Function	GWO	STD	Hybrid V3	STD	Z-value
F1	**1.45E − 27**	**2.39E − 27**	$1.78E-18$	$2.70E-18$	4.6680
F2	**1.28E − 15**	**1.01E − 15**	$4.19E-11$	$3.52E-11$	8.4273
F3	**1.96E − 05**	**4.19E − 05**	0.0095	0.0241	2.7765
F4	**8.40E − 07**	**8.83E − 07**	0.0248	0.0207	8.4815
F5	**28.3085**	**8.5466**	27.7054	0.5847	−0.4979
F6	0.7575	0.3307	**0.59297**	**0.35281**	−2.4065
F7	−843.5616	64.6298	**−753.7345**	**291.2693**	−2.1289
F8	13.2227	47.144	**0.1544**	**0.6904**	−1.9599
F9	**154.0219**	**272.2662**	6439.18	23019.40	1.9305
F10	4.6208	4.4183	**3.1066**	**3.9244**	−1.8118
F11	20.8930	0.0927	**1.33E − 09**	**1.34E − 09**	−1594.3538
F12	0.0617	0.0779	**0.0416**	**0.0249**	−1.7389
F13	0.0053	0.0103	**9.33E − 17**	**2.21E − 16**	−3.6561

Table 7: Comparison between FWA and the FWA-GWO in version 1 with 30 dimensions.

			30 dimensions		
Function	FWA	STD	Hybrid V1	STD	Z-value
F1	0.0237	0.0148	**6.43E − 28**	**1.17E − 27**	−11.3515
F2	0.5501	0.2083	**1.70E − 16**	**1.63E − 16**	−18.6715
F3	**2.19E − 04**	**0.0005**	8.47E − 04	0.0025	1.7326
F4	**0.0407**	**0.0078**	0.0383	0.0584	−0.2951
F5	29.8533	1.6787	**27.6283**	**0.5878**	−8.8458
F6	**1.0293**	**0.7131**	0.8856	0.4284	−1.2221
F7	−851.1396	80.7011	**−775.8908**	**98.3576**	−4.1822
F8	**0.3258**	**0.3166**	1.7697	6.7631	1.5079
F9	**9.0217**	**17.8576**	9641.7985	11406.7060	5.9714
F10	**1.6570**	**0.8116**	1.7310	2.2942	0.2150
F11	**0.0449**	**0.0081**	5.8606	9.4934	4.3318
F12	**0.0532**	**0.0318**	0.0693	0.0297	2.6125
F13	7.04E − 04	5.16E − 04	**0.00**	**0.00**	−9.6465

Table 8: Comparison between FWA and the FWA-GWO in version 2 with 30 dimensions.

			30 dimensions		
Function	FWA	STD	Hybrid V2	STD	Z-value
F1	0.0237	0.0148	**1.84E − 22**	**7.49E − 22**	−11.3515
F2	0.5501	0.2083	**1.17E − 13**	**1.20E − 13**	−18.6715
F3	**2.19E − 04**	**0.0005**	0.0044	0.0166	1.7642
F4	**0.0407**	**0.0078**	0.0409	0.0440	0.0250
F5	29.8533	1.6787	**27.5222**	**0.5276**	−9.3676
F6	1.0293	0.7131	**0.5746**	**0.3179**	−4.1179
F7	−851.1396	80.7011	**−811.8833**	**98.7647**	−2.1764
F8	0.3258	0.3166	**7.55E − 05**	**1.91E − 04**	−7.2752
F9	**9.0217**	**17.8576**	15241.4387	16021.8731	6.7226
F10	**1.6570**	**0.8116**	1.3627	2.0485	−0.9446
F11	0.0449	0.0081	**1.71E − 11**	**4.23E − 11**	−39.3428
F12	**0.0532**	**0.0318**	0.0705	0.0316	2.7297
F13	7.04E − 04	5.16E − 04	**0**	**0**	−9.6465

Table 9: Comparison between FWA and the FWA-GWO in version 3 with 30 dimensions.

			30 dimensions		
Function	FWA	STD	Hybrid V3	STD	Z-value
F1	0.0237	0.0148	**1.78E − 18**	**2.70E − 18**	−11.3515
F2	0.5501	0.2083	**4.19E − 11**	**3.52E − 11**	−18.6715
F3	**2.19E − 04**	**0.0005**	0.0095	0.0241	2.7175
F4	0.0407	0.0078	**0.0248**	**0.0207**	−5.0876
F5	29.8533	1.6787	**27.7054**	**0.5847**	−8.5443
F6	1.0293	0.7131	**0.59297**	**0.35281**	−3.8782
F7	−851.1396	80.7011	**−753.7345**	**291.2693**	−2.2788
F8	**0.3258**	**0.3166**	0.1544	0.6904	−1.5961
F9	**9.0217**	**17.8576**	6439.18	23019.40	1.9752
F10	**1.6570**	**0.8116**	3.1066	3.9244	2.5577
F11	0.0449	0.0081	**1.33E − 09**	**1.34E − 09**	−39.3428
F12	0.0532	0.0318	**0.0416**	**0.0249**	−2.0307
F13	7.04E − 04	5.16E − 04	**9.33E − 17**	**2.21E − 16**	−9.6465

Table 10: Comparison of the best results among the three methods with 60 dimensions.

	60 dimensions				
Function	GWO	FWA	Hybrid V1	Hybrid V2	Hybrid V3
F1	1.03$E-18$	8.76$E-03$	**1.45E − 20**	5.24$E-16$	1.49$E-12$
F2	2.04$E-10$	1.88$E-01$	**2.69E − 12**	2.81$E-10$	2.32$E-08$
F3	0.0246	**9.54E − 08**	0.0062	0.2206	0.0579
F4	**3.29E − 04**	1.72$E-02$	1.0835	0.4721	0.2902
F5	56.0643	**6.5639**	56.5365	56.7458	56.8672
F6	2.4988	**4.70E − 01**	2.9397	1.9858	1.5036
F7	−1103.01	−1123.702	−1022.80	−1114.42	**−963.5091**
F8	9.13$E-05$	1.27$E-01$	0.0016	**6.75E − 06**	0.0062
F9	8429.33	**2.6085**	21956.43	6268.08	853.71
F10	2.22$E-11$	1.0527	**2.27E − 13**	6.90$E-11$	5.10$E-08$
F11	20.9788	2.97$E-02$	**1.75E − 10**	2.21$E-09$	1.63$E-07$
F12	0.0595	5.27$E-03$	0.0067	0.0782	**8.48E − 04**
F13	**0.00**	1.46$E-04$	**0.00**	**0.00**	3.99$E-14$

Table 11: Comparison of the best results among the three methods with 90 dimensions.

	90 dimensions				
Function	GWO	FWA	Hybrid V1	Hybrid V2	Hybrid V3
F1	1.81$E-14$	2.91$E-02$	**1.70E − 15**	1.27$E-12$	7.25$E-09$
F2	5.12$E-08$	1.59$E-01$	**3.95E − 10**	3.34$E-08$	1.75$E-06$
F3	9.3068	**1.23E − 07**	1.1431	3.5748	0.7638
F4	0.0198	**2.40E − 02**	5.2841	1.2142	1.6948
F5	**85.9349**	89.8066	87.2861	87.5498	87.1859
F6	5.9709	**1.7693**	5.5327	5.1995	3.1307
F7	**−1495.39**	−1551.06	−2678.99	−1968.90	−1507.83
F8	2.33$E-04$	0.3271	0.0369	**2.22E − 04**	0.1971
F9	57096.74	**5.7946**	42806.79	19650.57	10246.35
F10	2.5244	2.0417	**5.19E − 10**	1.1558	9.9620
F11	21.1015	0.0235	**2.85E − 07**	7.28$E-07$	1.32$E-05$
F12	0.1476	0.0167	**0.0078**	0.3672	0.0338
F13	2.66$E-15$	0.0004	**7.77E − 16**	1.12$E-14$	1.02$E-10$

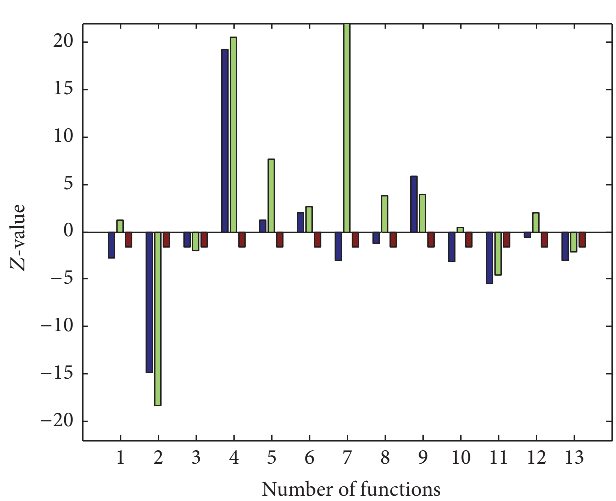

Figure 12: Comparison between GWO and the FWA-GWO in version 1 with 60 and 90 dimensions.

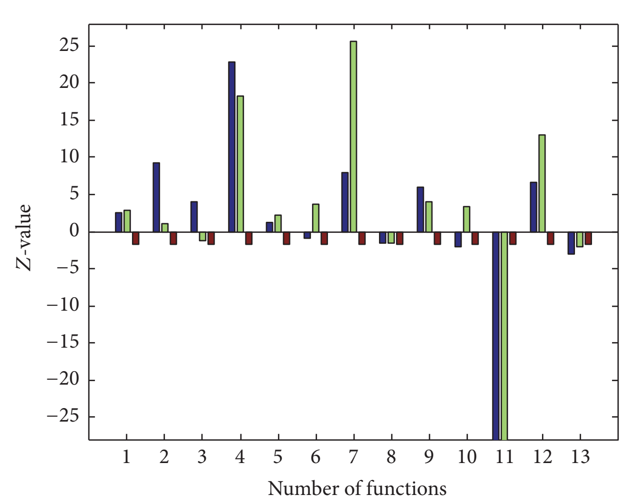

Figure 13: Comparison between GWO and the FWA-GWO in version 2 with 60 and 90 dimensions.

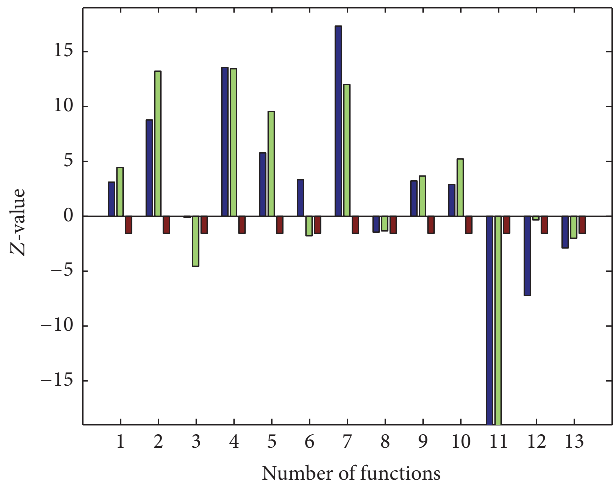

Figure 14: Comparison between GWO and FWA-GWO in version 3 with 60 and 90 dimensions.

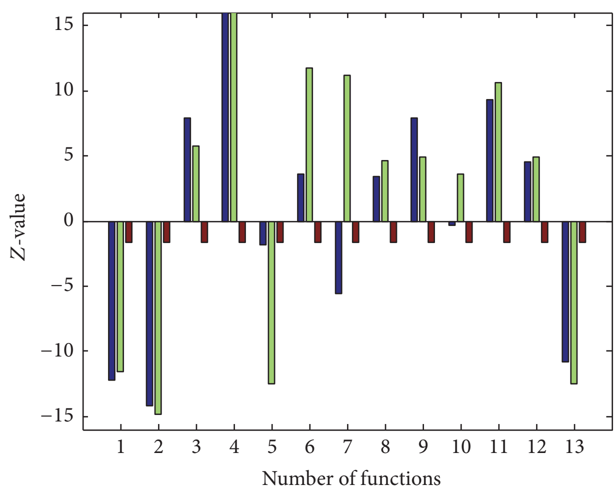

Figure 15: Comparison between FWA and FWA-GWO in version 1 with 60 and 90 dimensions.

a performance because it is better than the conventional FWA in 8 of the 13 total functions.

Finally, for 30 dimensions in Table 9 we are presenting the last version of the FWA-GWO algorithm that was analyzed in this paper and we can conclude that, for 30 dimensions, this version has the best configuration of the parameters because, according of the hypothesis testing, the FWA-GWO is better than FWA in 9 of the 13 benchmarks functions that were analyzed.

Based on Figure 15 we can mention that for 60 dimensions the FWA-GWO version 1 is better than the conventional FWA in 5 of the 13 benchmark functions that were analyzed in this paper and when the problems have 90 dimensions we can find that, for this number of dimensions, version number 1 is better in 4 benchmark functions that were analyzed.

In Figure 16 we can find in a graphical way the z values of the hypothesis test between FWA and the FWA-GWO in version 2 and for 60 dimensions we can mention that we have

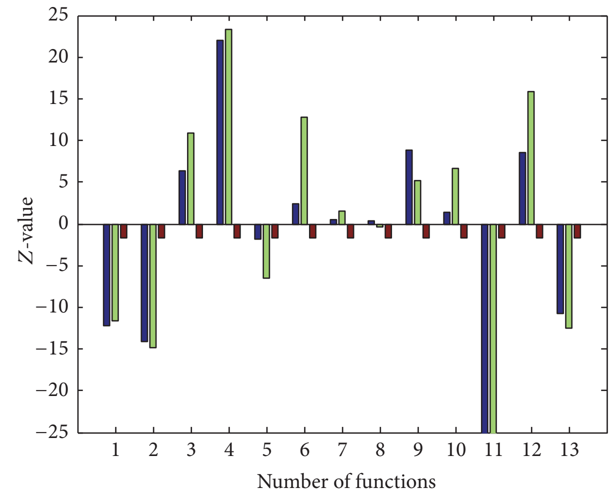

Figure 16: Comparison between FWA and FWA-GWO in version 2 with 60 and 90 dimensions.

a similar performance with respect to version 1 because both methods are better than the conventional FWA in 5 of the 13 analyzed benchmark functions. Finally, for 90 dimensions we can conclude that the FWA-GWO is better in 5 of the 13 benchmark functions that are analyzed.

Finally, in Figure 17 we are presenting the results of the hypothesis test between FWA and version 3 of the FWA-GWO and we can mention that for 60 dimensions we can conclude that in this version of FWA-GWO it only has better performance in 4 benchmark functions and for 90 dimensions FWA-GWO is better in 5 of the 13 benchmark functions that are analyzed in this paper.

4.3. Comparison of the Best Solutions among the Three Methods. In Figures 12, 13, and 14 we can notice that when the problem has 60 and 90 dimensions, respectively, we can find an interesting behavior and that, for these benchmark functions, the conventional GWO has better performance than the FWA-GWO according to the analyzed hypothesis test.

Also we can mention that, for the comparison between the FWA and the FWA-GWO, for 60 and 90 dimensions, respectively, the FWA-GWO has better performance around half of total analyzed benchmark functions and it is important to say that we are presenting only three different configurations of the FWA-GWO and for this comparison the performance can be significantly improved if we change these configurations.

In addition we consider that it is very important to summarize in the following tables a brief comparison only of the best results obtained in the 50 independent executions of each method for 60 and 90 dimensions, respectively, with the main goal of showing that although in a hypothesis test we cannot prove a best performance in some of the benchmark functions that are analyzed, the FWA-GWO still found a better result than the conventional algorithm.

Table 10 shows the best values of each method with 60 dimensions and we can conclude that in general the proposed FWA-GWO algorithm has better performance in 8 of the 13

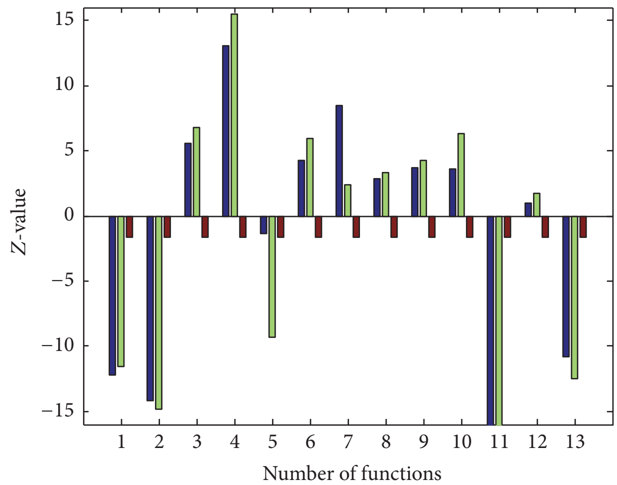

Figure 17: Comparison between FWA and FWA-GWO in version 3 with 60 and 90 dimensions.

analyzed functions, the FWA has better performance than the others in 4 of the 13 functions, and finally the GWO algorithm has better performance in 2 of the 13 benchmark functions that are analyzed in this paper. The best version of the FWA-GWO is with version 1 with 5 functions, and finally with versions 2 and 3, both methods are better in 2 benchmark functions, respectively.

Table 11 shows the general results for the best experiments of 50 runs that were executed with 90 dimensions and we can find that the FWA-GWO has better performance in 7 of the 13 analyzed benchmark functions. The FWA metaheuristic has better performance in 4 of the 13 functions and finally the GWO algorithm has better performance in 2 of 13 analyzed benchmark functions with 90 dimensions.

Based on Tables 10 and 11 we can find an interesting conclusion because we can note that the FWA-GWO has a better performance than the GWO and FWA, respectively, based on the best results that we obtained in the respective sample. This is very important because, for example, when we want to design fuzzy controllers, the design, the architecture, or the best plant is based on the best result that we obtained in the total experiments or executions and we can conclude that for these optimization problems the FWA-GWO has better performance than the conventional methods (GWO and FWA). Finally we can mention that for improving the results of the FWA-GWO in the hypothesis testing with 60 and 90 dimensions, respectively, we need to test other types of configurations of the FWA-GWO, because the configurations that we presented in this paper were chosen randomly.

Finally, we present the results of the rest of the total benchmark functions [F14, F22] that are analyzed in this paper, and these benchmark functions are called "fixed-dimension multimodal" and averages and standard deviation are presented in the following tables.

Tables 12 and 13 illustrate a brief comparison only with the averages of GWO, FWA, and the FWA-GWO in versions 1 and 2, respectively, and we can conclude that the three methods have a good performance with these benchmark functions.

In addition, Table 14 shows the comparison of the last version presented in this paper; the main goal of evaluating these benchmark functions is to prove that the methods work correctly and have a good performance with these types of problems and we only present a brief comparison in average without hypothesis test because the results are very similar.

4.4. Convergence of the FWA-GWO. In the following figures we are presenting convergence plots between the GWO and the FWA-GWO with two different problems and the difference is that in one problem the FWA-GWO (hybrid) has better performance and in others has not, according to the hypothesis test.

Figure 18 shows the convergence curve of the GWO and FWA-GWO algorithms, respectively, for the benchmark function number 4.

In Figure 19 we can notice the performance of the GWO and the FWA-GWO when we are using the benchmark function number 10. The red line represents the convergence curve of the GWO algorithm and the blue line represents the convergence of the FWA-GWO (hybrid).

It is important to mention that in conclusion we can say that, with the help of Figures 18 and 19, the FWA-GWO always finds other solutions throughout the iterations and this behavior is good because the algorithm always converges and avoids the local optima and this process occurs through the advantages of the hybridization between the GWO and FWA algorithms. On the other hand we can note in Figure 18 that, for example, in this problem (F10), the GWO algorithm falls into a local minimum and cannot avoid this problem. In addition, we can test with other configurations in the FWA-GWO toolbox, which is presented in [29].

5. Conclusions

In this paper we have presented the hybridization among two metaheuristics that were described above and the new hybrid method is called FWA-GWO. Is important to mention that

Table 12: Comparison among GWA, FWA, and FWA-GWO in version 1 in fixed-dimensions multimodal benchmark functions.

Fixed-dimensions multimodal benchmark functions						
Function	GWO	STD	FWA	STD	Hybrid V1	STD
F14	4.0425	4.2528	0.9981	$2.21E-04$	2.7499	2.8626
F15	$3.37E-04$	$6.25E-04$	$6.03E-04$	$2.76E-04$	$7.42E-04$	$2.70E-04$
F16	−1.0316	1.0316	−1.0316	$4.96E-05$	−1.0315	$5.60E-04$
F17	3.0000	3.0000	3.0011	0.0018	3.0007	$8.20E-04$
F18	−3.8626	3.8628	−0.2979	$6.20E-06$	−3.8612	0.0024
F19	−3.2865	3.2506	−3.2655	0.0599	−3.3124	0.0449
F20	−10.1514	9.14015	−10.1466	0.0054	−5.3093	0.6601
F21	−10.4015	8.5844	−10.3929	0.0144	−5.2378	0.5854
F22	−10.5343	8.55899	−10.5113	0.0481	−5.3646	0.4713

Table 13: Comparison among GWA, FWA, and FWA-GWO in version 2 in fixed-dimensions multimodal benchmark functions.

Fixed-dimensions multimodal benchmark functions						
Function	GWO	STD	FWA	STD	Hybrid V2	STD
F14	4.0425	4.2528	0.9981	$2.21E-04$	4.1664	3.7214
F15	$3.37E-04$	$6.25E-04$	$6.03E-04$	$2.76E-04$	$7.00E-04$	$3.18E-04$
F16	−1.0316	1.0316	−1.0316	$4.96E-05$	−1.0316	$2.61E-05$
F17	3.0000	3.0000	3.0011	0.0018	3.0009	$9.83E-04$
F18	−3.8626	3.8628	−0.2979	$6.20E-06$	−3.8612	0.0022
F19	−3.2865	3.2506	−3.2655	0.0599	−3.3093	0.0484
F20	−10.1514	9.14015	−10.1466	0.0054	−5.5335	0.9876
F21	−10.4015	8.5844	−10.3929	0.0144	−5.3205	0.6697
F22	−10.5343	8.55899	−10.5113	0.0481	−5.5058	0.7676

Table 14: Comparison among GWA, FWA, and FWA-GWO in version 3 in fixed-dimensions multimodal benchmark functions.

Fixed-dimensions multimodal benchmark functions						
Function	GWO	STD	FWA	STD	Hybrid V3	STD
F14	4.0425	4.2528	0.9981	$2.21E-04$	4.1338	3.5058
F15	$3.37E-04$	$6.25E-04$	$6.03E-04$	$2.76E-04$	$9.05E-04$	$6.44E-04$
F16	−1.0316	1.0316	−1.0316	$4.96E-05$	−1.0316	$4.86E-06$
F17	3.0000	3.0000	3.0011	0.0018	3.0016	0.0019
F18	−3.8626	3.8628	−0.2979	$6.20E-06$	−3.8617	0.0017
F19	−3.2865	3.2506	−3.2655	0.0599	−3.3220	$2.13E-05$
F20	−10.1514	9.14015	−10.1466	0.0054	−5.1421	0.4826
F21	−10.4015	8.5844	−10.3929	0.0144	−5.3531	0.8978
F22	−10.5343	8.55899	−10.5113	0.0481	−5.3103	0.6953

an advantage of this new hybrid method is the capacity of choosing between the exploitation and exploration phases in the algorithm that are two main features of metaheuristics, so it is very important to choose this abilities according to the problem that we want to solve.

According to the results presented in this paper we can conclude that for 30 dimensions the different versions of the hybrid method are better than the GWO and FWA, respectively, according to the hypothesis test presented. On the other hand, we can note that for 60 and 90 dimensions in the hypothesis tests the performance of FWA-GWO is poor, but we can note an interesting behavior, and that if we analyzed the best result of the total executions, we found the best performance in the different FWA-GWO versions. It is important to mention that the configuration of the hybrid method for these types of problems can be significantly improved and users can test other configurations of FWA-GWO on the Toolbox.

Taking into account the fact that the proposed FWA-GWO method obtained the best result in 60 and 90 dimensions, we can conclude that the FWA-GWO can perform in a better way in problems of control and design of structures in computational intelligence than the FWA or GWO algorithms [30–32], respectively.

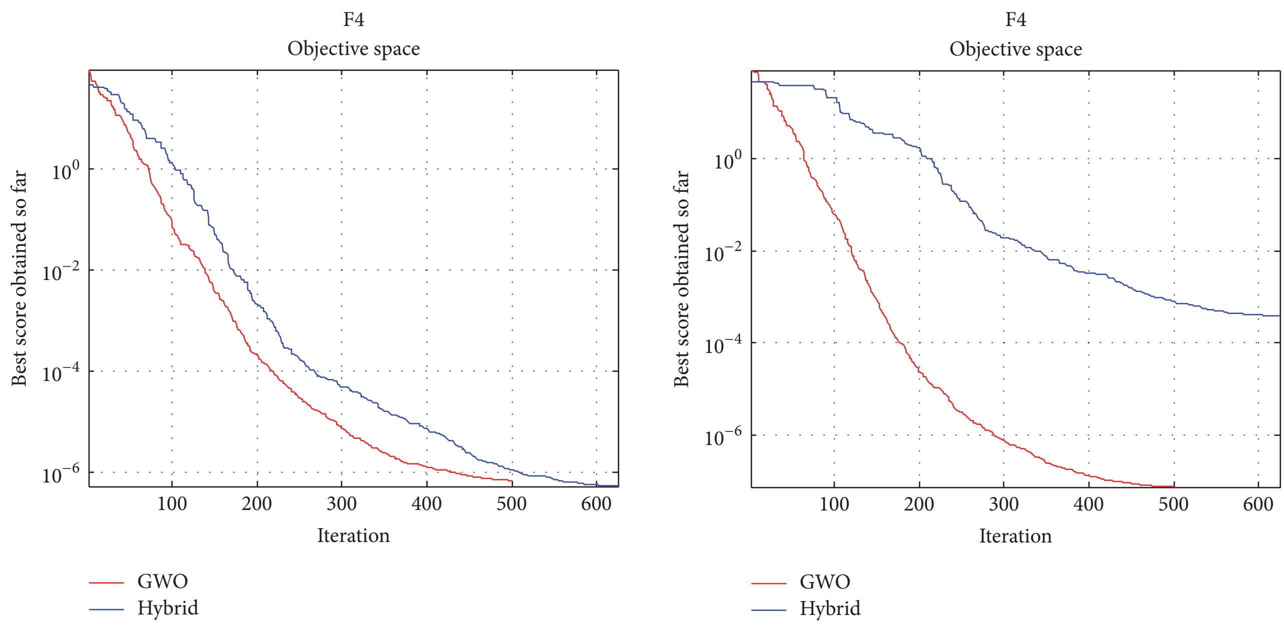

Figure 18: Convergence curve of GWO and FWA-GWO Algorithm with F4.

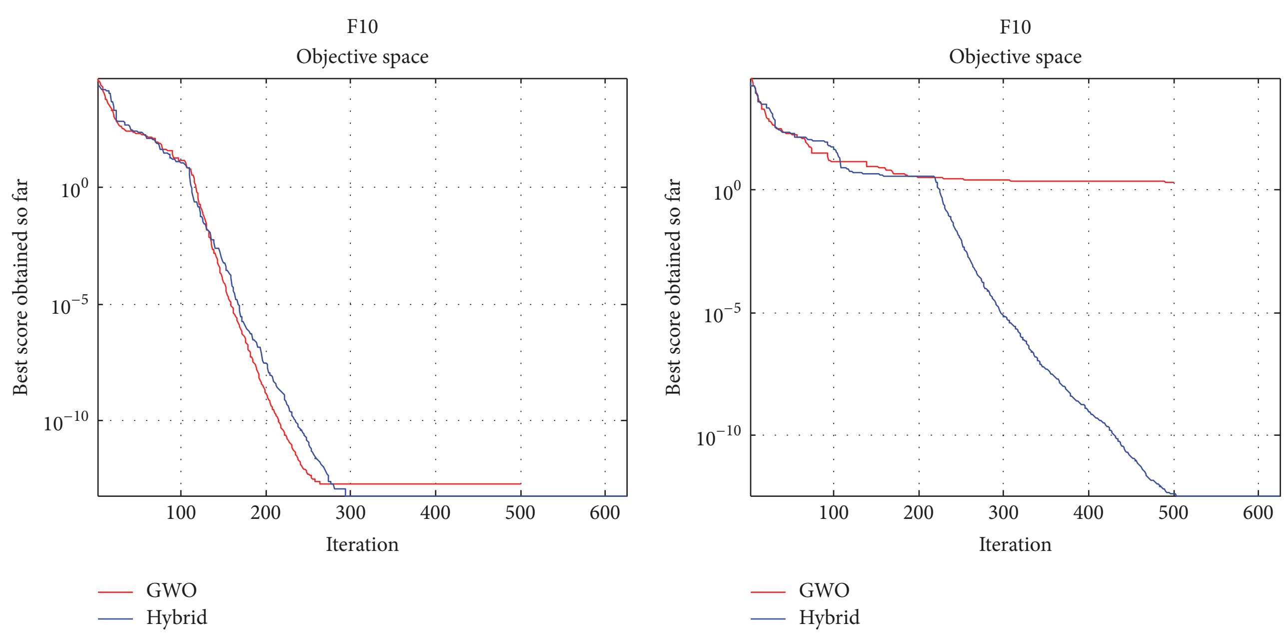

Figure 19: Convergence curve of GWO and FWA-GWO with F10.

Also we can conclude that the FWA-GWO is an algorithm that has diversity in the method for searching for the optimal value, because in the total population the algorithm has a set n of possible best solutions, so this feature is important because it helps avoid the local optima or false best values, searching always for possible different solutions. Finally, the proposed hybrid method (FWA-GWO) could be easily adapted to solve multiobjective problems.

As future work, we could test the FWA-GWO algorithm in problems with control of plants, as a first choice or as a second option, and dynamically adjust the parameters into the FWA-GWO algorithm with type 1 and interval type 2 fuzzy logic, respectively.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm intelligence: from natural to artificial systems*, OUP, USA, 1999.

[2] U. Can and B. Alatas, "Physics based metaheuristic algorithms for global optimization," *American Journal of Information Science and Computer Engineering*, vol. 1, pp. 94–106, 2015.

[3] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[4] B. Melián and J. Moreno, "Metaheurísticas: una visión global," *Revista Iberoamericana de Inteligencia Artificial*, vol. 19, pp. 7–28, 2003.

[5] F. Al Adwan, M. Al Shraideh, and M. R. Saleem Al Saidat, "A genetic algorithm approach for breaking of bimplified data encryption standard," *International Journal of Security and Its Applications*, vol. 9, no. 9, pp. 295–304, 2015.

[6] J. Soto, P. Melin, and O. Castillo, "Optimization of the Fuzzy Integrators in Ensembles of ANFIS Model for Time Series Prediction: The case of Mackey-Glass," in *Proceedings of the IEEE Conference on Norbert Wiener in the 21st Century (21CW)*, pp. 1–8, Boston, Mass, USA, June 2014.

[7] Y. Tan, *Fireworks Algorithm*, Springer-Verlag, Berlin, Heidelberg, Germany, 2015.

[8] L. Rodríguez, O. Castillo, and J. Soria, "A study of parameters of the grey wolf optimizer algorithm for dynamic adaptation with fuzzy logic," *Studies in Computational Intelligence*, vol. 667, pp. 371–390, 2017.

[9] Y. Zheng, Q. Song, and S. Chen, "Multiobjective fireworks optimization for variable-rate fertilization in oil crop production," *Applied Soft Computing*, vol. 13, no. 11, pp. 4253–4263, 2013.

[10] J. Li, S. Zheng, and Y. Tan, "Adaptive fireworks algorithm," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '14)*, pp. 3214–3221, Springer, Beijing, China, July 2014.

[11] Y. Tan and Y. Zhu, *Fireworks Algorithm for Optimization*, Springer-Verlag, Berlin, Heidelberg, Germany, 2010.

[12] J. Li, S. Zheng, and Y. Tan, "The Effect of Information Utilization: Introducing a Novel Guiding Spark in the Fireworks Algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 153–166, 2017.

[13] Y. Tan and S. Zheng, "Dynamic search in fireworks algorithm," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'14)*, pp. 3222–3229, July 2014.

[14] N. H. Abdulmajeed and M. Ayob, "A firework algorithm for solving capacitated vehicle routing problem," *International Journal of Advancements in Computing Technology (IJACT)*, vol. 6, no. 1, pp. 79–86, 2014.

[15] J. Li and Y. Tan, "The bare bones fireworks algorithm: A minimalist global optimizer," *Applied Soft Computing*, vol. 62, pp. 454–462, 2018.

[16] B. Zhang, Y.-J. Zheng, M.-X. Zhang, and S.-Y. Chen, "Fireworks Algorithm with Enhanced Fireworks Interaction," *IEEE Transactions on Computational Biology and Bioinformatics*, vol. 14, no. 1, pp. 42–55, 2017.

[17] S. Zheng, J. Li, A. Janecek, and Y. Tan, "A cooperative framework for fireworks algorithm," *IEEE Transactions on Computational Biology and Bioinformatics*, vol. 14, no. 1, pp. 27–41, 2017.

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

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

[20] L. Rodriguez, O. Castillo, J. Soria et al., "A fuzzy hierarchical operator in the grey wolf optimizer algorithm," *Applied Soft Computing*, vol. 57, pp. 315–328, 2017.

[21] A. A. Heidari and P. Pahlavani, "An efficient modified grey wolf optimizer with Lévy flight for optimization tasks," *Applied Soft Computing*, vol. 60, pp. 115–134, 2017.

[22] L. Rodriguez, O. Castillo, and J. Soria, "Grey wolf optimizer with dynamic adaptation of parameters using fuzzy logic," in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation, CEC 2016*, pp. 3116–3123, Canada, July 2016.

[23] J. Barraza, P. Melin, F. Valdez, and C. González, "Fireworks algorithm (FWA) with adaptation of parameters using fuzzy logic," *Studies in Computational Intelligence*, vol. 667, pp. 313–327, 2017.

[24] J. G. Digalakis and K. G. Margaritis, "On benchmarking functions for genetic algorithms," *International Journal of Computer Mathematics*, vol. 77, no. 4, pp. 481–506, 2001.

[25] M. Molga and C. Smutnicki, "Test functions for optimization needs," In press.

[26] X. S. Yang, "Test problems in optimization," in *Engineering Optimization: An Introduction with Metaheuristic Applications*, John Wiley & Sons, 2010, https://arxiv.org/abs/1008.0549v1.

[27] J. Barraza, P. Melin, F. Valdez, and C. I. Gonzalez, "Fuzzy FWA with dynamic adaptation of parameters," in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC'16)*, pp. 4053–4060, July 2016.

[28] R. Larson and B. Farber, *Elementary Statistics Picturing the World*, Perarson Education Inc, 2003.

[29] Toolbox FWAGWO, "Hybridization between the FWA and GWO algorithms," 2017, http://www.hafsamx.org/melin/ToolFWAGWO/.

[30] D. Guha, P. K. Roy, and S. Banerjee, "Load frequency control of interconnected power system using grey Wolf optimization," *Swarm and Evolutionary Computation*, vol. 27, pp. 97–115, 2016.

[31] R.-E. Precup, R.-C. David, and E. M. Petriu, "Grey Wolf optimizer algorithm-based tuning of fuzzy control systems with reduced parametric sensitivity," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 1, pp. 527–534, 2017.

[32] S. Saremi, S. Z. Mirjalili, and S. M. Mirjalili, "Evolutionary population dynamics and grey wolf optimizer," *Neural Computing and Applications*, vol. 26, no. 5, pp. 1257–1263, 2015.