A novel bio-inspired optimization model based on Yellow Saddle Goatfish behavior

[1]Daniel Zaldívar, [1]Bernardo Morales, [1,2]Alma Rodríguez, [1]Arturo Valdivia-G,
[1]Erik Cuevas, [1]Marco Pérez-Cisneros

[1]Departamento de Electrónica
Universidad de Guadalajara, CUCEI
Av. Revolución 1500, C.P 44430, Guadalajara, Jal,
México
erik.cuevas@cucei.udg.mx

[2]Desarrollo de Software
Centro de Enseñanza Técnica Industrial, Colomos
Calle Nueva Escocia 1885, Providencia 5a
Sección, C.P. 44638 Guadalajara, Jal, Mexico

Abstract

Several species of fish live in groups to increase their foraging efficiency and reproduction rates. Such groups are considered self-organized since they can adopt different cooperative actions without the presence of an apparent leader. One of their most interesting collaborative behaviors found in fish is the hunting strategy presented by the Yellow Saddle Goatfish (parupeneus cyclostomus). In this strategy, the complete group of fish is distributed in subpopulations to cover the whole hunting region. In each sub-population, all fish participate collectively in the hunt considering two different roles: chaser and blocker. In the hunt, a chaser fish actively tries to find the prey in a certain area whereas a blocker fish moves spatially to avoid the scape of the prey. In this paper, we develop the hunting model of Yellow Saddle Goatfish, which at some abstraction level can be characterized as a search strategy for optimization proposes. In the approach, different computational operators are designed in order to emulate this peculiar hunting behavior. With the use of this biological model, the new search strategy improves the optimization results in terms of accuracy and convergence in comparison to other popular optimization techniques. The performance of this method is tested by analyzing its results with other related evolutionary computation techniques. Several standard benchmark functions commonly used in the literature were considered to obtain optimization results. Furthermore, the proposed model is applied to solve certain engineering optimization problems. Analysis of the experimental results exhibits the efficiency, accuracy, and robustness of the proposed algorithm.

Keywords: Bio-inspired model, self-organized systems, evolutionary computation, optimization

1 Introduction

The cooperative behavior of several animal groups has attracted the attention of numerous scientific communities [1]. Biologists have examined this collaborative phenomenon to produce biological models while engineers have implemented these models as a method for finding the solution for many real-world complex problems [2].

From biological models, several approaches used to solve engineering problems has been produced [3]. One of the most important approaches of this category are the bio-inspired algorithms. They use our scientific knowledge about the collective behavior of many species as a guideline to design methods that can be conceived as search strategies. These techniques have demonstrated to be successfully applied in optimization problems [4]. Some of the most popular bio-inspired methods include the Particle Swarm Optimization (PSO) which mimics the social behavior of birds or fishes [5], the Synchronous-Asynchronous Particle Swarm Optimization (SA-PSO) that overcomes the PSO and is based on the time in which the particles are updated [6], the Artificial Bee Colony (ABC) that relies on the collaborative lifestyle of bee colonies [7], the Crow Search Algorithm (CSA) which is based on the intelligent behavior of crow [8], the Gray Wolf Optimization (GWO) which emulates the hunting strategies and the leadership of gray wolves [9], the Natural Aggregation Algorithm (NAA) based on collective decisions of group-living animal [10], the Whale Optimization Algorithm (WOA) which is inspired by the social behavior of humpback whales [11], the Selfish Herd Optimizer (SHO) that relies on selfish behavior in herds subjected to a predation risk [12], the Moth-Flame Optimization Algorithm (MFO) based on the navigation of moths in nature [13], or even the Artificial Fish Swarm Algorithm (AFSA) inspired by the collective movement of the fish according to their social behaviors [53]. Although these bio-inspired methods present interesting results, their search processes show numerous difficulties when they try to solve high multi-modal problems [14].

About half the fish species live in groups [15]. This aggregation provides many benefits, such as incrementing the foraging efficiency and reproduction rate [16]. Groups of fish can be categorized as self-organized systems since they can adopt different collaborative configurations without the presence of an apparent leader [17]. Cooperative behavior in fish emerges from several simple interactions among group members which are located in a specific area. The cooperation decision of each fish depends on the collective agreement of the whole group [18]. One of the most interesting collaborative behavior found in fish is the hunting strategy presented by the Yellow Saddle Goatfish (parupeneus cyclostomus) (YSG) [19]- [21]. In this strategy, the complete group of fish is distributed in sub-populations to cover the whole hunting region [20]. In each sub-population, all fish participates collectively in the hunt considering two different roles: chaser and blocker [20], [21]. In the hunt, a chaser fish actively tries to find the prey in a certain area whereas a blocker fish moves spatially to avoid the scape of the prey [20], [21].

In this paper, we develop the hunting model of Yellow Saddle Goatfish, which at some abstraction level can be characterized as a search strategy for optimization proposes. In the approach, called the Yellow Saddle Goatfish Algorithm (YSGA), the individuals emulate a set of fish which experiment with interaction based on the peculiar hunting behavior of the Yellow Saddle Goatfish. Initially, the proposed model designates a number of groups or sub-populations according to the spatial distribution of the individuals. The algorithm contemplates two distinct categories of search agents (fish): chasers and blockers. In each sub-population one fish assumes the role of chaser while the rest are considered blockers. Depending on the category, each element is undergone by a set of different evolutionary operations which emulate the different collaborative behaviors that are present in the natural hunting process. The fitness value represents the relative success that an element experiment during the hunting process. With the use of this biological model, the new search strategy improves the optimization results in terms of accuracy and convergence in comparison to other popular optimization techniques. The performance of our method is analyzed considering several state-of-the-art optimization techniques with a set of 27 well-known functions and five real-world engineering problems. The results validate that the introduced method reaches the best balance regarding accuracy and computational cost over its counterparts.

The organization of the paper is as follows: section 2 describes the yellow saddle goatfish behavior. In section 3, the proposed approach is explained in detail. Section 4 presents a comparative analysis among the proposed method and similar approaches. In section 5 presents the experimental results, comparisons, and applications to several engineering optimization problems. Finally, the conclusions are discussed in section 6.

2 Yellow Saddle Goatfish shoal behavior

On the reefs of the Indo-Pacific, especially on the Red Sea, researchers have observed the behavior and characteristics of the goatfish *Parupeneus cyclostomus* which is a very particular kind of fish of the family Mullidae and is often called yellow saddle goatfish [19]. Goatfishes has sensory barbels designed to detect hidden prey in sedimentary substrates. Its diet is mainly based on small fishes [19]-[21]. In addition to these characteristics, specialists have detected that goatfishes manifest a cooperative hunting strategy never seen before in other fish species [20]. This unusual behavior has attracted the attention of several scientific communities [22]- [24], since such a collaboration has been observed so far only in a few species, mostly on mammals such as lions, chimpanzees, dolphins, and killer whales [25]–[29].

Goatfish hunt in groups. Once an area is selected, they distribute themselves in several sub-populations or groups [20]. Sub-populations are generated by several elements that share the same location. In the hunt, no territorial clashes have been observed. Each group usually ignores other sub-populations except that group which obtains the best preys [20].

In each group, goatfish play two different roles: the chaser and the blocker [20], [21]. Only one member of the group can become a chaser by either starting or continue a hunt while the remaining individuals will become blockers. When the hunting begins, preys will try to escape among corals and hide into cracks. The blockers strategically surround the corals to avoid the escape routes. Goatfish do not share prey. By the effect of the hunting process, preys can approach to a blocker fish so close that it turns in a new chaser [20], [21]. Roles are flexible and interchangeable, individuals may act as chasers on some occasions and as blockers in others, the exchange of roles happens even in the same hunt [20]. Finally, once an area has been thoroughly exploited (by hunting all preys), the group changes its position to another sector [20].

3 Yellow Saddle Goatfish Algorithm (YSGA)

Collective hunting behavior model of yellow saddle goatfish has been considered as a guideline to develop a novel bio-inspired optimization algorithm. The approach considers the hunting area as the search space. In the approach, individuals emulate a set of fish. Initially, the proposed model designates a number of groups or sub-populations according to the spatial distribution of the individuals. The algorithm contemplates two distinct categories of search agents (fish): chasers and blockers. In each sub-population one fish assumes the role of chaser while the rest are considered blockers. Depending on the category, each element is undergone by a set of different evolutionary operations which emulate the different collaborative behaviors that are present in the natural hunting process. The fitness value represents the relative success that an element experiment during the hunting process.

The hunting model considers five different steps: (3.1) Initial stage, (3.2) Chaser fish, (3.3) Blocker fish, (3.4) Exchange of roles and (3.5) Change of zone.

3.1 Initial stage

A population $\boldsymbol{P}$ of m goatfishes (individuals) $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \dots, \boldsymbol{p}_m\}$ is randomly generated and uniformly distributed within the boundaries $b^{high}$and $b^{low}$of the n-dimensional search space, where m is the population size and $\boldsymbol{p}_i \in \boldsymbol{P}$ is the vector of decision variables defined as $\boldsymbol{p_i} = \{p_i^1,\ p_i^2, \dots, p_i^n\}$. The initialization is formulated as follows:

$$p_i^j = rand \cdot \left(b_j^{high} - b_j^{low}\right) + b_j^{low} \qquad (1)$$
$$i = 1, 2, \dots, m; \quad j = 1, 2, \dots, n,$$

where $rand$ is a random number among [0,1].

The yellow saddle goatfish tends to form groups for collaborative hunting [19]. Under this assertion, the entire population $\boldsymbol{P}$ is divided into clusters or sub-populations to model this behavior. The main objective is to build groups within a spatial neighborhood to start the cooperative search. **Figure 1** illustrates the partition of the population into k clusters where every cluster $\boldsymbol{c}_k$ has a chaser fish $\boldsymbol{\Phi}_l$ at the center and the blocker fish $\boldsymbol{\varphi}_g$ swims around it. In concordance with the biological model [21], once the k clusters have been generated, they will not change during the evolution process in terms of size and number of elements.

Clustering algorithms have been classified into hierarchical and partitional [30], [31]. Several hierarchical [32] and partitional [33] clustering methods have been proposed in the literature [34]. However, YSGA uses the K-means algorithm [35] for clustering due to its simplicity, efficiency and easy implementation [34].

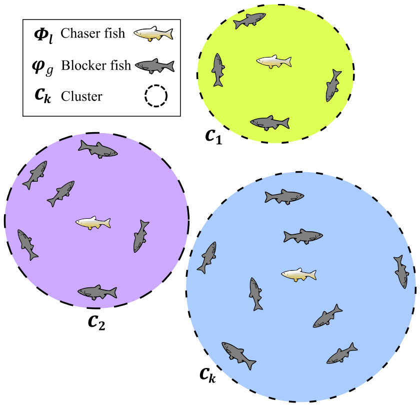
$\boldsymbol{\Phi}_l$ Chaser fish
$\boldsymbol{\varphi}_g$ Blocker fish
$\boldsymbol{c}_k$ Cluster
$\boldsymbol{c}_1$
$\boldsymbol{c}_2$
$\boldsymbol{c}_k$

Figure 1. Role identification and partition of the goatfish population $\boldsymbol{P}$ into k clusters.

K-means is one of the most popular clustering algorithms and has been widely used for dividing consistently elements [31]. This method groups a data set into k number of sub-sets $\{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k\}$ called clusters and calculates the mean μ_l of each cluster $\boldsymbol{c}_l$ such that the sum of the squared error of every data point to its nearest cluster's mean is minimized [36]-[38]. Generally, Euclidean metric is used to calculate the distance between data points and the mean of the cluster [39]. Considering the population of goatfishes $\boldsymbol{P}$ as the data set; the squared error between μ_l and the set of data points $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_h\}$ in cluster $\boldsymbol{c}_l$ is defined as:

$$e(\boldsymbol{c}_l) = \sum_{\boldsymbol{p}_g \in \boldsymbol{c}_l} \|\boldsymbol{p}_g - \mu_l\|^2 \qquad (2)$$

$$g = 1, 2, \ldots, h; \quad l = 1, 2, \ldots, k$$

In this case, h is determined according to k-means algorithm and its value can be different for every cluster $\boldsymbol{c}_l$. The purpose of k-means is to minimize the objective function given by the sum of the squared error over all k clusters, this function is denoted as:

$$E(\boldsymbol{C}) = \sum_{l=1}^{k} e(\boldsymbol{c}_l) \qquad (3)$$

3.2 Chaser fish

Every group of goatfishes has only a chaser fish $\boldsymbol{\Phi}_l \in \boldsymbol{P}$ who leads the hunt [20], see **Figure 1**. The selection of the particle who represents the chaser fish is according to its fitness value. For each cluster, the particle who has the best fitness value must be the closest to the solution. Thus, it is selected to play the role of the chaser fish.

In the hunting, preys will try to escape among corals and will attempt to hide in crevices. The strategy of the chaser fish is to insert its barbels into a crevice where the prey is hiding and search randomly, in different cracks, in case the prey has moved [20], [21]. This behavior is modeled with Lévy flight which is a class of non-Gaussian random processes that uses a Lévy stable distribution to generate random walks [40].

The chaser fish will try to find crevices where prey may be hiding by changing its position with a random walk. The new location of the chaser fish is calculated as:

$$\begin{gathered} \boldsymbol{\Phi}_l^{t+1} = \boldsymbol{\Phi}_l^{t} + \alpha \oplus L\acute{e}vy(\beta) \\ 0 < \beta \leq 2 \end{gathered} \tag{4}$$

The new and current position of the chaser fish are $\boldsymbol{\Phi}_l^{t+1}$ and $\boldsymbol{\Phi}_l^{t}$ respectively. The step size is defined by α, in this approach $\alpha = 1$. The product $\oplus$ means entry-wise multiplications. The parameter β is known as Lévy index and controls the shape of the probability distribution, specially the tail [41]. For $\beta = 1$ the probability distribution function takes the shape of the Cauchy distribution, on the other hand, for $\beta = 2$ the probability distribution corresponds to a Gaussian distribution [42].

A smaller value of β produces longer jumps, since the tail of the distribution is also higher [41]. On the other hand, a bigger value of β provokes shorter jumps since the tail of the distribution is also smaller. This means that α as well as β also allow to regulate the step size of the perturbation. Although several approaches use α to control the step size, there exist recently a tendency to use β as a control strategy [42]- [44]. In this paper, we use β as a control strategy for the step size. Thus, α is not considered in this approach to influence the step size for this reason α is set to 1. In YSGA β increases linearly from 1.99 to 2, the objective is to decrease the length of the steps through the generations, this mechanism leads to a Gaussian distribution which favors the exploitation as iterations increase. Under such circumstances, the chaser tends to exploit its search space since it is the element closest to a pray. But occasionally, chasers tend to explore the space of crevices where the pray may be hidden [20]. In this context, the β value is delimited in a small range (from 1.99 to 2) to assure small steps. With this values, the exploitation process is promoted since the step size gets smaller over iterations. On the other hand, the exploration process of the chaser is also considered, since the randomness of the Lévy flight occasionally generates larger steps. This mechanism avoids falling into a local optimum. **Figure 2** shows how step size decreases as β increases over the course of iterations and it can also be observed how longer steps occur occasionally.

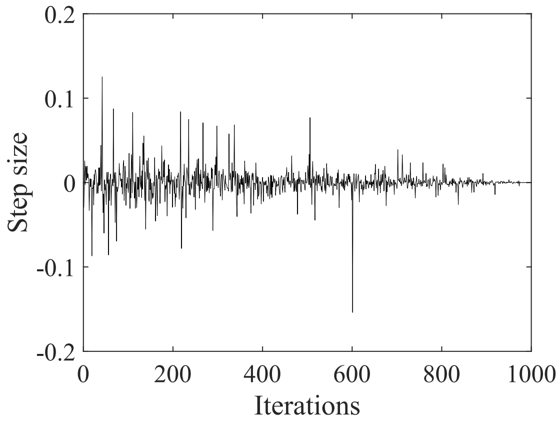

Figure 2. Step size evolution over the course of iterations.

The value of β is calculated as:

$$\beta = 1.99 + \frac{0.001t}{t_{max}/10} \tag{5}$$

where t is the current generation and t_{max} is the maximum number of iterations. Essentially, Lévy flight implements random walks by generating random steps from a Lévy distribution [43], [45]. Since each group usually ignores other sub-populations except that group which obtains the best preys [20], this behavior is modeled as follows:

$$S = \alpha \oplus levy(\beta) \sim \alpha\left(\frac{u}{|v|^{1/\beta}}\right)(\boldsymbol{\Phi}_l^t - \boldsymbol{\Phi}_{best}^t) \tag{6}$$

here, S is the random step and $\boldsymbol{\Phi}_{best}^t$ is the best chaser fish so far among all the clusters. From the normal distribution, u and v are defined as:

$$\begin{gathered} u \sim N(0,\sigma_u^2) \\ v \sim N(0,\sigma_v^2) \end{gathered} \tag{7}$$

where σ_u and σ_v are denoted as follows, consider Γ as the Gamma function:

$$\sigma_u = \left\{\frac{\Gamma(1+\beta)\sin\frac{\pi\beta}{2}}{\Gamma\left(\frac{1+\beta}{2}\right)\beta 2^{(\beta-1)/2}}\right\}^{1/\beta}, \sigma_v = 1 \tag{8}$$

Under these assertions, the new position of the chaser fish given in **Equation (4)** can be rewritten as:

$$\boldsymbol{\Phi}^{t+1}_l = \boldsymbol{\Phi}_l^t + S \tag{9}$$

Equation (9) is valid for every chaser fish $\boldsymbol{\Phi}_l$ corresponding to every cluster $\boldsymbol{c}_l$ except for the global best $\boldsymbol{\Phi}_{best}$. If the new position of the best chaser fish is calculated with **Equation (9)** the subtraction ($\boldsymbol{\Phi}_i^t - \boldsymbol{\Phi}_{best}^t$) shown in **Equation (6)** will be zero, hence S will be zero and the position of the best chaser fish will not change. Thus, the new position of the best chaser fish is calculated as:

$$\boldsymbol{\Phi}_{best}^{t+1} = \boldsymbol{\Phi}_{best}^{t} + S' \tag{10}$$

where $S'$is defined as follows:

$$S' = \alpha\left(\frac{u}{|v|^{1/\beta}}\right) \tag{11}$$

3.3 Blocker fish

Once the chaser fish has been chosen for each cluster, the remaining goatfishes become blockers, see **Figure 1**. The hunting strategy of the blocker fish $\boldsymbol{\varphi}_g \in \boldsymbol{P}$ is to surround the corals to block escape routes of preys while the chaser fish try to hunt the prey [20], [21]. To model this behavior, a logarithmic spiral is applied to the movement of the blocker fish. Since chaser fish is the closest to the prey, the behavior of the blocker fish is to follow a spiral path around the chaser goatfish. **Figure 3** shows the spiral movement of the blocker goatfish, notice that each blocker fish has its own spiral path which is changing over the course of iterations.

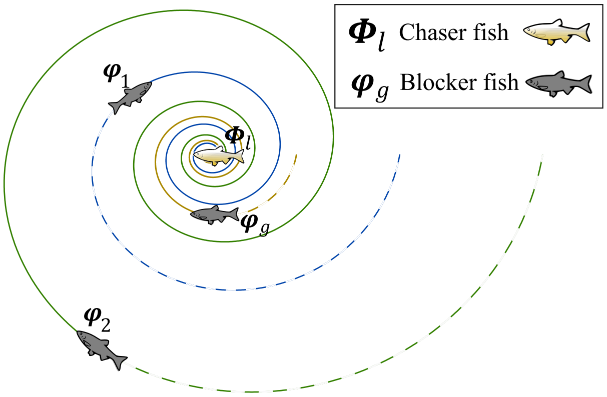

Figure 3. Spiral path of blocker fish.

The new position of the blocker fish $\boldsymbol{\varphi}_g^{t+1}$ is calculated according to a logarithmic spiral defined as follows:

$$\boldsymbol{\varphi}_g^{t+1} = D_g \cdot e^{b\rho} \cdot \cos 2\pi\rho + \boldsymbol{\Phi}_l \tag{12}$$

where ρ is a random number among $[a, 1]$ which defines how close is the blocker to the chaser fish considering the spiral path. To intensify exploitation, a is linearly decreased from

-1 to -2 as iterations increase. The position of the chaser fish is defined when $\rho =- 2$. Hence, the closest position to the chaser fish occurs when $\rho \approx- 2$ and the farthest when $\rho = 1$, see **Figure 4**.

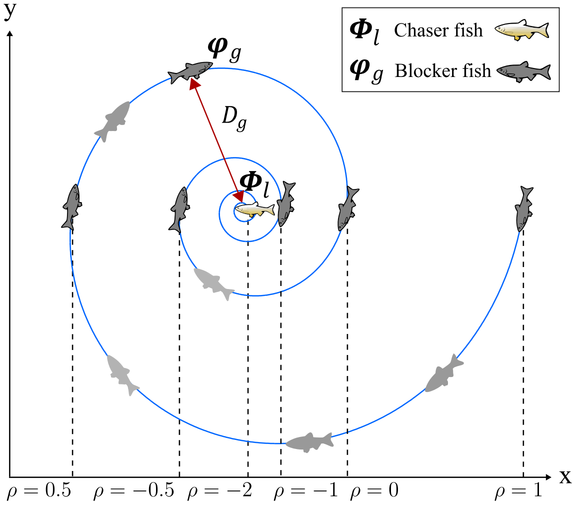

Figure 4. Some of the possible positions reached by a blocker fish depending on the value of ρ.

Since the range tends to $[-2, 1]$ over iterations, ρ increase the possibility of approaching -2 over the course of generations with a minimum probability of reaching it (due to randomness) which is just as expected, **see Figure 5**. Under this schema, blocker goatfish tend to exploit more, proportionally to the number of iterations.

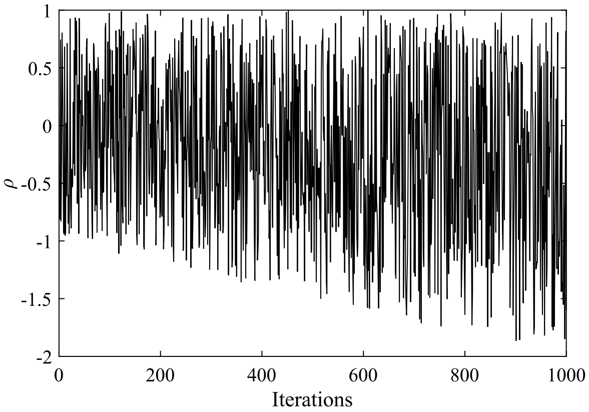

Figure 5. Different values of ρ over the course of iterations.

The parameter b in **Equation (12)** is a constant which determines the shape and the direction of the spiral, in this approach $b = 1$. The term D_g is the distance between the current position of the blocker fish $\boldsymbol{\varphi}_g^t$ and the chaser fish $\boldsymbol{\Phi}_l$ in the cluster $\boldsymbol{c}_l$ described as:

$$\begin{gathered} D_g = \left| r \cdot \boldsymbol{\Phi}_l - \boldsymbol{\varphi}_g^t \right| \\ \left\{ \boldsymbol{\Phi}_l, \boldsymbol{\varphi}_g^t \right\} \in \boldsymbol{c}_l \end{gathered} \qquad (13)$$

where r is a random number among $[-1,1]$ which disturbs the distance D_g avoiding over-exploitation and promoting diversity in the trajectories that define the movement of blocker goatfish.

In nature, several species of animals exhibit similar behavior patterns in specific processes. For example, humpback whales [46], [47], moths [48] or goatfishes [19-21] develop a circular movement which gradually is reduced. This pattern is used for these biological entities for different proposes such as navigating, mating or preying. A logarithmic function is the simplest form to model this movement [11,13]. This movement has demonstrated to explore a certain area of the search space effectively. For this reason, the logarithmic movement has been adopted for many metaheuristic algorithms based on very different metaphors. It is important to remark that these similarities take place only in specific mechanisms since in general terms the biological model of different species are quite distinct.

3.4 Exchange of roles

The main objective of a blocker fish is to avoid the scape of a prey [21]. During the hunting process, a prey moves in the hunting area. Therefore, the blocker fish closest to the prey will lead the hunt becoming the new chaser fish and the current chaser fish will become a blocker fish [19,21]. This process is called the exchange of roles. This behavior is modeled assuming that the element with the best fitness value of a group becomes the chaser fish. In **Figure 6**, a role exchange is illustrated. In this example, it is assumed that the blocker fish $\boldsymbol{\varphi}_2$ has achieved better fitness value than the other goatfish in the current iteration t. This means that $\boldsymbol{\varphi}_2$ has moved closer to the prey becoming a chaser fish. Hence, the role exchange is carried out in iteration $t+1$.

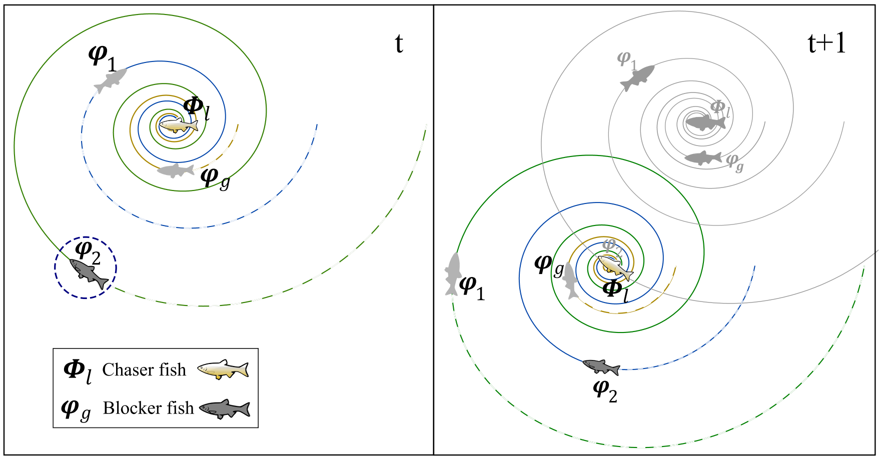

***Figure 6**:* Exchange of roles

YSGA verifies possible role exchanges based on the performance of the search agents. The proximity of a goatfish to the prey determines its role. The goatfish closest to the prey must be the chaser. Closeness is defined according to the fitness value of every goatfish. At the end of each generation, a role update is executed if necessary. For each cluster, if a blocker fish has better fitness value than the chaser, means that it has found a better solution, then a role exchange is performed.

Although blockers do not have memory to retain the best individual positions as other swarm algorithms do (like PSO), in YSGA algorithm, chasers maintain a memory to record their best-reached positions, which it makes possible to guide the search strategy in every cluster. The memory of the chasers is not deleted if there is an exchange of roles. This mechanism avoids instability in the algorithm.

The exchange of roles is implemented for each cluster independently. For this reason, this process can be considered a local search in parallel. However, there is interaction between clusters when a change of zone is produced. Besides, the information of the best particle seen so far are also shared among clusters in order to guide the change of zone in the hunting area if it is necessary.

3.5 Change of zone

Once an area has been completely exploited (by hunting all preys), the group changes its position to other sector in order to find new preys [20]. Under such conditions, the YSGA model considers an overexploitation parameter λ. Therefore, for every cluster, if a predefined number of iterations λ has been exceeded without finding a better solution, then it is considered as a successful hunt so a change of area is performed for all the goatfish in the cluster as follows:

$$\boldsymbol{p}^{t+1}_{g} = \frac{\boldsymbol{\Phi}_{best} + \boldsymbol{p}^{t}_{g}}{2} \tag{14}$$

where $\boldsymbol{p}^{t+1}_{g}$ is the new position of a goatfish regardless of whether it is a chaser or blocker. The best solution so far among all the clusters is given by the chaser fish $\boldsymbol{\Phi}_{best}$, $\boldsymbol{p}^{t}_{g}$ is the current position of a goatfish (chaser or blocker) member of the cluster. In **Figure 7**, a change of zone is illustrated. In this example, the chaser $\boldsymbol{\Phi}_1$ has not improved; in that case, the cluster $\boldsymbol{c}_1$ must change to another area considering the position of the global best $\boldsymbol{\Phi}_{best}$. This process avoids to be trapped in local optima.

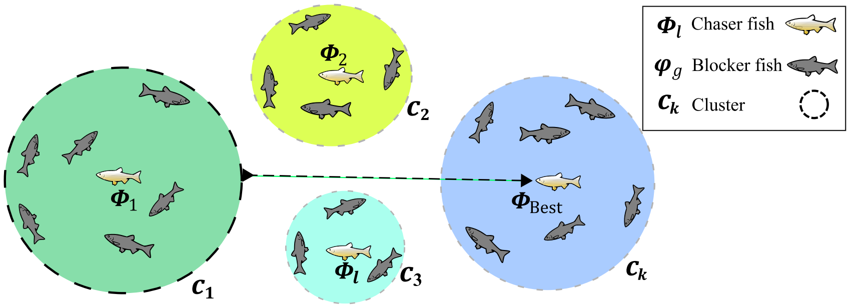

***Figure** 7*. Change of zone of cluster $\boldsymbol{c}_1$.

3.6 Computational procedure

The proposed approach has been implemented as an iterative process in which a sequence of operations are executed. As input data, the algorithm requires the number of goatfish elements m, the number of clusters k, the maximum number of iterations t_{max} and the search space **S**. The goatfish population $\boldsymbol{P}$ is initialized and uniformly distributed over the search space **S** within the predefined limits b^{high} and b^{low} to represent the initial population $\boldsymbol{P}(t)$ where $t = 1$ for the first iteration, also known as the first generation.

After initialization, the fitness value of every goatfish is calculated, the global best $\boldsymbol{\Phi}_{best}$ is identified. Then, k-means algorithm is applied to divide the goatfish population into k clusters $\{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k\}$. For very cluster $\boldsymbol{c}_l$, the goatfish with better fitness value will be identified as the chaser fish $\boldsymbol{\Phi}_l$ of that cluster and the remaining will be considered as the blocker fish $\boldsymbol{\varphi}_g$.

Later, different operators are applied to the search agents: a hunting routine for chaser fish and a blocker routine for blocker goatfish. Each particle moves according to these operators. The hunting routine changes the position of the chaser goatfish by generating random steps according to a Lévy distribution while the blocker routine changes the position of the blocker fish by following a logarithmic spiral path around the chaser.

Afterward, the fitness value of every goatfish is calculated. In each cluster, if a blocker has better fitness value than the chaser, then a change of role is executed, the current chaser will become a blocker and the blocker will become the new chaser fish. On the other hand, if any of the chaser goatfish (considering all chasers in all clusters) outperforms the global best, then the global best is updated.

For each iteration and every cluster, if the chaser goatfish has not improved, then a counter q that regulates stagnation is increased until a predefined limit λ is reached. If q becomes bigger than λ, then a change of zone is made, this means that there is no more prey to hunt in that area so it is necessary to search for preys in a different zone.

The whole process is repeated until the maximum number of iterations is reached. The YSGA algorithm summary is explained in the pseudo-code of **Table 1**.

Pseudo-code

```
Input: $m, k, t_{max}$, S
Initialize the goatfish population $\boldsymbol{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_m\}$
Calculate the fitness value of each particle
Identify the global best $\boldsymbol{\Phi}_{best}$
Partition the population $\boldsymbol{P}$ into $k$ clusters $\{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k\}$
Identify the chaser $\boldsymbol{\Phi}_l$ and the blocker $\boldsymbol{\varphi}_g$ fish for every cluster
While ($t < t_{max}$)
```

```
    For every cluster $\boldsymbol{c}_l$
        Execute hunting routine for chaser fish
        Execute blocking routine for blocker fish
        Calculate the fitness value of each goatfish
        If $\boldsymbol{\varphi}_g$ has better fitness value than $\boldsymbol{\Phi}_l$
            exchange roles by updating $\boldsymbol{\Phi}_l$
        End If
        If $\boldsymbol{\Phi}_l$ has better fitness value than $\boldsymbol{\Phi}_{best}$
            update $\boldsymbol{\Phi}_{best}$
        End If
        If the fitness value of $\boldsymbol{\Phi}_l$ has not improved
            $q \leftarrow q + 1$
        End If
        If $q > \lambda$
            execute a routine for changing zone
            $q \leftarrow 0$
        End If
    End For
    $t \leftarrow t + 1$
End While
Output $\boldsymbol{\Phi}_{best}$
```

Table 1. Pseudo-code of the YSGA algorithm

4. Comparative analysis of YSGA with other similar approaches

Although YSGA maintains some similarities with other metaheuristic algorithms, there are significant differences among them. In this section, the main characteristics among the proposed YSGA method and other similar metaheuristic approaches are contrasted.

The Particle Swarm Optimization algorithm (PSO) is considered the most popular swarm approach currently in use. Similar to YSGA, PSO emulates the behavior of animals in groups [5]. However, the mechanisms employed by both methods are entirely different. In PSO, a single operator of attraction to the best individual positions is considered to model the swarm movements. On the other, YSGA implements a set of operations that modify the search positions of two different individual categories.

The Swallow swarm optimization algorithm (SSO) is a metaheuristic method which models the swarm movements presented in swallow groups. SSO implements three kinds of search agents: explorer particles, aimless particles and leader particles [49]. The positions of these agents are updated by applying an operator which considers only the particle velocity in their trajectories. In YSGA, chaser and blocker elements do not use representations of velocity in their operators. Instead of this, YSGA employs perturbations produced by a Lévy distribution and logarithmic patterns as central operations to modify the position of its search agents. Under the SSO method, explorer and leader particles build sub-populations, like chasers and blockers generate clusters in the YSGA. In SSO, not all search agents are considered to integrate these groups whereas in the proposed YSGA all its elements participate in the cluster operation. In both algorithms, such groups are produced based on different criteria. In SSO, sub-populations are assembled based on fitness values, while in YSGA clusters are constructed according to spatial information.

The Whale Optimization Algorithm (WOA) is another scheme with which the YSGA presents some similarities. WOA reproduces the hunting strategy of humpback whales [11]. In WOA, individuals emulate whales that search for the optimal solution individually, not in groups as in the YSGA algorithm. Although WOA updates the position of its search agents by a logarithmic spiral, as the YSGA does with the blocker elements, both algorithms implement other completely different operators. For example, in WOA, particles are undergone by operations such as randomness, attraction to the best individual and probabilistic decisions. In YSGA, elements are modified according to operations such as Lévy perturbations, position reset, changes of roles and clustering. In WOA, all operators are applied to every single individual in the population while in YSGA the operations are conducted depending on the element category (chaser or blocker).

The moth-flame optimization algorithm (MFO) is a metaheuristic technique which considers the navigation method of moths as a metaphor [13]. Similar to YSGA, MFO considers two search agents: flames and moths. However, such agents do not produce groups or sub-populations as YSGA. In MFO, individuals are categorized as flames and moths without changing their roll during the evolution process. On the other hand, in YSGA, blockers and chasers continually change their category in the optimization process. Both algorithms, YSGA and MFO, implement very different operators in their search strategies with the exception of the logarithmic perturbation which is the only common operation.

The Selfish Herd Optimizer (SHO) is a recent metaheuristic algorithm based on the selfish behavior in herds when they are exposed to a predation risk [12]. Similar to YSGA, in the SHO algorithm, two search agents are considered to emulate the animal behavior: the members of a selfish herd (the prey) and a pack of hungry predators. However, both methods define different mechanisms to implement their search strategies. Their main differences are the structure of the population (complete for SHO and subgroups in YSGA), movement operators (attraction for SHO, Lévy perturbations and logarithmic patterns for YSGA) and roll of the individuals (fixed for SHO and variable for YSGA).

Finally, the Artificial Fish Swarm Algorithm (AFSA) is a swarm technique inspired by the collective movement of the fish according to their social behaviors [50]. AFSA emulates social behaviors of a generic fish such as grouping, reproduction and scape of predators. On the other hand, YSGA reproduces the hunting model of yellow saddle goatfish. Since both methods model different biological mechanisms, they implement very different operators to define their search strategy. Therefore, the only similarity is the use of a metaphor that considers fish as a biological entity.

5 Experimental results

The purpose of this section is to analyze the performance of the proposed method on numeric optimization problems. Furthermore, the results of YSGA are contrasted with some other similar optimization algorithms by using the comprehensive set of benchmark functions. The results are validated by a statistical analysis of the experimental data.

In the experiments, three different indicators are considered: The Average Best-so-far (**AB**) solution, the Median Best-so-far (**MB**) solution and the Standard Deviation (**SD**) of the best-so-far solutions. Every benchmark function is tested on each algorithm 30 times which generates 30 results that represent the best-found solutions. The mean of these 30 outcomes is calculated to obtain **AB**. Similarly, the median of these 30 results is determined to generate **MB** and the standard deviation of the 30 results is estimated to acquire **SD** of the best-so-far solutions. Indicators **AB** and **MB** evaluate the accuracy of the solutions while **SD** the dispersion and therefore the robustness of the algorithm.

The experimental results are divided into five sub-sections. In the first section (5.1), the performance of YSGA is evaluated with regard to some of its important parameters. In the second section (5.2), the results of the YSGA method in comparison with other similar approaches are analyzed. In the third section (5.3), a comparative study among the algorithms examining different population sizes is accomplished. In the fourth section (5.4), the ability of all algorithms to converge is analyzed. Finally, in the fifth section (5.5), the performance analysis over five popular optimization problems is conducted.

5.1 Performance evaluation of YSGA with regard to its important parameters

Without taking into account the total population m and the maximum number of iterations t_{max}, which are common parameters to set in all metaheuristic methods, we consider the number of clusters k as the only parameter to be configured in the YSGA algorithm. Parameters α and β of Lévy flight are considered internal so it is not recommended for the user to modify them. The overexploitation parameter λ is not considered in the setting configuration, since its value is also fixed $\lambda = 10$ without experimenting changes. In this sub-section we analyze the behavior of the proposed algorithm considering the different settings of the parameters k and λ. The study for α and β are not considered in view of their effects have been extensively exanimated [40]-[43].

In YSGA, the entire population $\boldsymbol{P}$ is divided into k clusters or sub-populations. For this propose the approach uses the k-means method. The number of sub-populations k affects the expected performance of the proposed YSGA method. Several techniques have been implemented to select an appropriate number of clusters [51]. Most of these methods require a great amount of iterations and complex procedures in order to obtain a satisfactory association. Therefore, to maintain the YSGA model as simple as possible the number of clusters k is considered fixed. Its value has been determined through a sensibility analysis.

For the sake of simplicity, only a set of representative functions ($f_{10}(\mathbf{x})$, $f_{11}(\mathbf{x})$, $f_{17}(\mathbf{x})$, f_{22} $(\mathbf{x})$, $f_{25}(\mathbf{x})$ and $f_{27}(\mathbf{x})$) have been considered in the tuning process. Such functions has been selected due to its high complexity. In the experiments, all the functions operate in 30 dimension. In our analysis, the parameter k is evaluated, while the other parameters remain fixed to their default values. In the experiments, the values of k vary from 3 to 6 whereas the performance of the YSGA is examined. To minimize the stochastic effect of the algorithm, each benchmark function is executed independently a total of 30 times. As a termination criterion, the maximum number of iterations (t_{max}) is set to 1000. In all simulations, the population size m is fixed to 50 individuals. The results are registered in **Table 2**. These values represent The Average Best-so-far (**AB**) solution, the Median Best-so-far (**MB**) solution and the Standard Deviation (**SD**) of the best-so-far solutions, obtained in terms of the number of clusters k. The best results are highlighted in boldface.

FUNCTION		$k = 3$	$k = 4$	$k = 5$	$k = 6$
$f_{10}(\mathbf{x})$	**AB**	1.29E+02	**8.38E+01**	9.43E+01	1.05E+02
	MB	4.27E+01	**1.10E+01**	1.38E+01	5.49E+01
	SD	2.15E+02	1.40E+02	**1.32E+02**	1.56E+02
$f_{11}(\mathbf{x})$	**AB**	**6.67E-01**	**6.67E-01**	**6.67E-01**	**6.67E-01**
	MB	**6.67E-01**	**6.67E-01**	**6.67E-01**	**6.67E-01**
	SD	6.01E-07	**1.18E-07**	1.31E-07	1.81E-07
$f_{17}(\mathbf{x})$	**AB**	2.89E+01	**1.44E+01**	2.24E+01	2.29E+01
	MB	1.49E+01	**0.00E+00**	6.97E+00	9.95E+00
	SD	3.61E+01	**2.48E+01**	2.52E+01	2.87E+01
$f_{22}(\mathbf{x})$	**AB**	**0.00E+00**	**0.00E+00**	7.66E-250	5.56E-218
	MB	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	SD	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
$f_{25}(\mathbf{x})$	**AB**	**2.90E+01**	**2.90E+01**	**2.90E+01**	**2.90E+01**
	MB	**2.90E+01**	**2.90E+01**	**2.90E+01**	**2.90E+01**
	SD	3.28E-06	3.63E-06	3.97E-06	**1.54E-06**
$f_{27}(\mathbf{x})$	**AB**	**2.90E+01**	**2.90E+01**	**2.90E+01**	**2.90E+01**
	MB	**2.90E+01**	**2.90E+01**	**2.90E+01**	**2.90E+01**
	SD	2.29E-15	**0.00E+00**	2.64E-15	**0.00E+00**

Table 2. Optimal results of the multimodal and composite test sample functions considering $k = 3$, $k = 4$, $k = 5$, and $k = 6$ in 30 dimensions.

From **Table 2**, we can conclude that the proposed YSGA method with k=4 maintains the best performance. Under this configuration, the algorithm obtains the best results in 6 from 6 functions.

To determine the optimal value of overexploitation parameter λ, a sensibility analysis has been also conducted. Under this analysis, the performance of the proposed algorithm is evaluated considering different values for λ. In the experiment, the parameter λ is varied assuming the values 5, 10, 15 and 20 whereas the parameter k remain fixed at 4. The statistical results obtained by the YSGA algorithm using different values of λ are presented in **Table**

3. From the results, it is clear that the YSGA method with $\lambda = 10$ outperforms the other parameter configurations. Under this configuration, the algorithm obtains the best results in 6 of the 6 functions. However, if another value of λ is used, it results in a bad performance.

FUNCTION		$\lambda = 5$	$\lambda = 10$	$\lambda = 15$	$\lambda = 20$
$f_{10}(\mathbf{x})$	**AB**	9.49E+01	**8.38E+01**	9.74E+01	9.48E+01
	MB	3.05E+01	**3.10E+01**	5.49E+01	8.58E+01
	SD	1.48E+02	**1.40E+02**	1.49E+02	2.79E+02
$f_{11}(\mathbf{x})$	**AB**	0.24E-01	**6.67E-01**	0.44E-01	0.62E-01
	MB	**6.67E-01**	**6.67E-01**	**6.67E-01**	**6.67E-01**
	SD	4.71E-07	1.98E-07	2.18E-06	**1.16E-07**
$f_{17}(\mathbf{x})$	**AB**	3.05E+01	**1.44E+01**	2.92E+01	1.94E+01
	MB	9.45E+00	**3.51E+00**	5.47E+00	6.01E+00
	SD	3.99E+01	**2.68E+01**	3.68E+01	3.64E+01
$f_{22}(\mathbf{x})$	**AB**	1.22E-280	**2.29E-287**	2.37E-280	9.33E-256
	MB	**0.00E+00**	**0.00E+00**	0.00E+00	0.00E+00
	SD	**0.00E+00**	**0.00E+00**	0.00E+00	0.00E+00
$f_{25}(\mathbf{x})$	**AB**	3.21E+02	**2.90E+01**	1.45E+02	4.10E+02
	MB	**2.90E+01**	**2.90E+01**	**2.90E+01**	**2.90E+01**
	SD	4.45E-06	4.27E-06	4.74E-06	**1.21E-06**
$f_{27}(\mathbf{x})$	**AB**	**2.90E+01**	**2.90E+01**	**2.90E+01**	**2.90E+01**
	MB	**2.90E+01**	**2.90E+01**	**2.90E+01**	**2.90E+01**
	SD	2.64E-15	**1.87E-15**	**1.87E-15**	2.64E-15

Table 3. Optimal results of the multimodal and composite test sample functions considering $\lambda = 5$, $\lambda = 10$, $\lambda = 15$, and $\lambda\ = 20$ in 30 dimensions.

In general, the experimental results shown in **Tables 2** and **3** suggest that a proper combination of the parameter values of k and λ can improve the performance of the proposed method and the quality of solutions. In this experiment, we can conclude that the best parameter set is composed by the following values: k=4 and $\lambda = 10$.

5.2 Comparison with other optimization approaches

YSGA is evaluated by comparing the results against other related optimization techniques. In the experiments, the following approaches have been considered: The Gray Wolf Optimization (GWO), the Particle Swarm Optimization (PSO), the Synchronous-Asynchronous Particle Swarm Optimization (SA-PSO), the Artificial Bee Colony (ABC), the Crow Search Algorithm (CSA), the Natural Aggregation Algorithm (NAA), the Moth Flame Optimization (MFO), the Artificial Fish Swarm Algorithm (AFSA) and the Whale Optimization Algorithm (WOA). All these techniques represent some of the most popular bio-inspired methods currently in used. The set of 27 benchmark functions has been solved using these algorithms including YSGA and the results have been compared.

To demonstrate its effectiveness, the proposed methodology is tested on a set of 27 benchmark functions ($f_1 - f_{27}$) described in detail in **Appendix A** (**Tables AI**, **AII** and

AIII). These functions are classified into three main categories according to its characteristics: unimodal, multimodal and composite. **Table AI** contains the unimodal test functions while **Table AII** the multimodal and **Table AIII** the composite functions.

In the experiments, the total population m has been set to 50 individuals and the maximum number of iterations t_{max} has been set to 1000 for all the algorithms involved. The test functions have been implemented in $n = 30$, $n = 50$ and $n = 100$ dimensions. Each experiment is repeated considering 30 independent executions to verify the consistency of the results.

The parameters used in each method have been configured according to the reported values in which their best performance are achieved. **Table 4** shows the configuration of their parameters.

Settings configuration		
	YSGA	Number of clusters $k = 4$
[9]	GWO	Parameters configured according to the reference
[52]	PSO	$c_1 = 2$, $c_2 = 2$, velocity clamp $= 2$, minW $= 0.2$, maxW $= 0.9$
[6]	SA-PSO	$c_1 = 2$, $c_2 = 2$, velocity clamp $= 2$, minW $= 0.2$, maxW $= 0.9$, number of groups $c = 5$, group size $= 10$, initial distance 50% search space
[7]	ABC	Onlooker 50%, employees 50%, acceleration coefficient upper bound $= 1$, abandonment limit $L =$ round(0.6*dimensions*population)
[8]	CSA	$p_a = 0.25$
[10]	NAA	Number of shelters $N^s = 8$, shelter capacity $Cp^s = 6$, the probability of local search $Cr_{global} = 0.9$, scaling factor $\delta = 1$, generalized movement factor $\alpha = 1.2$
[13]	MFO	Parameters configured according to the authors reference
[50]	AFSA	Visual=2.85, δ=9, Step=1, Number of lookups=10
[11]	WOA	Parameters configured according to the authors reference

Table 4. Parameter configuration according to the reported references.

Several experiments have been conducted for comparing the performance of the proposed YSGA. The tests have been divided into Unimodal functions (4.2.1), Multimodal functions (4.2.2) and composite functions (4.2.3).

5.2.1 Results of unimodal test functions

In **Table AI** described in **Appendix A**, from f_1 to f_5 are unimodal functions, this kind of functions have a single optimum. The experimental results are listed in **Tables 5**, **6** and **7** considering the performance of each algorithm for each test function in 30, 50 and 100 dimensions respectively. Indicators **AB**, **MB** and **SD** illustrate the accuracy of the results. The best outcomes are highlighted in boldface.

		YSGA	GWO	PSO	SA-PSO	ABC	CSA	NAA	MFO	AFSA	WOA
$f_1(\mathbf{x})$	**AB**	**1.40E-182**	1.53E-72	8.74E-01	3.84E-07	1.29E-03	3.98E-05	6.36E-58	5.24E+00	9.43E-02	2.45E-25
	MB	**2.60E-188**	1.15E-73	1.60E-09	4.18E-07	1.21E-03	3.92E-05	1.95E-60	6.13E-07	9.75E-02	7.36E-29
	SD	**0.00E+00**	3.53E-72	4.79E+00	1.13E-07	5.74E-04	2.12E-05	2.11E-57	1.07E+01	1.81E-02	1.16E-24
$f_2(\mathbf{x})$	**AB**	**1.80E-172**	5.69E-71	7.80E+02	8.06E-05	4.91E-02	3.68E-01	1.15E-55	3.90E+02	4.48E+00	9.93E-24
	MB	**6.40E-185**	7.00E-72	6.00E+02	3.87E-05	4.55E-02	3.39E-01	1.81E-59	2.00E+02	4.50E+00	1.85E-27
	SD	**0.00E+00**	2.30E-70	7.27E+02	8.04E-05	2.32E-02	2.24E-01	5.48E-55	5.01E+02	8.18E-01	3.86E-23
$f_3(\mathbf{x})$	**AB**	**0.00E+00**	1.06E-235	6.86E-18	1.32E-11	1.73E-03	3.64E-09	5.57E-50	8.87E-17	5.88E-06	5.96E-36
	MB	**0.00E+00**	1.49E-241	1.60E-20	1.64E-12	1.21E-03	1.36E-09	5.41E-63	1.14E-19	6.20E-06	8.13E-42
	SD	**0.00E+00**	**0.00E+00**	3.57E-17	2.14E-11	1.55E-03	5.61E-09	3.05E-49	3.84E-16	1.81E-06	2.62E-35
$f_4(\mathbf{x})$	**AB**	**2.20E-170**	3.20E-68	1.68E+06	8.02E-03	4.33E+01	1.68E+03	6.93E-52	1.23E+06	3.79E+04	1.59E-20
	MB	**3.00E-182**	7.49E-69	1.51E+06	2.88E-03	3.52E+01	1.38E+03	4.00E-55	8.60E+05	3.63E+04	3.22E-24
	SD	**0.00E+00**	7.80E-68	1.16E+06	1.11E-02	2.54E+01	1.15E+03	3.74E-51	1.17E+06	8.12E+03	5.36E-20
$f_5(\mathbf{x})$	**AB**	**7.50E-174**	5.11E-69	3.02E+04	9.26E-05	2.38E+00	1.39E+01	4.43E-56	2.22E+04	3.26E+02	2.92E-23
	MB	**7.00E-185**	3.23E-70	2.58E+04	5.92E-05	1.98E+00	1.01E+01	2.83E-57	1.29E+04	3.21E+02	4.16E-25
	SD	**0.00E+00**	1.34E-68	2.40E+04	9.28E-05	1.24E+00	9.93E+00	1.40E-55	2.87E+04	6.22E+01	9.65E-23

Table 5. Optimal results of the unimodal test functions considering 30 dimensions.

From **Table 5**, it can be seen how the YSGA algorithm has a better performance than the other methods. Indicators **AB**, **MB,** and **SD** demonstrate that the proposed approach stands out from the rest of the techniques for all the test functions. Although the GWO algorithm has the same **SD** value as the YSGA in function f_3, the **AB** and **MB** values are better in YSGA which means that both have the same dispersion but YSGA has better quality in the solution. From **Table 5**, it is also observed that PSO and MFO have the worst performance while the other methods achieve different quality values.

To test the scalability of the YSGA algorithm, the experiments have been repeated but now considering 50 dimensions instead of 30. The results of this test are exposed in **Table 6**.

		YSGA	GWO	PSO	SA-PSO	ABC	CSA	NAA	MFO	AFSA	WOA
$f_1(\mathbf{x})$	**AB**	**4.31E-172**	3.96E-87	2.80E+01	3.88E-04	4.20E+00	1.20E-04	1.29E-57	1.96E+01	4.01E-01	2.66E-22
	MB	**3.03E-185**	2.29E-88	2.62E+01	3.97E-04	4.23E+00	1.16E-04	2.75E-62	2.62E+01	4.03E-01	8.43E-28
	SD	**0.00E+00**	1.82E-86	1.94E+01	1.25E-04	9.17E-01	4.78E-05	7.04E-57	2.07E+01	6.20E-02	1.45E-21
$f_2(\mathbf{x})$	**AB**	**3.05E-162**	6.17E-86	3.46E+03	3.49E-02	1.76E+02	1.29E+00	2.05E-54	2.67E+03	8.85E+01	5.14E-23
	MB	**8.58E-184**	2.85E-86	2.90E+03	1.58E-02	1.77E+02	8.99E-01	3.08E-60	2.75E+03	8.92E+01	1.85E-26
	SD	**1.68E-161**	7.57E-86	2.32E+03	4.03E-02	4.25E+01	9.36E-01	1.12E-53	2.12E+03	1.64E+01	1.82E-22
$f_3(\mathbf{x})$	**AB**	**0.00E+00**	**0.00E+00**	1.01E-13	7.29E+09	1.89E-01	8.76E-10	6.84E-47	4.54E-09	1.06E-05	2.52E-36
	MB	**0.00E+00**	**0.00E+00**	1.52E-14	8.11E+05	1.71E-01	6.40E-10	7.52E-53	5.24E-10	1.11E-05	2.11E-42
	SD	**0.00E+00**	**0.00E+00**	2.07E-13	1.26E+10	1.11E-01	9.00E-10	2.74E-46	1.19E-08	2.95E-06	8.32E-36
$f_4(\mathbf{x})$	**AB**	**6.74E-167**	3.10E-82	1.61E+07	6.14E-01	1.58E+05	7.83E+03	2.10E-54	1.01E+07	5.62E+05	1.08E-17
	MB	**2.05E-178**	8.24E-83	1.64E+07	6.16E-01	1.54E+05	7.35E+03	1.33E-56	8.44E+06	5.62E+05	2.90E-23
	SD	**0.00E+00**	6.07E-82	7.26E+06	4.03E-01	3.09E+04	4.22E+03	9.13E-54	7.25E+06	7.17E+04	5.87E-17
$f_5(\mathbf{x})$	**AB**	**2.70E-170**	4.33E-84	1.87E+05	1.53E-02	8.12E+03	5.35E+01	1.08E-54	7.18E+04	4.19E+03	2.66E-20
	MB	**7.86E-181**	8.16E-85	1.68E+05	5.06E-03	7.73E+03	4.98E+01	7.85E-58	6.45E+04	4.13E+03	1.11E-24
	SD	**0.00E+00**	7.30E-84	9.33E+04	1.84E-02	2.18E+03	2.79E+01	3.16E-54	5.30E+04	6.53E+02	1.02E-19

Table 6. Optimal results of the unimodal test functions considering 50 dimensions.

From **Table 6**, it can be appreciated that the YSGA algorithm outperforms the other methods. The best solutions are registered in indicators **AB** and **MB** corresponding to the proposed approach while the dispersion **SD** is also better in the YSGA algorithm than the other techniques. On the other hand, the GWO algorithm reaches equivalent results to the YSGA in the test function f_3.

Additionally, the experiments have been repeated but now considering 100 dimensions which suppose a more significant challenge. The results of this test are reported in **Table 7**.

		YSGA	**GWO**	**PSO**	**SA-PSO**	**ABC**	**CSA**	**NAA**	**MFO**	**AFSA**	**WOA**
$f_1(\mathbf{x})$	**AB**	**2.55E-162**	5.13E-37	1.23E+02	7.09E-01	2.04E+02	8.83E-05	2.36E-122	5.44E+01	1.49E+00	8.22E-50
	MB	**7.28E-176**	2.44E-37	1.31E+02	1.91E-01	2.06E+02	8.62E-05	5.82E-125	5.23E+01	1.50E+00	2.86E-56
	SD	**1.28E-161**	8.64E-37	4.31E+01	4.31E+01	1.52E+01	1.49E-05	8.19E-122	3.78E+01	1.16E-01	4.43E-49
$f_2(\mathbf{x})$	**AB**	**1.62E-162**	9.32E-35	3.12E+04	8.60E+00	2.52E+04	2.95E+00	1.52E-117	1.32E+04	6.15E+02	7.28E-46
	MB	**1.88E-172**	5.84E-35	2.99E+04	9.17E+00	2.52E+04	2.58E+00	1.94E-123	1.14E+04	6.23E+02	1.78E-55
	SD	**8.89E-162**	9.30E-35	7.94E+03	7.94E+03	2.07E+03	1.18E+00	8.30E-117	7.75E+03	5.29E+01	3.78E-45
$f_3(\mathbf{x})$	**AB**	**3.13E-233**	9.17E-179	2.18E-02	6.18E+46	1.16E+00	1.33E-10	1.42E-82	1.29E-04	7.44E-06	1.31E-67
	MB	**0.00E+00**	1.85E-201	5.48E-07	6.86E+42	1.23E+00	7.55E-11	5.34E-97	2.74E-07	7.51E-06	8.24E-77
	SD	**0.00E+00**	**0.00E+00**	9.31E-02	9.31E-02	2.93E-01	2.16E-10	7.71E-82	6.79E-04	1.81E-06	7.17E-67
$f_4(\mathbf{x})$	**AB**	**6.56E-158**	2.96E-31	1.88E+08	3.59E+02	9.06E+07	3.65E+04	6.40E-115	8.79E+07	5.94E+06	1.06E-44
	MB	**1.76E-172**	2.59E-31	1.97E+08	2.02E+02	9.16E+07	3.50E+04	2.44E-120	8.80E+07	5.99E+06	5.64E-53
	SD	**3.30E-157**	2.11E-31	4.70E+07	4.70E+07	6.79E+06	1.30E+04	3.41E-114	3.33E+07	4.74E+05	5.75E-44
$f_5(\mathbf{x})$	**AB**	**2.61E-159**	4.71E-33	1.20E+06	2.24E+01	1.10E+06	1.01E+02	1.02E-115	5.92E+05	2.75E+04	3.02E-46
	MB	**3.16E-173**	2.75E-33	1.25E+06	2.24E+01	1.12E+06	9.48E+01	1.63E-121	5.33E+05	2.75E+04	5.30E-54
	SD	**1.43E-158**	6.04E-33	3.86E+05	3.86E+05	9.76E+04	4.12E+01	5.61E-115	3.48E+05	3.38E+03	1.40E-45

Table 7. Optimal results of the unimodal test functions considering 100 dimensions.

From **Table 7**, it can be observed that the YSGA algorithm has better performance than the other methods in all the benchmark functions. Indicators **AB**, **MB,** and **SD** prove that the proposed approach records the best results. Although the GWO algorithm has the same **SD** value as the YSGA for function f_3, the **AB** and **MB** values are better in YSGA which means that both have the same dispersion but YSGA has better quality in the solution. From **Table 5**, it is also detected that the contending techniques GWO, PSO, SA-PSO, ABC, MFO and AFSA get worst results as the number of dimensions increases. From **Tables 5**, **6** and **7** it can be concluded that the proposed method has a good performance in high-dimensional problems.

The results obtained in **Tables 5**, **6** and **7** are examined for validation in a statistical context by applying a non-parametric analysis identified as Wilcoxon test [54], [55]. The examination measures the difference between two methods that are compared. For both algorithms in comparison, a 5% (0.05) of the significance level for the Average Best-so-far (**AB**) is considered by the Wilcoxon test. Under this schema, the proposed method is compared with the other techniques one by one. For comparison, the following group configuration is set: YSGA vs. GWO, YSGA vs. PSO, YSGA vs. SA-PSO, YSGA vs. ABC, YSGA vs. CSA, YSGA vs. NAA, YSGA vs. MFO, YSGA vs. AFSA, and YSGA vs. WOA. In Wilcoxon test, a null hypothesis is established in which the discrepancy between the two approaches is not perceptible. In contrast, an alternative hypothesis is also accepted in which the difference between the two methods is significant.

However, the possibility of falling into an error Type I [56] increases as the number of comparisons involved in the Wilcoxon test is incremented. To avoid such error, the significance level must be reconsidered using the Bonferroni correction [57]. Hence, the significance level is recalculated by Bonferroni correction generating a value of 3.3E-04.

The p-values resulting from the Wilcoxon test, considering the Bonferroni correction, are visualized in **Tables 8**, **9** and **10**. After applying the Wilcoxon test over the **AB** data from **Table 5**, the p-values generated were reported in **Table 8**. In the same way, the p-values generated by the Wilcoxon test over the **AB** data from **Tables 6** and **7** were listed in **Table 9** and **10** respectively. In the test, the null hypothesis is rejected for all the p-values that are lower than 3.3E-04, which means that both methods in comparison differ significantly in their results. To simplify the analysis, the symbols ▲, ▼, and ► are used to identify the outcomes of the Wilcoxon test. The symbol ▲ means that the YSGA approach has better performance than the compared method. The symbol ▼ indicates that the YSGA method performs worse than its competitor and the symbol ► implies that the Wilcoxon analysis could not identify any difference between the YSGA algorithm and its contender.

From **Table 8**, it can be observed that the column of the YSGA vs. GWO has a p-value (equal to 3.0E-11) in the first row (corresponding to the test function f_1) which is lower than the significance value of 3.3E-04 calculated with the Bonferroni correction. This means that the null hypothesis is rejected. Hence, there is a difference between both algorithms which supports the results obtained in **Table 5** for that function. Under this assumption, the symbol ▲ is placed on the right side of the p-value indicating that the YSGA has better performance than the GWO. Examining **Table 8**, it is clear that all the p-values in all the columns show evidence against the null hypothesis. After the analysis, it can be concluded that the YSGA algorithm performs better than the other methods for all the unimodal test functions in 30 dimensions.

	YSGA vs GWO	**YSGA vs PSO**	**YSGA vs SA-PSO**	**YSGA vs ABC**	**YSGA vs CSA**	**YSGA vs NAA**	**YSGA vs MFO**	**YSGA vs AFSA**	**YSGA vs WOA**
$f_1(\mathbf{x})$	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	2.46E-11▲	1.18E-11▲	3.59E-12▲
$f_2(\mathbf{x})$	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	2.74E-11▲	1.98E-11▲	1.50E-11▲
$f_3(\mathbf{x})$	1.2E-12▲	1.2E-12▲	1.2E-12▲	1.2E-12▲	1.2E-12▲	1.2E-12▲	1.54E-13▲	2.07E-13▲	1.16E-12▲
$f_4(\mathbf{x})$	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	2.76E-11▲	2.13E-11▲	1.03E-11▲
$f_5(\mathbf{x})$	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	1.91E-11▲	9.61E-13▲	1.77E-11▲

Table 8. p-values of Wilcoxon-Bonferroni test comparing YSGA vs GWO, YSGA vs PSO, YSGA vs SA-PSO, YSGA vs ABC, YSGA vs CSA, YSGA vs NAA, YSGA vs MFO, YSGA vs AFSA, and YSGA vs WOA over the **AB** values from **Table 5**, $n = 30$.

The Wilcoxon test for the unimodal functions considering 50 dimensions is displayed in **Table 9**. In this case, it is visible that all the p-values in the columns YSGA vs PSO, YSGA vs SA-PSO, YSGA vs ABC, YSGA vs CSA, YSGA vs NAA, YSGA vs MFO, YSGA vs AFSA, and YSGA vs WOA validate that the proposed approach outperforms its competitors for all the functions. Besides, despite the tie in function f_3 of **Table 6** for the YSGA and GWO algorithms, the Wilcoxon analysis could determine discrepancy between the YSGA and the GWO, thus the symbol ▲ is placed on the right side of the p-value indicating that the YSGA has better performance than the GWO.

	YSGA vs GWO	**YSGA vs PSO**	**YSGA vs SA-PSO**	**YSGA vs ABC**	**YSGA vs CSA**	**YSGA vs NAA**	**YSGA vs MFO**	**YSGA vs AFSA**	**YSGA vs WOA**
$f_1(\mathbf{x})$	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	7.30E-12▲	2.45E-12▲	6.82E-12▲
$f_2(\mathbf{x})$	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	1.22E-11▲	2.81E-11▲	5.16E-12▲
$f_3(\mathbf{x})$	1.2E-12▲	3.3E-11▲	1.2E-12▲	1.2E-12▲	1.2E-12▲	1.2E-12▲	2.28E-13▲	9.40E-13▲	5.39E-13▲
$f_4(\mathbf{x})$	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.99E-12▲	1.47E-11▲	1.32E-11▲
$f_5(\mathbf{x})$	1.2E-12▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	2.84E-11▲	1.32E-11▲	9.39E-12▲

Table 9. p-values of Wilcoxon-Bonferroni test comparing YSGA vs GWO, YSGA vs PSO, YSGA vs SA-PSO, YSGA vs ABC, YSGA vs CSA, YSGA vs NAA, YSGA vs MFO, YSGA vs AFSA, and YSGA vs WOA over the **AB** values from **Table 6**, $n = 50$.

The results of the Wilcoxon test for the unimodal functions in 100 dimensions are presented in **Table 10**. From these results, it is evident that all the p-values in the columns YSGA vs GWO, YSGA vs PSO, YSGA vs SA-PSO, YSGA vs ABC, YSGA vs CSA, YSGA vs NAA, YSGA vs MFO, YSGA vs AFSA, and YSGA vs WOA confirm that the proposed approach performs better than its contenders for all the test functions.

	YSGA vs GWO	**YSGA vs PSO**	**YSGA vs SA-PSO**	**YSGA vs ABC**	**YSGA vs CSA**	**YSGA vs NAA**	**YSGA vs MFO**	**YSGA vs AFSA**	**YSGA vs WOA**
$f_1(\mathbf{x})$	8.6E-07▲	4.4E-13▲	1.4E-08▲	2.7E-12▲	8.4E-10▲	1.9E-12▲	2.11E-11▲	1.93E-11▲	1.08E-12▲
$f_2(\mathbf{x})$	7.5E-10▲	1.3E-11▲	6.5E-12▲	4.6E-10▲	9.1E-12▲	7.6E-11▲	1.93E-11▲	2.89E-11▲	5.31E-12▲
$f_3(\mathbf{x})$	1.0E-10▲	7.5E-13▲	4.1E-12▲	3.0E-12▲	5.5E-13▲	2.4E-11▲	5.44E-13▲	2.92E-13▲	1.17E-11▲
$f_4(\mathbf{x})$	2.8E-06▲	5.4E-11▲	1.9E-12▲	8.8E-11▲	1.0E-13▲	2.9E-12▲	2.08E-12▲	2.04E-11▲	1.43E-11▲
$f_5(\mathbf{x})$	1.7E-06▲	4.8E-12▲	4.9E-12▲	1.9E-12▲	2.8E-14▲	4.9E-11▲	9.65E-12▲	8.73E-12▲	4.61E-12▲

Table 10. p-values of Wilcoxon-Bonferroni test comparing YSGA vs GWO, YSGA vs PSO, YSGA vs SA-PSO, YSGA vs ABC, YSGA vs CSA, YSGA vs NAA, YSGA vs MFO, YSGA vs AFSA, and YSGA vs WOA over the **AB** values from **Table 7**, $n = 100$.

From **Tables 8**, **9** and **10**, it can be resolved that the statistical test exhibits evidence supporting the results registered in **Tables 5**, **6** and **7** exposing that the YSGA algorithm outperforms its opponents.

After the Wilcoxon analysis of the proposed approach against the other methods and its performance over the unimodal benchmark functions considering 30, 50 and 100 dimensions, it can be concluded that the YSGA algorithm is functional in high-dimensional optimization problems and achieve better performance than its opponents.

5.2.2 Results of multimodal test functions

In **Appendix A**, the **Table AII** exhibits from f_6 to f_{23} the multimodal functions which are characterized by having multiple local optima that increase exponentially as the number of dimensions gets bigger. Therefore, this kind of functions are more difficult to optimize, so this test challenges the algorithms to demonstrate their ability to find the global optimum under these conditions. The experimental results are shown in **Tables 11**, **12** and **13** considering the performance of each method for each test function in 30, 50 and 100 dimensions respectively. Indicators **AB**, **MB** and **SD** illustrate the accuracy of the results. The best results are highlighted in boldface.

		YSGA	GWO	PSO	SA-PSO	ABC	CSA	NAA	MFO	AFSA	WOA
$f_6(\mathbf{x})$	**AB**	2.49E-15	1.17E-14	1.86E+01	1.49E-03	8.75E-01	3.11E+00	**1.18E-16**	1.25E+01	1.52E+01	3.53E-10
	MB	3.55E-15	1.42E-14	1.99E+01	1.32E-03	7.90E-01	3.13E+00	**0.00E+00**	1.52E+01	1.85E+01	2.32E-12
	SD	1.66E-15	3.25E-15	5.06E+00	9.52E-04	3.81E-01	8.95E-01	**6.49E-16**	7.11E+00	6.36E+00	1.15E-09
$f_7(\mathbf{x})$	**AB**	**5.23E-08**	4.08E-06	2.02E+03	5.35E-02	3.83E+02	6.33E-01	3.23E-02	8.44E+02	1.13E+01	7.91E-05
	MB	**6.62E-15**	3.18E-06	1.93E+03	2.23E-02	3.82E+02	6.34E-01	1.81E-02	2.64E+02	1.14E+01	1.36E-06
	SD	**2.85E-07**	3.85E-06	1.72E+03	5.57E-02	8.26E+01	3.06E-01	4.65E-02	1.22E+03	4.89E+00	3.62E-07
$f_8(\mathbf{x})$	**AB**	**2.43E+01**	2.67E+01	2.11E+05	7.95E+01	6.90E+02	8.82E+01	6.22E+01	8.93E+04	6.75E+01	3.78E+01
	MB	**2.43E+01**	2.62E+01	1.70E+05	8.22E+01	6.92E+02	7.78E+01	4.51E+01	5.56E+04	6.37E+01	3.55E+01
	SD	**2.24E-01**	8.32E-01	1.42E+05	6.55E+00	1.76E+02	5.84E+01	5.05E+01	9.62E+04	1.33E+01	1.20E+01
$f_9(\mathbf{x})$	**AB**	**2.35E-99**	8.96E-40	9.04E+02	2.07E-03	9.10E+10	1.64E+02	6.77E-46	4.53E+02	1.34E+27	5.58E-14
	MB	**8.64E-106**	5.94E-40	8.50E+02	2.29E-03	1.64E+06	1.69E+02	5.15E-47	5.00E+02	1.93E+26	1.56E-14
	SD	**1.20E-98**	9.45E-40	3.35E+02	1.11E-03	4.86E+11	3.97E+01	2.25E-45	2.00E+02	2.81E+27	1.06E-13
$f_{10}(\mathbf{x})$	**AB**	8.38E+01	1.17E+02	1.25E+38	**1.44E+01**	8.21E+78	4.61E+02	2.06E+02	1.96E+63	7.18E+42	1.36E+03
	MB	3.10E+01	3.90E+01	6.05E+19	**1.30E+01**	4.12E+77	2.87E+02	6.98E+01	1.70E+37	1.99E+41	9.75E+02
	SD	1.40E+02	1.69E+02	6.85E+38	**1.51E+01**	2.64E+79	4.25E+02	2.99E+02	1.07E+64	2.68E+43	1.01E+03
$f_{11}(\mathbf{x})$	**AB**	**6.67E-01**	**6.67E-01**	2.85E+04	7.87E-01	1.48E+02	9.13E-01	1.64E+00	3.26E+04	7.75E+00	9.40E-01
	MB	**6.67E-01**	**6.67E-01**	9.58E+01	8.27E-01	1.46E+02	8.48E-01	6.96E-01	8.26E+01	7.36E+00	9.36E-01
	SD	**1.98E-07**	4.38E-07	5.68E+04	1.05E-01	6.04E+01	1.85E-01	1.60E+00	7.96E+04	1.56E+00	1.34E-01
$f_{12}(\mathbf{x})$	**AB**	-1.13E+03	-9.42E+02	-1.07E+03	-1.01E+03	-8.28E+02	-1.01E+03	-1.06E+03	-1.04E+03	-1.04E+03	**-1.17E+03**
	MB	-1.13E+03	-9.44E+02	-1.08E+03	-9.77E+02	-8.15E+02	-9.91E+02	-1.06E+03	-1.04E+03	-1.04E+03	**-1.17E+03**
	SD	2.86E+01	6.84E+01	3.25E+01	4.90E+01	4.72E+01	4.12E+01	3.25E+01	3.49E+01	1.87E+01	**8.00E-01**
$f_{13}(\mathbf{x})$	**AB**	**8.74E+00**	9.64E+00	1.46E+01	1.62E+01	1.19E+01	8.83E+00	9.91E+00	1.48E+01	1.41E+01	1.02E+01
	MB	**8.78E+00**	8.79E+00	1.35E+01	1.69E+01	1.20E+01	8.83E+00	9.96E+00	1.13E+01	1.42E+01	1.00E+01
	SD	**3.03E-01**	4.09E-01	4.25E+00	1.71E+00	6.37E-01	4.48E-01	7.39E-01	7.39E+00	9.60E-01	7.86E-01
$f_{14}(\mathbf{x})$	**AB**	**1.03E-01**	1.50E-01	1.31E+00	9.00E-01	2.20E+00	8.03E-01	2.27E-01	5.20E+00	2.40E+00	1.97E+00
	MB	**9.99E-02**	1.50E-01	7.00E-01	9.00E-01	2.24E+00	8.50E-01	2.00E-01	3.65E+00	2.44E+00	1.99E+00
	SD	**1.83E-02**	5.09E-02	2.43E+00	1.00E-01	2.42E-01	1.38E-01	6.92E-02	4.18E+00	4.14E-01	6.15E-01
$f_{15}(\mathbf{x})$	**AB**	**7.16E+01**	1.37E+06	1.05E+02	8.12E+01	1.57E+07	8.83E+01	8.29E+01	8.82E+01	6.60E+03	5.17E+03
	MB	**7.10E+01**	1.00E+06	8.48E+01	8.42E+01	1.63E+07	8.55E+01	8.26E+01	8.73E+01	6.28E+03	3.46E+03
	SD	**1.38E+00**	9.43E+05	5.06E+01	5.32E+00	3.68E+06	1.31E+01	3.38E+00	4.07E+00	2.24E+03	7.38E+03
$f_{16}(\mathbf{x})$	**AB**	**1.25E+01**	1.25E+05	1.13E+02	8.62E+01	1.56E+06	1.15E+02	1.03E+02	9.43E+01	6.85E+03	1.76E+03
	MB	**1.25E+01**	1.25E+05	1.08E+02	8.63E+01	1.59E+06	1.14E+02	1.02E+02	9.43E+01	6.76E+03	1.19E+03
	SD	**3.35E+00**	8.53E+04	1.27E+01	4.26E+00	3.71E+05	5.74E+00	6.99E+00	7.29E+00	2.17E+03	1.45E+03
$f_{17}(\mathbf{x})$	**AB**	1.44E+01	**6.96E-01**	1.40E+02	5.55E+00	2.20E+02	2.05E+01	1.16E+01	1.47E+02	1.79E+02	1.86E+01
	MB	3.51E+00	**4.10E-01**	1.44E+02	5.18E+00	2.21E+02	1.84E+01	1.04E+01	1.48E+02	1.76E+02	1.85E+01
	SD	2.68E+01	2.44E+00	3.33E+01	**1.52E+00**	1.25E+01	7.80E+00	9.34E+00	3.78E+01	1.95E+01	6.11E-01
$f_{18}(\mathbf{x})$	**AB**	**2.00E+00**	3.03E+00	1.17E+02	2.82E+01	**2.00E+00**	7.82E+07	2.07E+00	**2.00E+00**	**2.00E+00**	**2.00E+00**
	MB	**2.00E+00**	2.06E+00	9.00E+00	2.82E+01	**2.00E+00**	1.03E+07	2.57E+00	**2.00E+00**	**2.00E+00**	**2.00E+00**
	SD	**0.00E+00**	2.53E+00	2.32E+02	1.02E+01	**0.00E+00**	1.47E+08	1.81E-01	**0.00E+00**	**0.00E+00**	**0.00E+00**
$f_{19}(\mathbf{x})$	**AB**	**2.00E+00**	9.73E+00	1.36E+02	2.82E+01	**2.00E+00**	8.47E+07	2.05E+00	2.07E+00	**2.00E+00**	**2.00E+00**
	MB	**2.00E+00**	2.49E+00	4.35E+01	2.82E+01	**2.00E+00**	2.36E+07	2.01E+00	2.01E+00	**2.00E+00**	**2.00E+00**
	SD	**0.00E+00**	3.12E+01	2.26E+02	1.03E+01	**0.00E+00**	1.95E+08	1.04E-01	3.57E-01	**0.00E+00**	**0.00E+00**
$f_{20}(\mathbf{x})$	**AB**	**0.00E+00**	1.57E-11	**0.00E+00**	1.86E-17	9.29E-05	5.69E-12	4.39E-13	9.81E-08	2.75E-07	7.55E-11
	MB	**0.00E+00**	1.49E-11	**0.00E+00**	4.92E-18	7.19E-05	1.72E-12	1.25E-18	6.77E-07	3.03E-07	8.83E-12
	SD	**0.00E+00**	1.04E-11	**0.00E+00**	2.41E-17	8.87E-05	9.92E-12	1.63E-12	4.17E-07	1.51E-07	1.16E-08
	AB	**-3.00E+01**	**-3.00E+01**	-2.82E+01	-4.92E+14	-1.91E+01	-2.98E+01	**3.00E+01**	**-3.00E+01**	-2.97E+01	-2.89E+01

$f_{21}(\mathbf{x})$	**MB**	**-3.00E+01**	**-3.00E+01**	-2.81E+01	-5.06E+14	-1.92E+01	-2.99E+01	**3.00E+01**	**-3.00E+01**	-2.98E+01	-2.92E+01
	SD	**6.33E-04**	9.57E-04	1.73E+00	4.58E+13	1.31E+00	4.16E-01	3.15E-03	1.47E+00	1.27E-01	1.04E+00
$f_{22}(\mathbf{x})$	**AB**	**2.29E-287**	1.88E-153	4.49E-13	5.07E-12	2.24E-03	3.17E-12	5.03E-36	3.86E-02	2.49E-08	7.77E-46
	MB	**0.00E+00**	4.37E-158	3.51E-15	4.38E-12	1.91E-03	1.51E-14	4.30E-44	1.75E-08	8.35E-09	2.69E-60
	SD	**0.00E+00**	1.00E-152	1.98E-12	4.94E-12	1.68E-03	1.35E-11	2.65E-35	2.12E-01	3.69E-08	4.22E-45
$f_{23}(\mathbf{x})$	**AB**	**0.00E+00**	**0.00E+00**	2.10E+01	9.04E-03	7.95E-01	9.54E-02	4.92E-03	1.81E+01	1.14E+00	1.25E-07
	MB	**0.00E+00**	**0.00E+00**	8.63E-03	9.87E-03	8.05E-01	9.53E-02	1.32E-15	1.86E-02	1.14E+00	0.00E+00
	SD	**0.00E+00**	**0.00E+00**	4.55E+01	8.65E-03	9.20E-02	3.65E-02	1.09E-02	3.68E+01	2.58E-02	5.27E-07

Table 11. Optimal results of the multimodal test functions considering 30 dimensions.

From **Table 11**, it can be appreciated that the YSGA algorithm achieves better performance than its competitors in functions f_7, f_8, f_9, f_{13}, f_{14}, f_{15}, f_{16} and f_{22}. On the other hand, the NAA method reaches better indicators than the YSGA in function f_6 while the SA-PSO obtains the best results in function f_{10} and WOA in function f_{12}. In the case of function f_{17}, although the best **SD** was achieved by the SA-PSO approach, the best **AB** and **MB** indicators were obtained by the GWO algorithm which means that the SA-PSO only has the best dispersion of the data but the GWO accomplish the best results in terms of quality solution and also its dispersion has the best second value. Consequently, it can be concluded that for function f_{17}, the best performance was obtained by the GWO algorithm. Regarding function f_{11}, the YSGA and the GWO reach the same values in **AB** and **MB** but the YSGA achieves better dispersion than the GWO which means that the YSGA is a more consistent algorithm in terms of the generated results. Similarly, in function f_{21} the YSGA, the GWO, the NAA and the MFO obtain the same values in **AB** and **MB** but the YSGA achieves better dispersion. Finally, in function f_{18}, the YSGA, the ABC, the MFO, the AFSA, and the WOA reach the same values in **AB**, **MB** and **SD** indicators. Likewise, in f_{19} the YSGA, the ABC, the AFSA and the WOA reach the same values in **AB**, **MB** and **SD** indicators while the YSGA and the PSO tie in f_{20} as well as the YSGA and the GWO in f_{23}.

To test the adaptability of the YSGA algorithm, the experiments have been repeated but now considering 50 dimensions. The results of this test are informed in **Table 12**.

		YSGA	**GWO**	**PSO**	**SA-PSO**	**ABC**	**CSA**	**NAA**	**MFO**	**AFSA**	**WOA**
$f_6(\mathbf{x})$	**AB**	2.49E-15	1.58E-14	1.99E+01	1.40E-02	8.57E+00	3.67E+00	**0.00E+00**	1.84E+01	2.01E+01	3.70E-11
	MB	3.55E-15	1.42E-14	1.99E+01	1.57E-02	8.60E+00	3.77E+00	**0.00E+00**	1.89E+01	2.02E+01	2.45E-12
	SD	2.12E-15	3.81E-15	2.44E-01	3.18E-03	5.83E-01	8.38E-01	**0.00E+00**	2.63E+00	3.28E-01	1.03E-10
$f_7(\mathbf{x})$	**AB**	**1.05E-14**	1.95E-06	8.25E+03	4.67E+00	2.84E+03	1.99E+00	6.62E-01	3.25E+03	6.98E+01	2.83E-13
	MB	**1.48E-42**	1.04E-06	7.82E+03	5.78E+00	2.93E+03	1.93E+00	2.49E-01	2.95E+03	6.84E+01	5.57E-11
	SD	**5.76E-14**	2.35E-06	3.69E+03	2.24E+00	3.94E+02	6.17E-01	1.04E+00	2.25E+03	1.96E+01	1.01E-12
$f_8(\mathbf{x})$	**AB**	**5.04E-95**	1.44E-48	7.91E+16	2.24E-01	8.58E+32	3.27E+02	5.33E-60	3.15E+05	4.33E+02	5.96E+01
	MB	**2.09E-101**	6.27E-49	2.31E+03	1.15E-01	1.07E+30	3.18E+02	1.52E-61	2.52E+05	4.44E+02	5.76E+01
	SD	**2.63E-94**	2.47E-48	4.33E+17	2.26E-01	3.26E+33	4.27E+01	1.78E-59	2.59E+05	9.16E+01	2.75E+00
$f_9(\mathbf{x})$	**AB**	2.10E+130	3.26E+02	1.09E+114	3.81E+02	2.15E+146	3.04E+16	9.34E+07	8.45E+02	1.89E+55	**3.34E-13**
	MB	8.07E+03	1.28E+02	3.98E+102	3.80E+02	6.48E+144	2.14E+05	4.50E+02	8.02E+02	5.97E+52	**3.83E-14**
	SD	1.15E+131	4.09E+02	5.85E+114	3.73E+02	4.69E+146	1.67E+17	5.11E+08	2.96E+02	9.65E+55	**1.15E-12**
$f_{10}(\mathbf{x})$	**AB**	**6.67E-01**	7.67E-01	3.22E+05	4.21E+00	1.53E+05	2.20E+00	1.75E+01	9.64E+139	9.64E+99	5.90E+18
	MB	**6.67E-01**	7.67E-01	1.88E+05	5.27E+00	1.57E+05	1.85E+00	1.20E+01	1.97E+118	1.97E+98	5.86E+22
	SD	**1.48E-08**	1.70E-07	3.18E+05	3.06E+00	3.40E+04	9.01E-01	1.88E+01	5.28E+140	5.28E+100	1.54E+06
$f_{11}(\mathbf{x})$	**AB**	**0.00E+00**	**0.00E+00**	8.00E+03	2.07E+01	1.54E+03	3.79E+01	**0.00E+00**	2.36E+05	2.75E+01	1.49E+00
	MB	**0.00E+00**	**0.00E+00**	1.00E+04	1.60E+01	1.57E+03	3.75E+01	**0.00E+00**	3.77E+04	2.81E+01	1.47E+00
	SD	**0.00E+00**	**0.00E+00**	8.47E+03	1.08E+01	2.47E+02	1.07E+01	**0.00E+00**	3.64E+05	5.83E+00	8.46E-01

$f_{12}(\mathbf{x})$	**AB**	-1.82E+03	-1.41E+03	-1.74E+03	-1.64E+03	-1.11E+03	-1.65E+03	-1.72E+03	-1.68E+03	-1.62E+03	**-1.94E+03**
	MB	-1.89E+03	-1.42E+03	-1.73E+03	-1.66E+03	-1.09E+03	-1.66E+03	-1.72E+03	-1.68E+03	-1.62E+03	**-1.95E+03**
	SD	4.18E+01	7.93E+01	5.41E+01	5.35E+01	7.25E+01	4.53E+01	4.75E+01	5.41E+01	2.31E+01	**3.44E+00**
$f_{13}(\mathbf{x})$	**AB**	**1.19E+01**	1.69E+01	4.35E+01	4.99E+01	3.45E+01	1.73E+01	2.00E+01	5.34E+01	2.91E+01	1.88E+01
	MB	**1.40E+01**	1.70E+01	3.88E+01	5.17E+01	3.39E+01	1.74E+01	2.00E+01	4.26E+01	2.87E+01	1.87E+01
	SD	**2.06E-01**	4.72E-01	1.65E+01	1.84E+01	3.52E+00	5.84E-01	1.60E+00	2.61E+01	2.15E+00	9.58E-01
$f_{14}(\mathbf{x})$	**AB**	**9.99E-02**	1.80E-01	7.52E+00	1.73E+00	8.49E+00	1.19E+00	3.47E-01	1.46E+01	1.03E+01	1.17E-01
	MB	**9.99E-02**	2.00E-01	1.03E+01	1.70E+00	8.39E+00	1.20E+00	4.00E-01	1.44E+01	1.05E+01	1.34E-01
	SD	**3.36E-12**	4.07E-02	5.61E+00	2.52E-01	6.35E-01	1.58E-01	8.19E-02	3.87E+00	1.17E+00	8.74E-02
$f_{15}(\mathbf{x})$	**AB**	4.76E+03	7.79E+06	6.16E+03	**1.41E+02**	1.16E+08	1.62E+02	1.42E+02	8.36E+05	6.54E+05	1.57E+04
	MB	6.00E+02	7.00E+06	9.88E+02	**1.40E+02**	1.14E+08	1.57E+02	1.43E+02	1.81E+04	6.40E+05	6.60E+03
	SD	6.77E+03	2.36E+06	1.92E+04	**2.24E+00**	2.12E+07	1.68E+01	4.76E+00	3.85E+06	2.20E+05	2.54E+04
$f_{16}(\mathbf{x})$	**AB**	2.16E+02	5.17E+05	1.62E+03	**1.52E+02**	1.09E+08	2.08E+02	1.97E+02	6.92E+07	3.13E+04	3.46E+03
	MB	2.16E+02	5.00E+05	4.87E+02	**1.53E+02**	1.07E+08	2.07E+02	1.79E+02	2.33E+04	3.22E+04	1.64E+03
	SD	3.53E+00	1.52E+05	2.61E+03	**2.50E+00**	2.86E+07	1.00E+01	7.05E+01	1.55E+08	9.94E+03	5.98E+03
$f_{17}(\mathbf{x})$	**AB**	**0.00E+00**	1.79E+00	3.33E+02	4.49E+01	4.73E+02	3.36E+01	4.58E+01	3.06E+02	3.15E+02	1.16E-08
	MB	**0.00E+00**	0.00E+00	3.27E+02	1.83E+01	4.71E+02	3.14E+01	4.34E+01	3.09E+02	3.18E+02	2.84E-14
	SD	**0.00E+00**	6.02E+00	5.69E+01	4.88E+01	1.97E+01	1.15E+01	3.10E+01	4.18E+01	2.40E+01	6.36E-08
$f_{18}(\mathbf{x})$	**AB**	**2.00E+00**	3.41E+03	1.81E+09	1.41E+14	**2.00E+00**	5.46E+18	5.52E+00	**2.00E+00**	3.02E+02	**2.00E+00**
	MB	**2.00E+00**	5.84E+01	2.10E+06	1.41E+14	**2.00E+00**	1.07E+18	4.24E+00	**2.00E+00**	7.81E+02	**2.00E+00**
	SD	**0.00E+00**	1.79E+04	6.57E+09	5.44E-16	7.99E-14	1.04E+19	4.73E+00	**0.00E+00**	7.63E+02	**0.00E+00**
$f_{19}(\mathbf{x})$	**AB**	**2.00E+00**	1.26E+03	4.62E+09	1.41E+14	**2.00E+00**	7.69E+19	7.10E+00	2.23E+00	4.13E+01	**2.00E+00**
	MB	**2.00E+00**	6.81E+01	4.87E+05	1.41E+14	**2.00E+00**	1.21E+17	5.08E+00	2.10E+00	8.88E+00	**2.00E+00**
	SD	**0.00E+00**	4.86E+03	2.50E+10	5.44E-16	2.22E-13	2.64E+20	6.12E+00	1.28E+00	9.01E+01	**0.00E+00**
$f_{20}(\mathbf{x})$	**AB**	**0.00E+00**	2.15E-11	**0.00E+00**	1.85E-12	8.86E-03	3.65E-11	5.09E-07	4.99E-05	1.81E-05	6.80E-09
	MB	**0.00E+00**	3.49E-12	**0.00E+00**	5.58E-13	8.86E-03	2.19E-11	1.10E-08	2.06E-06	1.83E-05	9.53E-07
	SD	**0.00E+00**	3.73E-11	**0.00E+00**	2.60E-12	3.33E-03	4.61E-11	1.85E-06	1.31E-05	4.52E-06	3.13E-08
$f_{21}(\mathbf{x})$	**AB**	**-5.00E+01**	-4.89E+01	-4.48E+01	-7.78E+14	-2.61E+01	-4.90E+01	**-5.00E+01**	-4.77E+01	-5.11E+14	-4.73E+01
	MB	**-5.00E+01**	-4.00E+01	-4.62E+01	-7.71E+14	-2.50E+01	-4.95E+01	**-5.00E+01**	-4.81E+01	-5.14E+14	-4.74E+01
	SD	**2.24E-03**	5.90E+00	3.45E+00	7.73E+13	3.22E+00	1.02E+00	5.40E-03	1.97E+00	5.35E+13	2.69E+00
$f_{22}(\mathbf{x})$	**AB**	**2.69E-288**	6.12E-178	2.50E-08	1.43E-05	4.22E-01	7.62E-08	1.07E-18	7.72E-02	1.65E-02	1.18E-48
	MB	**0.00E+00**	2.29E-182	9.05E-09	2.01E-06	4.42E-01	9.56E-09	5.23E-25	2.68E-06	1.33E-02	5.18E-57
	SD	**0.00E+00**	**0.00E+00**	3.72E-08	2.24E-05	1.18E-01	1.46E-07	5.37E-18	2.94E-01	1.30E-02	5.62E-48
$f_{23}(\mathbf{x})$	**AB**	**0.00E+00**	3.50E-04	9.03E+01	5.76E-03	1.49E+01	1.74E-01	2.63E-03	5.53E+01	3.50E-03	8.03E-04
	MB	**0.00E+00**	**0.00E+00**	9.02E+01	2.21E-05	1.45E+01	1.72E-01	**0.00E+00**	1.49E+00	3.38E-03	0.00E+00
	SD	**0.00E+00**	1.92E-03	8.22E+01	9.95E-03	3.84E+00	3.75E-02	5.37E-03	9.41E+01	8.60E-04	4.22E-03

Table 12. Optimal results of the multimodal test functions considering 50 dimensions.

From **Table 12**, it can be detected that the YSGA algorithm achieves better performance than its competitors in functions f_7, f_8, f_{10}, f_{13}, f_{14}, f_{17}, f_{22} and f_{23}. By contrast, the NAA outperform the YSGA in function f_6 and the WOA in functions f_9 and f_{12} while the SA-PSO reaches the best results in functions f_{15} and f_{16}. In function f_{11}, the YSGA, the GWO and the NAA reach the same values in **AB**, **MB** and **SD** indicators while the YSGA and the PSO tie in f_{20}. Regarding functions f_{18} and f_{19} the YSGA and the ABC reach the same values in **AB** and **MB** but the YSGA achieves better **SD** value than the ABC which means that the YSGA is more consistent in terms of the generated results. Besides, the YSGA, the MFO and WOA tie in f_{18} while in f_{19} the YSGA and the WOA also get the same values. Similarly, the YSGA and the NAA reach the same values **AB** and **MB** in function f_{21} but the YSGA achieves better dispersion than the NAA.

Additionally, the experiments have been repeated but now considering 100 dimensions. The results of this test are exposed in **Table 13**.

		YSGA	GWO	PSO	SA-PSO	ABC	CSA	NAA	MFO	AFSA	WOA
$f_6(\mathbf{x})$	AB	**3.20E-15**	6.94E-14	1.99E+01	3.17E-01	1.88E+01	4.32E+00	2.41E-01	1.94E+01	2.03E+01	2.72E-12
	MB	**3.55E-15**	6.75E-14	2.00E+01	1.86E-01	1.89E+01	4.29E+00	0.00E+00	1.92E+01	2.04E+01	3.55E-13
	SD	**1.43E-15**	5.81E-15	2.26E-01	2.26E-01	3.25E-01	7.39E-01	9.16E-01	4.17E-01	7.19E-02	2.59E-13
$f_7(\mathbf{x})$	AB	**1.11E-24**	1.05E-05	3.39E+04	1.13E+02	2.88E+04	4.06E+00	2.13E+00	1.15E+04	2.63E+02	2.13E-18
	MB	**5.17E-103**	5.04E-06	3.37E+04	1.09E+02	2.90E+04	4.08E+00	7.90E-02	1.08E+04	2.46E+02	4.27E-20
	SD	**6.08E-24**	1.47E-05	8.04E+03	8.04E+03	3.63E+03	6.58E-01	1.08E+01	5.15E+03	6.31E+01	1.00E-18
$f_8(\mathbf{x})$	AB	7.53E-93	9.59E-20	1.01E+75	1.42E+00	7.13E+103	2.93E+17	**2.32E-123**	9.49E+05	1.38E+03	1.21E+02
	MB	5.37E-100	8.36E-20	4.91E+03	1.33E+00	6.34E+95	6.81E+02	**1.81E-125**	7.65E+05	1.15E+03	9.80E+01
	SD	2.66E-92	4.52E-20	5.50E+75	5.50E+75	2.00E+104	1.60E+18	**6.47E-123**	6.83E+05	1.64E+02	2.29E+01
$f_9(\mathbf{x})$	AB	6.67E-01	6.67E-01	4.05E+06	2.80E+02	7.01E+06	5.51E+00	1.22E+01	1.84E+03	1.87E+03	**6.86E-28**
	MB	6.67E-01	6.67E-01	3.60E+06	2.80E+02	6.99E+06	4.73E+00	1.38E+01	1.75E+03	1.98E+03	**1.11E-29**
	SD	5.72E-07	5.72E-07	1.93E+06	1.93E+06	9.25E+05	3.22E+00	8.80E+00	6.02E+02	7.20E+02	**2.26E-27**
$f_{10}(\mathbf{x})$	AB	**1.30E+03**	2.42E+03	3.31E+03	3.36E+03	1.53E+03	3.29E+03	1.72E+03	4.76E+288	3.69E+282	1.25E+14
	MB	**1.33E+03**	2.39E+03	3.31E+03	3.36E+03	1.55E+03	3.27E+03	1.72E+03	1.61E+275	1.56E+274	1.25E+21
	SD	**3.51E+01**	1.30E+02	1.31E+02	1.31E+02	1.77E+02	5.92E+01	4.16E+01	6.55E+04	5.65E+04	3.54E+08
$f_{11}(\mathbf{x})$	AB	**0.00E+00**	**0.00E+00**	5.70E+04	2.90E+01	7.54E+04	2.19E+02	**0.00E+00**	1.72E+06	2.32E+02	1.06E+01
	MB	**0.00E+00**	**0.00E+00**	5.50E+04	2.90E+01	7.58E+04	2.10E+02	**0.00E+00**	1.16E+06	2.35E+02	1.07E+01
	SD	**0.00E+00**	**0.00E+00**	1.97E+04	1.97E+04	5.15E+03	4.72E+01	**0.00E+00**	1.73E+06	2.84E+01	5.76E+00
$f_{12}(\mathbf{x})$	AB	-3.86E+03	-3.91E+02	-3.05E+02	-5.17E+02	-5.11E+02	-3.96E+01	-1.97E+01	-3.31E+03	-3.13E+03	**-3.91E+03**
	MB	-3.87E+03	-3.92E+02	-3.03E+02	-5.17E+02	-5.22E+02	-3.99E+01	-1.95E+01	-3.32E+03	-3.13E+03	**-3.91E+03**
	SD	7.77E-01	7.98E-01	1.17E+02	1.17E+02	6.99E+01	9.06E-01	1.67E+00	7.73E+01	3.32E+01	**2.92E+00**
$f_{13}(\mathbf{x})$	AB	**1.03E-01**	2.57E-01	2.19E+01	4.40E+00	2.99E+01	2.00E+00	2.67E-01	2.69E+02	6.67E+01	4.10E+01
	MB	**9.99E-02**	3.00E-01	2.33E+01	4.40E+00	3.00E+01	2.00E+00	3.00E-01	2.59E+02	6.70E+01	4.09E+01
	SD	**1.83E-02**	5.68E-02	4.23E+00	4.23E+00	1.36E+00	1.67E-01	7.11E-02	1.26E+02	1.87E+00	1.60E+00
$f_{14}(\mathbf{x})$	AB	2.27E+05	3.41E+07	9.44E+07	3.59E+02	1.22E+09	3.86E+02	1.44E+02	3.21E+01	1.70E+01	**1.23E-01**
	MB	1.59E+05	3.46E+07	5.95E+05	3.59E+02	1.22E+09	3.48E+02	1.40E+02	3.21E+01	1.69E+01	**1.07E-01**
	SD	2.35E+05	5.07E+06	1.42E+08	1.42E+08	2.10E+08	1.07E+02	1.86E+01	4.39E+00	1.43E+00	**8.58E-02**
$f_{15}(\mathbf{x})$	AB	2.56E+03	2.21E+06	3.56E+08	3.75E+02	1.84E+09	4.31E+02	**1.79E+02**	7.86E+07	7.21E+06	6.79E+06
	MB	2.00E+03	2.25E+06	4.10E+08	3.75E+02	1.82E+09	4.27E+02	**1.76E+02**	1.99E+07	7.12E+06	6.07E+06
	SD	2.03E+03	2.64E+05	4.27E+08	4.27E+08	3.63E+08	2.23E+01	**1.38E+01**	1.07E+08	1.41E+06	4.98E+05
$f_{16}(\mathbf{x})$	AB	**0.00E+00**	7.36E-01	7.83E+02	1.38E+02	1.21E+03	5.60E+01	3.72E+01	1.46E+08	3.52E+05	2.89E+03
	MB	**0.00E+00**	1.14E-13	7.87E+02	1.38E+02	1.22E+03	5.28E+01	3.88E+01	1.58E+07	3.52E+05	2.21E+03
	SD	**0.00E+00**	2.67E+00	7.37E+01	7.37E+01	4.21E+01	1.47E+01	2.46E+01	2.24E+08	6.28E+04	1.25E+03
$f_{17}(\mathbf{x})$	AB	**0.00E+00**	1.71E-12	**0.00E+00**	1.84E-07	1.37E-01	5.05E-11	1.67E-07	6.82E+02	7.71E+02	2.37E-07
	MB	**0.00E+00**	1.81E-14	**0.00E+00**	1.84E-07	1.36E-01	1.21E-11	8.31E-11	6.79E+02	7.72E+02	1.71E-07
	SD	**0.00E+00**	5.08E-12	**0.00E+00**	2.40E-01	2.74E-02	1.02E-10	4.81E-07	8.61E+01	4.09E+01	6.63E-08
$f_{18}(\mathbf{x})$	AB	**2.00E+00**	1.16E+24	3.95E+37	-7.04E+13	**2.00E+00**	6.98E+55	5.77E+00	6.63E+00	4.08E+01	**2.00E+00**
	MB	**2.00E+00**	1.18E+15	1.55E+28	-7.04E+13	**2.00E+00**	6.09E+51	3.18E+00	2.00E+00	1.22E-131	**2.00E+00**
	SD	**0.00E+00**	5.07E+24	2.16E+38	2.16E+38	2.01E-08	2.87E+56	9.05E+00	1.14E+01	1.57E+02	**0.00E+00**
$f_{19}(\mathbf{x})$	AB	**2.00E+00**	1.81E+22	4.71E+36	-7.04E+13	**2.00E+00**	1.33E+56	4.64E+00	3.27E+01	3.60E+01	**2.00E+00**
	MB	**2.00E+00**	4.09E+17	1.55E+28	-7.04E+13	**2.00E+00**	1.91E+53	3.19E+00	2.00E+00	2.51E+00	**2.00E+00**
	SD	**0.00E+00**	9.06E+22	2.53E+37	2.53E+37	2.06E-08	4.94E+56	3.75E+00	1.14E+02	1.41E+02	**0.00E+00**
$f_{20}(\mathbf{x})$	AB	1.07E-03	2.83E-01	2.48E+01	1.39E+15	1.77E+01	8.50E+01	5.07E-01	7.84E-05	1.09E-04	**6.58E-07**
	MB	9.99E-02	9.99E-01	8.08E+01	1.39E+15	4.29E+01	9.78E+01	5.00E-01	4.40E-06	1.07E-04	**1.93E-05**
	SD	2.09E-03	2.84E-02	5.59E+00	5.59E+00	4.48E+00	3.02E+00	1.81E-02	3.10E-04	2.18E-05	**3.58E-06**
$f_{21}(\mathbf{x})$	AB	**-1.00E+02**	-2.58E+01	-1.04E+00	-7.40E+01	-7.96E+01	-7.56E+00	1.98E+01	-9.12E+01	-8.39E+01	-9.78E+01
	MB	**-1.00E+02**	-2.35E+01	-1.16E+00	-7.40E+01	-8.31E+01	-6.75E+00	9.22E+00	-9.04E+01	-8.39E+01	-9.88E+01
	SD	**0.00E+00**	1.35E-73	8.79E-01	8.79E-01	1.85E+00	5.62E-07	1.08E-22	4.80E+00	1.03E+01	2.93E+00
$f_{22}(\mathbf{x})$	AB	**0.00E+00**	6.76E-04	4.18E+02	5.65E-02	6.88E+02	1.15E-01	**0.00E+00**	3.69E-01	3.21E-01	7.86E-93
	MB	**0.00E+00**	0.00E+00	4.06E+02	5.65E-02	6.92E+02	1.15E-01	**0.00E+00**	7.62E-03	2.86E-01	4.34E-119
	SD	**0.00E+00**	3.71E-03	1.63E+02	1.63E+02	5.02E+01	1.45E-02	**0.00E+00**	9.93E-01	1.13E-01	4.30E-92
$f_{23}(\mathbf{x})$	AB	3.20E-15	6.94E-14	1.99E+01	3.17E-01	1.88E+01	4.32E+00	2.41E-01	1.66E+02	6.45E-03	**7.57E-36**
	MB	3.55E-15	6.75E-14	2.00E+01	1.86E-01	1.89E+01	4.29E+00	0.00E+00	1.83E+02	6.41E-03	**7.34E-39**
	SD	1.43E-15	5.81E-15	2.26E-01	2.26E-01	3.25E-01	7.39E-01	9.16E-01	1.13E+02	7.89E-04	**5.54E-37**

Table 13. Optimal results of the multimodal test functions considering 100 dimensions.

From **Table 13**, it can be visualized that the YSGA algorithm achieves better performance than its competitors in functions f_6, f_7, f_{10}, f_{13}, f_{16}, and f_{21}. By contrast, the NAA outperforms the YSGA in functions f_8 and f_{15} while WOA gets better results in functions f_9, f_{12}, f_{14}, f_{20} and f_{23}. In function f_{11}, the YSGA, the GWO and the NAA reach the same values in **AB**, **MB** and **SD** indicators while the YSGA and the PSO tie in f_{17}. Similarly, the YSGA achieves the same values than the NAA in terms of **AB**, **MB** and **SD** for function f_{22}. Regarding functions f_{18} and f_{19}, the YSGA and the ABC reach the same values in **AB** and **MB** but the YSGA achieves better **SD** value than the ABC which means that the YSGA is more consistent in relation to the produced results. Besides, in functions f_{18} and f_{19}, the YSGA and the WOA reach the same values in **AB**, **MB** and **SD**.

The p-values resulting from the Wilcoxon test, considering the Bonferroni correction, are reported in **Tables 14**, **15** and **16**. After applying the Wilcoxon test over the **AB** data from **Table 11**, the p-values generated were registered in **Table 14**. In the same way, the p-values generated by the Wilcoxon test over the **AB** data from **Tables 12** and **13** were registered in **Table 15** and **16** respectively.

The Wilcoxon test for the multimodal functions considering 30 dimensions is listed in **Table 14**. In this case, it is visible that all the p-values in the columns YSGA vs ABC, YSGA vs CSA, YSGA vs MFO and YSGA vs AFSA validate that the proposed approach outperforms its competitors for all the functions. Regardless of the tie in functions f_{18} and f_{19} from **Table 11** between YSGA, ABC, MFO, AFSA and WOA, the Wilcoxon analysis could determine discrepancy between the algorithms, hence the symbol ▲ is placed on the right side of the p-value indicating that the YSGA has better performance than its contenders. In the same way, despite the tie in function f_{23} of **Table 11** between YSGA and GWO, the Wilcoxon analysis could determine discrepancy between both algorithms, therefore the symbol ▲ is placed on the right side of the p-value indicating that the YSGA has better performance than the GWO. In the case of function f_{20}, in which the YSGA and PSO also ties, the Wilcoxon test shows (in column YSGA vs PSO) that the p-value is higher than the significance value of 3.3E-04. This result indicates that the statistical test could not determine discrepancy between the two approaches thus the symbol ► is placed on the right side of the p-value. Besides, it can be observed that functions f_6, f_{10}, f_{12}, and f_{17} have the symbol ▼ as an indicator that for these functions the NAA, SA-PSO, WOA and GWO (respectively) reached better performances than the YSGA. For the rest of the functions in all comparisons, the Wilcoxon test supports the results recorded in **Table 11** exposing that the proposed approach outperforms its opponents.

	YSGA	**YSGA**	**YSGA**	**YSGA**	**YSGA**	**YSGA**	**YSGA**	**YSGA**	**YSGA**
	vs	**vs**	**vs**	**vs**	**vs**	**vs**	**vs**	**vs**	**vs**
	GWO	**PSO**	**SA-PSO**	**ABC**	**CSA**	**NAA**	**MFO**	**AFSA**	**WOA**
$f_6(\mathbf{x})$	1.2E-06▲	6.6E-12▲	1.3E-11▲	1.4E-11▲	1.4E-11▲	8.2E-06▼	9.90E-13▲	2.81E-12▲	2.27E-12▲
$f_7(\mathbf{x})$	6.0E-07▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	8.41E-12▲	1.39E-12▲	7.62E-12▲
$f_8(\mathbf{x})$	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	1.1E-08▲	3.05E-10▲	2.93E-12▲	1.42E-10▲
$f_9(\mathbf{x})$	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	2.89E-11▲	2.49E-11▲	1.53E-11▲
$f_{10}(\mathbf{x})$	9.5E-05▲	4.5E-07▲	3.0E-11▼	3.0E-11▲	7.7E-06▲	9.1E-06▲	2.91E-11▲	2.10E-11▲	8.43E-11▲

$f_{11}(\mathbf{x})$	1.7E-12▲	8.9E-06▲	3.7E-11▲	3.0E-11▲	3.0E-11▲	2.6E-06▲	8.78E-11▲	9.58E-12▲	7.56E-09▲
$f_{12}(\mathbf{x})$	2.5E-07▲	1.7E-07▲	4.4E-07▲	3.0E-11▲	1.7E-08▲	5.1E-06▲	5.41E-10▲	1.15E-12▲	2.90E-11▼
$f_{13}(\mathbf{x})$	3.0E-11▲	9.6E-05▲	3.0E-11▲	3.0E-11▲	5.0E-08▲	3.4E-08▲	2.89E-11▲	1.04E-12▲	6.60E-11▲
$f_{14}(\mathbf{x})$	3.0E-11▲	6.0E-08▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	1.46E-11▲	1.32E-11▲	1.67E-11▲
$f_{15}(\mathbf{x})$	1.4E-08▲	2.2E-10▲	6.5E-08▲	3.0E-11▲	1.7E-07▲	6.0E-07▲	2.42E-11▲	1.15E-11▲	4.51E-12▲
$f_{16}(\mathbf{x})$	3.0E-11▲	1.1E-06▲	3.3E-06▲	3.0E-11▲	3.6E-06▲	3.3E-11▲	4.28E-12▲	2.31E-11▲	2.82E-11▲
$f_{17}(\mathbf{x})$	8.5E-06▼	2.7E-05▲	1.2E-11▲	1.1E-11▲	5.5E-07▲	6.7E-08▲	1.03E-11▲	1.29E-11▲	9.19E-11▲
$f_{18}(\mathbf{x})$	5.5E-06▲	1.2E-12▲	4.3E-09▲	1.1E-11▲	1.2E-12▲	1.2E-12▲	2.79E-09▲	2.26E-13▲	9.14E-12▲
$f_{19}(\mathbf{x})$	3.3E-07▲	1.2E-12▲	1.4E-11▲	1.1E-11▲	1.2E-12▲	1.2E-12▲	9.25E-11▲	5.93E-13▲	1.64E-11▲
$f_{20}(\mathbf{x})$	2.2E-08▲	4.2E-04►	1.4E-11▲	1.2E-12▲	1.2E-12▲	1.2E-12▲	1.16E-12▲	5.40E-13▲	4.84E-12▲
$f_{21}(\mathbf{x})$	1.2E-12▲	3.0E-11▲	1.9E-06▲	3.0E-11▲	3.0E-11▲	5.6E-10▲	1.98E-11▲	1.95E-11▲	2.81E-11▲
$f_{22}(\mathbf{x})$	4.1E-12▲	4.1E-12▲	4.1E-12▲	4.1E-12▲	4.1E-12▲	4.1E-12▲	5.78E-13▲	1.15E-11▲	5.66E-12▲
$f_{23}(\mathbf{x})$	4.1E-12▲	4.1E-12▲	1.2E-12▲	1.2E-12▲	1.2E-12▲	1.4E-08▲	1.03E-12▲	9.15E-13▲	3.18E-12▲

Table 14. p-values of Wilcoxon-Bonferroni test comparing YSGA vs GWO, YSGA vs PSO, YSGA vs SA-PSO, YSGA vs ABC, YSGA vs CSA, YSGA vs NAA, YSGA vs MFO, YSGA vs AFSA, and YSGA vs WOA over the **AB** values from **Table 11**, $n = 30$.

The Wilcoxon test for the multimodal functions considering 50 dimensions is visualized in **Table 15**. In this case, it is visible that all the p-values in the columns YSGA vs GWO, YSGA vs PSO, YSGA vs ABC, YSGA vs CSA, YSGA vs MFO, and YSGA vs AFSA validate that the proposed approach outperforms its competitors for all the functions. The Wilcoxon analysis could determine discrepancy between the YSGA vs NAA and YSGA vs WOA in functions f_6 and f_9 respectively, the symbol ▼ indicates that the YSGA has worse performance than its contenders. In the same way, there was no difference detected by the statistical test between YSGA and SA-PSO in functions f_{15} and f_{16} as well as YSGA and WOA in functions f_9 and f_{12}, hence the symbol ▼ is placed. For the rest of the functions in all comparisons, the Wilcoxon test supports the results recorded in **Table 12** exposing that the proposed approach outperforms its opponents.

	YSGA vs GWO	**YSGA vs PSO**	**YSGA vs SA-PSO**	**YSGA vs ABC**	**YSGA vs CSA**	**YSGA vs NAA**	**YSGA vs MFO**	**YSGA vs AFSA**	**YSGA vs WOA**
$f_6(\mathbf{x})$	1.3E-11▲	4.1E-12▲	1.1E-11▲	1.2E-11▲	1.2E-11▲	1.8E-07▼	1.48E-11▲	4.53E-12▲	1.43E-11▲
$f_7(\mathbf{x})$	7.0E-07▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	1.74E-11▲	9.25E-12▲	1.30E-11▲
$f_8(\mathbf{x})$	3.0E-11▲	2.0E-10▲	3.0E-11▲	3.0E-11▲	1.1E-07▲	3.0E-11▲	1.81E-12▲	1.54E-11▲	5.58E-12▲
$f_9(\mathbf{x})$	3.0E-11▲	4.6E-10▲	7.0E-07▲	3.0E-11▲	2.9E-07▲	3.0E-11▲	7.09E-12▲	1.54E-11▲	2.73E-11▼
$f_{10}(\mathbf{x})$	1.3E-11▲	3.2E-09▲	1.3E-08▲	3.0E-11▲	1.8E-09▲	2.4E-05▲	1.65E-08▲	2.47E-11▲	2.96E-11▲
$f_{11}(\mathbf{x})$	1.1E-06▲	1.1E-10▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	5.6E-10▲	2.48E-11▲	2.40E-11▲	1.33E-11▲
$f_{12}(\mathbf{x})$	6.0E-08▲	3.0E-11▲	3.6E-11▲	3.0E-11▲	2.4E-10▲	1.6E-11▲	4.65E-13▲	1.95E-11▲	3.36E-12▼
$f_{13}(\mathbf{x})$	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	1.2E-12▲	4.1E-11▲	1.30E-12▲	1.14E-11▲	1.44E-10▲
$f_{14}(\mathbf{x})$	3.0E-11▲	7.0E-07▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	5.10E-12▲	2.45E-11▲	2.28E-10▲
$f_{15}(\mathbf{x})$	5.4E-11▲	3.0E-11▲	3.1E-12▼	3.0E-11▲	6.0E-08▲	8.5E-09▲	1.96E-11▲	1.61E-11▲	1.80E-11▲
$f_{16}(\mathbf{x})$	3.0E-11▲	3.0E-11▲	3.0E-11▼	3.0E-11▲	4.0E-10▲	1.1E-07▲	2.21E-11▲	1.06E-11▲	7.92E-12▲
$f_{17}(\mathbf{x})$	2.0E-11▲	2.3E-05▲	3.2E-12▲	3.2E-12▲	9.4E-12▲	4.0E-10▲	2.66E-12▲	1.52E-11▲	9.75E-12▲
$f_{18}(\mathbf{x})$	2.8E-11▲	1.2E-12▲	1.1E-12▲	1.3E-05▲	1.2E-12▲	1.2E-12▲	7.85E-12▲	8.46E-12▲	8.62E-13▲
$f_{19}(\mathbf{x})$	2.3E-11▲	1.2E-12▲	1.2E-12▲	2.2E-06▲	1.2E-12▲	1.2E-12▲	6.63E-13▲	6.67E-13▲	2.69E-13▲
$f_{20}(\mathbf{x})$	3.7E-11▲	1.2E-12▲	2.0E-10▲	1.2E-12▲	1.2E-12▲	1.2E-12▲	3.59E-13▲	7.54E-13▲	1.42E-13▲
$f_{21}(\mathbf{x})$	1.2E-12▲	1.9E-06▲	1.1E-06▲	3.0E-11▲	3.0E-11▲	2.5E-09▲	2.60E-09▲	1.77E-11▲	8.96E-12▲
$f_{22}(\mathbf{x})$	1.4E-11▲	1.4E-11▲	1.4E-11▲	1.4E-11▲	1.4E-11▲	1.4E-11▲	2.73E-12▲	3.36E-12▲	4.60E-12▲
$f_{23}(\mathbf{x})$	1.3E-11▲	1.4E-11▲	1.2E-12▲	1.2E-12▲	1.2E-12▲	5.6E-08▲	8.32E-13▲	3.65E-13▲	1.03E-11▲

Table 15. p-values of Wilcoxon-Bonferroni test comparing YSGA vs GWO, YSGA vs PSO, YSGA vs SA-PSO, YSGA vs ABC, YSGA vs CSA, YSGA vs NAA, YSGA vs MFO, YSGA vs AFSA, and YSGA vs WOA over the **AB** values from **Table 12**, $n = 50$.

The results of the Wilcoxon test for the multimodal functions in 100 dimensions are presented in **Table 16**. From these results, it is evident that all the p-values in the columns YSGA vs GWO, YSGA vs ABC, YSGA vs CSA, YSGA vs MFO, and YSGA vs AFSA confirm that the proposed approach performs better than its contenders for all the test functions. The Wilcoxon analysis could determine discrepancy between the YSGA vs NAA in functions f_8 and f_{15}; and between YSGA vs WOA in functions f_9, f_{12}, f_{14}, f_{20} and f_{23} hence the symbol ▼ is located indicating that the YSGA has worse performance than its contenders according to the results of **Table 13**. On the other hand, there was no difference detected by the statistical test in YSGA vs PSO for functions f_{21} and f_{23}, thus the symbol ► is placed to indicate it. Similarly, there was no difference detected by the statistical test in YSGA vs SA-PSO in functions f_9 and f_{12}. For the rest of the functions in all comparisons, the Wilcoxon test supports the results registered in **Table 13** exposing that the proposed approach outperforms its adversaries.

	YSGA vs GWO	YSGA vs PSO	YSGA vs SA-PSO	YSGA vs ABC	YSGA vs CSA	YSGA vs NAA	YSGA vs MFO	YSGA vs AFSA	YSGA vs WOA
$f_6(\mathbf{x})$	1.0E-07▲	3.4E-11▲	2.7E-11▲	1.3E-10▲	1.6E-10▲	6.0E-08▲	1.63E-10▲	6.82E-12▲	1.03E-11▲
$f_7(\mathbf{x})$	2.3E-10▲	1.4E-12▲	2.3E-11▲	2.3E-11▲	4.6E-12▲	1.9E-11▲	1.98E-11▲	2.10E-11▲	2.02E-06▲
$f_8(\mathbf{x})$	1.6E-06▲	3.7E-10▲	7.7E-12▲	1.1E-11▲	2.5E-11▲	3.0E-09▼	1.23E-11▲	2.05E-12▲	5.79E-12▲
$f_9(\mathbf{x})$	1.3E-06▲	7.8E-06▲	2.7E-03►	1.7E-11▲	4.5E-09▲	1.5E-11▲	2.48E-11▲	7.69E-12▲	2.23E-11▼
$f_{10}(\mathbf{x})$	4.7E-10▲	1.2E-12▲	1.2E-12▲	1.3E-13▲	1.7E-12▲	6.9E-13▲	8.70E-13▲	6.77E-12▲	2.94E-13▲
$f_{11}(\mathbf{x})$	4.6E-10▲	9.6E-12▲	2.7E-11▲	1.6E-12▲	2.4E-12▲	8.1E-09▲	2.93E-11▲	2.02E-11▲	1.11E-12▲
$f_{12}(\mathbf{x})$	2.9E-06▲	2.9E-11▲	9.0E-02►	1.6E-11▲	3.0E-05▲	7.2E-12▲	1.60E-11▲	2.55E-11▲	8.13E-12▼
$f_{13}(\mathbf{x})$	2.9E-06▲	9.8E-10▲	1.7E-11▲	2.4E-11▲	2.2E-10▲	3.7E-12▲	9.82E-12▲	1.04E-11▲	3.83E-09▲
$f_{14}(\mathbf{x})$	1.5E-06▲	1.3E-11▲	4.2E-12▲	2.8E-11▲	2.9E-11▲	1.5E-11▲	3.19E-12▲	2.36E-11▲	5.69E-12▼
$f_{15}(\mathbf{x})$	2.4E-06▲	1.2E-11▲	5.0E-12▲	3.9E-12▲	1.4E-13▲	7.2E-12▼	7.40E-13▲	2.04E-11▲	3.48E-13▲
$f_{16}(\mathbf{x})$	4.3E-07▲	2.3E-11▲	7.8E-12▲	1.7E-11▲	4.3E-11▲	1.3E-11▲	9.44E-13▲	2.03E-13▲	1.10E-13▲
$f_{17}(\mathbf{x})$	1.3E-06▲	3.1E-08▲	1.0E-12▲	5.7E-13▲	9.9E-13▲	9.6E-12▲	5.13E-13▲	9.74E-12▲	9.32E-12▲
$f_{18}(\mathbf{x})$	7.5E-10▲	2.3E-13▲	2.9E-13▲	1.4E-14▲	1.1E-12▲	1.1E-12▲	6.87E-12▲	5.10E-12▲	5.10E-12▲
$f_{19}(\mathbf{x})$	6.5E-10▲	5.9E-13▲	9.5E-13▲	4.1E-13▲	1.0E-13▲	1.1E-12▲	3.45E-12▲	5.88E-12▲	4.02E-12▲
$f_{20}(\mathbf{x})$	7.9E-10▲	5.4E-13▲	2.8E-13▲	2.0E-13▲	4.8E-13▲	5.9E-13▲	1.86E-13▲	1.39E-15▲	5.16E-13▼
$f_{21}(\mathbf{x})$	2.0E-06▲	5.8E-02►	2.8E-11▲	2.4E-11▲	1.9E-11▲	1.5E-11▲	1.18E-10▲	1.40E-11▲	1.95E-11▲
$f_{22}(\mathbf{x})$	9.0E-08▲	8.6E-13▲	4.2E-13▲	3.8E-13▲	9.7E-13▲	4.1E-13▲	8.61E-12▲	6.87E-12▲	1.27E-11▲
$f_{23}(\mathbf{x})$	7.0E-10▲	2.5E-01►	2.4E-13▲	6.4E-13▲	5.2E-13▲	1.1E-12▲	6.39E-13▲	5.59E-13▲	8.23E-13▼

Table 16. p-values of Wilcoxon-Bonferroni test comparing YSGA vs GWO, YSGA vs PSO, YSGA vs SA-PSO, YSGA vs ABC, YSGA vs CSA, YSGA vs NAA, YSGA vs MFO, YSGA vs AFSA, and YSGA vs WOA over the **AB** values from **Table 13**, $n = 100$.

From **Tables 14**, **15** and **16**, it can be concluded that the statistical test exhibits evidence that ratifies the results registered in **Tables 11**, **12** and **13** exposing that the proposed approach outperforms its competitors. The statistical analysis of the YSGA algorithm against the other methods and its performance over the multimodal benchmark functions considering 30, 50 and 100 dimensions demonstrate that the YSGA method is efficient in high-dimensional optimization problems and reaches better performance than its adversaries.

5.2.3 Results of composite test functions

Functions from f_{24} to f_{27} are included in **Appendix A** and listed in **Table AIII**. These are composite functions which are used to test the effectiveness of the proposed approach in a higher level of complexity. This kind of functions are generated as a mixture of different multimodal functions. Therefore, it is evident that composite functions represent greater difficulty in solving them. The experimental results collected from 30 distinct executions are exhibited in **Tables 17**, **18** and **19**. Indicators **AB**, **MB** and **SD** illustrate the accuracy of the results considering the performance of each algorithm for every test function in 30, 50 and 100 dimensions respectively. The best results are highlighted in boldface.

		YSGA	GWO	PSO	SA-PSO	ABC	CSA	NAA	MFO	AFSA	WOA
$f_{24}(\mathbf{x})$	**AB**	**3.71E-99**	9.19E-41	2.61E+04	7.28E-05	5.28E-01	8.72E-02	7.91E-46	2.78E+04	5.57E+03	2.32E-14
	MB	**4.51E-106**	7.31E-41	2.01E+04	6.72E-05	5.17E-01	8.26E-02	4.76E-47	2.51E+04	4.28E+02	5.04E-15
	SD	**2.03E-98**	7.98E-41	1.70E+04	1.26E-05	1.49E-01	3.28E-02	2.24E-45	1.80E+04	9.79E+03	4.36E-14
$f_{25}(\mathbf{x})$	**AB**	**2.90E+01**	3.90E+01	1.73E+02	8.52E+01	2.12E+02	1.05E+02	6.32E+01	2.31E+02	3.21E+02	2.91E+01
	MB	**2.90E+01**	3.90E+01	9.31E+01	8.37E+01	2.12E+02	1.03E+02	2.90E+01	1.59E+02	3.28E+02	2.92E+01
	SD	**3.63E-06**	5.52E-05	1.57E+02	1.09E+01	1.97E+01	2.34E+01	5.28E+01	1.49E+02	3.97E+01	9.54E-05
$f_{26}(\mathbf{x})$	**AB**	4.11E+01	**3.20E+01**	1.48E+02	3.30E+01	1.89E+05	6.01E+01	4.38E+01	1.37E+07	2.53E+02	1.46E+03
	MB	3.97E+01	**3.20E+01**	1.46E+02	3.30E+01	1.42E+05	6.01E+01	3.91E+01	1.16E+02	2.55E+02	1.45E+03
	SD	9.23E+00	**1.14E-05**	3.63E+01	4.05E-01	1.59E+05	9.86E+00	1.74E+01	7.49E+07	3.08E+01	7.39E+02
$f_{27}(\mathbf{x})$	**AB**	**2.90E+01**	3.00E+01	1.02E+03	1.07E+02	1.36E+02	1.20E+02	3.28E+01	1.06E+03	1.12E+03	2.91E+01
	MB	**2.90E+01**	3.00E+01	9.68E+02	1.05E+02	1.36E+02	1.19E+02	2.90E+01	1.13E+03	9.81E+02	2.93E+01
	SD	**0.00E+00**	9.38E-15	5.24E+02	4.18E+01	3.57E+01	2.83E+01	2.06E+01	5.27E+02	4.97E+02	1.33E-04

Table 17. Optimal results of the composite test functions considering 30 dimensions.

The results contained in **Table 17** exhibit evidence that the YSGA algorithm achieve superior performance than its competitors in functions f_{24}, f_{25} and f_{27}. On the other hand, the GWO method reaches better indicators **AB**, **MB** and **SD** for function f_{26}.

To examine the adjustability of the YSGA algorithm, the experiments have been repeated but now considering 50 dimensions instead of 30. The results of this test are shown in **Table 18**.

		YSGA	GWO	PSO	SA-PSO	ABC	CSA	NAA	MFO	AFSA	WOA
$f_{24}(\mathbf{x})$	**AB**	**1.67E-96**	1.74E-49	7.56E+04	1.56E-02	1.81E+03	2.03E-01	5.37E-58	6.19E+04	1.73E+05	7.18E-14
	MB	**1.29E-101**	8.93E-50	8.03E+04	7.97E-03	1.84E+03	1.88E-01	7.30E-60	5.52E+04	1.40E+05	8.22E-15
	SD	**5.86E-96**	1.94E-49	3.75E+04	1.37E-02	3.50E+02	6.21E-02	1.66E-57	3.45E+04	1.12E+05	1.47E-13
$f_{25}(\mathbf{x})$	**AB**	**4.90E+01**	5.90E+01	6.80E+02	1.57E+02	6.83E+02	1.80E+02	1.81E+02	5.95E+02	4.39E+02	4.91E+01
	MB	**4.90E+01**	5.90E+01	6.84E+02	1.71E+02	6.86E+02	1.76E+02	1.68E+02	5.42E+02	4.32E+02	4.92E+01
	SD	**1.38E-05**	1.77E-04	3.07E+02	5.14E+01	3.08E+01	3.31E+01	1.30E+02	2.93E+02	2.90E+01	1.61E-04
$f_{26}(\mathbf{x})$	**AB**	1.11E+02	**5.40E+01**	7.91E+02	6.54E+01	9.76E+07	1.46E+02	6.72E+02	2.73E+07	9.84E+02	7.16E+03
	MB	1.04E+02	**5.40E+01**	7.85E+02	6.53E+01	9.52E+07	1.46E+02	6.33E+02	1.81E+03	9.88E+02	6.52E+03
	SD	3.87E+01	**2.38E-05**	1.77E+02	7.00E-01	2.52E+07	2.02E+01	1.95E+02	1.04E+08	1.16E+02	3.42E+03
$f_{27}(\mathbf{x})$	**AB**	**4.90E+01**	5.10E+01	2.57E+03	2.04E+02	6.30E+02	2.03E+03	6.68E+01	2.20E+03	5.02E+01	4.96E+01
	MB	**4.90E+01**	5.00E+01	2.86E+03	2.08E+02	6.34E+02	9.40E+02	5.90E+01	2.09E+03	5.02E+01	4.95E+01
	SD	**4.76E-15**	2.21E-14	1.16E+03	4.73E+01	3.28E+01	1.73E+03	5.74E+01	1.10E+03	1.07E-01	8.66E-04

Table 18. Optimal results of the composite test functions considering 50 dimensions.

From **Table 18**, it can be detected that the YSGA algorithm achieves better performance than its competitors in functions f_{24}, f_{25} and f_{27}. By contrast, the GWO outperforms the YSGA in function f_{26}.

Additionally, the experiments have been repeated but now considering 100 dimensions as an alternative to provide increased complexity to the test. The results are reported in **Table 19**.

		YSGA	GWO	PSO	SA-PSO	ABC	CSA	NAA	MFO	AFSA	WOA
$f_{24}(\mathbf{x})$	**AB**	**5.32E-116**	2.55E-20	3.22E+05	2.93E+00	2.57E+05	2.26E-01	1.09E-93	1.10E+05	4.37E+25	1.52E-28
	MB	**1.38E-122**	2.17E-20	3.35E+05	2.93E+00	1.46E+05	2.28E-01	3.89E-97	1.10E+05	3.66E+20	2.64E-30
	SD	**2.91E-115**	1.41E-20	6.78E+04	6.78E+04	5.93E+05	3.39E-02	4.58E-93	3.97E+04	2.37E+26	3.85E-28
$f_{25}(\mathbf{x})$	**AB**	**9.90E+01**	1.01E+02	2.17E+03	2.84E+02	3.37E+03	3.38E+02	1.38E+02	1.41E+03	1.02E+03	9.98E+01
	MB	**9.90E+01**	1.01E+02	2.13E+03	2.84E+02	3.36E+03	3.36E+02	4.90E+02	1.35E+03	1.03E+03	9.97E+01
	SD	**1.02E-04**	4.69E-04	5.21E+02	5.21E+02	1.86E+02	4.92E+01	1.09E+02	3.63E+02	4.53E+01	4.79E-04
$f_{26}(\mathbf{x})$	**AB**	4.51E+02	**1.09E+02**	2.87E+08	3.46E+02	1.92E+09	3.72E+02	5.42E+02	3.07E+08	4.38E+03	3.68E+04
	MB	4.32E+02	**1.09E+02**	4.10E+08	3.46E+02	1.95E+09	3.68E+02	4.85E+02	3.53E+08	4.31E+03	3.12E+04
	SD	1.90E+02	**6.09E-05**	2.88E+08	2.88E+08	3.60E+08	4.74E+01	2.74E+02	3.11E+08	4.81E+02	1.52E+04
$f_{27}(\mathbf{x})$	**AB**	**9.90E+01**	1.04E+02	9.00E+06	3.38E+02	6.24E+03	3.94E+02	8.69E+02	4.51E+03	1.02E+02	9.98E+01
	MB	**9.90E+01**	1.04E+02	1.08E+04	3.38E+02	4.63E+03	3.88E+02	4.90E+02	4.23E+03	1.02E+02	9.98E+01
	SD	**0.00E+00**	3.24E-14	4.93E+07	4.93E+07	8.03E+03	5.70E+01	7.03E+01	1.75E+03	1.47E-01	1.28E-03

Table 19. Optimal results of the composite test functions considering 100 dimensions.

From **Table 19**, it can be visualized that the YSGA algorithm achieves better performance than its competitors in functions f_{24}, f_{25} and f_{27}. On the other hand, the GWO outperforms the YSGA in functions f_8, f_{14} and f_{15}.

The p-values resulting from the Wilcoxon test, considering the Bonferroni correction, are stated in **Tables 20**, **21** and **22**. After applying the Wilcoxon test over the **AB** data from **Table 17**, the p-values generated were listed in **Table 20**. In the same way, the p-values generated by the Wilcoxon test over the **AB** data from **Tables 18** and **19** were registered in **Table 21** and **22** respectively.

The Wilcoxon test for the multimodal functions considering 30 dimensions is listed in **Table 20**. In this case, it is visible that all the p-values in the columns YSGA vs PSO, YSGA vs SA-PSO, YSGA vs ABC and YSGA vs CSA, YSGA vs NAA, YSGA vs MFO, YSGA vs AFSA and YSGA vs WOA validate that the proposed approach outperforms its competitors for all the functions. Besides, it can be observed that function f_{26} in column YSGA vs GWO has the symbol ▼ as an indicator that for this function the GWO reaches better performances than the YSGA. For the rest of the functions in the same column, the Wilcoxon test supports the results recorded in **Table 17** exposing that the YSGA algorithm outperforms its opponents.

	YSGA vs GWO	YSGA vs PSO	YSGA vs SA-PSO	YSGA vs ABC	YSGA vs CSA	YSGA vs NAA	YSGA vs MFO	YSGA vs AFSA	YSGA vs WOA
$f_{24}(\mathbf{x})$	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	2.82E-11▲	8.34E-12▲	7.58E-12▲

$f_{25}(\mathbf{x})$	2.9E-11▲	4.6E-06▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.8E-06▲	2.05E-11▲	2.05E-11▲	1.86E-11▲
$f_{26}(\mathbf{x})$	1.1E-11▼	3.0E-11▲	3.0E-11▲	3.0E-11▲	2.3E-04▲	1.3E-10▲	4.16E-11▲	1.98E-11▲	1.58E-11▲
$f_{27}(\mathbf{x})$	7.2E-13▲	5.4E-13▲	2.4E-12▲	2.4E-12▲	2.4E-12▲	6.0E-07▲	9.01E-13▲	1.97E-13▲	5.77E-12▲

Table 20. p-values of Wilcoxon-Bonferroni test comparing YSGA vs GWO, YSGA vs PSO, YSGA vs SA-PSO, YSGA vs ABC, YSGA vs CSA, YSGA vs NAA, YSGA vs MFO, YSGA vs AFSA, and YSGA vs WOA over the **AB** values from **Table 17**, $n = 30$.

The Wilcoxon test for the multimodal functions considering 50 dimensions is written in **Table 21**. In this case, it is visible that all the p-values in the columns YSGA vs PSO, YSGA vs SA-PSO, YSGA vs ABC and YSGA vs CSA, YSGA vs NAA, YSGA vs MFO, YSGA vs AFSA and YSGA vs WOA validate that the proposed approach outperforms its competitors for all the functions. The Wilcoxon analysis could determine discrepancy between the YSGA vs GWO in function f_{26} therefore the symbol ▼ specifies that the YSGA has worse performance than the GWO. For the rest of the functions in the same column, the Wilcoxon test supports the results registered in **Table 18** exposing that the proposed approach outperforms its opponent.

	YSGA vs GWO	**YSGA vs PSO**	**YSGA vs SA-PSO**	**YSGA vs ABC**	**YSGA vs CSA**	**YSGA vs NAA**	**YSGA vs MFO**	**YSGA vs AFSA**	**YSGA vs WOA**
$f_{24}(\mathbf{x})$	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	5.54E-12▲	1.42E-11▲	1.53E-11▲
$f_{25}(\mathbf{x})$	2.9E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	3.0E-11▲	9.2E-05▲	1.11E-11▲	6.96E-12▲	2.58E-12▲
$f_{26}(\mathbf{x})$	1.8E-11▼	3.0E-11▲	3.0E-11▲	3.0E-11▲	4.0E-05▲	3.0E-11▲	1.89E-11▲	2.55E-11▲	7.93E-12▲
$f_{27}(\mathbf{x})$	7.4E-13▲	8.2E-13▲	4.1E-12▲	4.1E-12▲	4.1E-12▲	1.0E-11▲	3.20E-12▲	2.36E-13▲	1.94E-11▲

Table 21. p-values of Wilcoxon-Bonferroni test comparing YSGA vs GWO, YSGA vs PSO, YSGA vs SA-PSO, YSGA vs ABC, YSGA vs CSA, YSGA vs NAA, YSGA vs MFO, YSGA vs AFSA, and YSGA vs WOA over the **AB** values from **Table 18**, $n = 50$.

The results of the Wilcoxon test for the multimodal functions in 100 dimensions are presented in **Table 22**. From these results, it is evident that all the p-values in the columns YSGA vs PSO, YSGA vs SA-PSO, YSGA vs ABC and YSGA vs CSA and YSGA vs NAA confirm that the proposed approach performs better than its contenders for all the test functions. The Wilcoxon analysis could determine discrepancy between the YSGA vs GWO in function f_{26}, thus the symbol ▼ is placed on the right side of the p-value indicating that the YSGA has worse performance than the GWO according to the results of **Table 19**.

	YSGA vs GWO	**YSGA vs PSO**	**YSGA vs SA-PSO**	**YSGA vs ABC**	**YSGA vs CSA**	**YSGA vs NAA**	**YSGA vs MFO**	**YSGA vs AFSA**	**YSGA vs WOA**
$f_{24}(\mathbf{x})$	2.8E-06▲	3.0E-12▲	2.8E-12▲	1.8E-12▲	1.0E-11▲	4.1E-12▲	1.38E-11▲	2.33E-11▲	1.92E-11▲
$f_{25}(\mathbf{x})$	3.0E-07▲	2.1E-11▲	1.9E-11▲	1.8E-11▲	5.5E-12▲	7.4E-02▲	2.64E-11▲	9.74E-12▲	2.85E-11▲
$f_{26}(\mathbf{x})$	2.3E-06▼	2.0E-11▲	1.4E-11▲	7.9E-12▲	7.6E-04▲	1.7E-05▲	1.56E-11▲	2.37E-11▲	6.31E-12▲
$f_{27}(\mathbf{x})$	2.2E-06▲	1.8E-13▲	4.3E-13▲	7.9E-13▲	1.8E-13▲	3.6E-05▲	1.14E-12▲	5.71E-13▲	8.59E-13▲

Table 22. p-values of Wilcoxon-Bonferroni test comparing YSGA vs GWO, YSGA vs PSO, YSGA vs SA-PSO, YSGA vs ABC, YSGA vs CSA, YSGA vs NAA, YSGA vs MFO, YSGA vs AFSA, and YSGA vs WOA over the **AB** values from **Table 19**, $n = 100$.

From **Tables 20**, **21** and **22**, it can be concluded that the Wilcoxon test validates the results contained in **Tables 17**, **18** and **19** exposing that the proposed approach outperforms its opponents. The statistical study of the YSGA algorithm against the other algorithms and its performance over the composite benchmark functions considering 30, 50 and 100 dimensions prove that the YSGA method is effective in high-dimensional optimization problems and obtains better performance than its challengers.

5.3 Test results considering different populations

Additional experiments have been conducted to analyze the performance of the algorithms with different population size. In the experiments, populations of $m = 10$, $m = 20$, $m = 50$, and $m = 100$ have been considered along with three benchmark functions: one unimodal f_1, one multimodal f_{17} and one composite f_{26}. The experimental results collected from 30 distinct executions and 1000 iterations, are exhibited in **Tables 23**, **24**, **25** and **26** for $m = 10$, $m = 20$, $m = 50$, and $m = 100$, respectively. Indexes **AB**, **MB** and **SD** illustrate the accuracy of the results considering the performance of each algorithm for every test function in 50 dimensions. The best results are highlighted in boldface.

		YSGA	GWO	PSO	SA-PSO	ABC	CSA	NAA	MFO	AFSA	WOA
$f_1(\mathbf{x})$	**AB**	**3.37E-82**	5.65E-47	1.59E+02	3.09E-03	1.06E+02	7.09E-02	1.02E-29	9.57E-01	2.50E-20	1.25E+02
	MB	**6.43E-104**	1.05E-47	1.11E+02	9.23E-04	1.02E+02	6.70E-02	7.23E-34	9.69E-01	1.49E-24	1.07E+02
	SD	**1.84E-81**	1.74E-46	1.16E+02	4.79E-03	2.63E+01	3.01E-02	4.08E-29	1.06E-01	1.35E-19	1.05E+02
$f_{17}(\mathbf{x})$	**AB**	**7.65E+01**	1.33E+02	4.03E+02	1.20E+02	4.88E+02	8.00E+01	7.49E+01	3.92E+02	9.87E+01	3.57E+02
	MB	**8.46E+01**	9.52E+01	4.00E+02	1.18E+02	4.94E+02	8.71E+01	9.14E+01	3.95E+02	1.02E+02	3.46E+02
	SD	**4.66E+01**	5.11E+01	6.32E+01	6.82E+01	6.44E+01	5.06E+01	7.40E+01	8.06E+01	1.66E+02	6.62E+01
$f_{27}(\mathbf{x})$	**AB**	**5.70E+01**	8.90E+01	6.05E+01	1.78E+02	8.04E+01	6.80E+01	6.90E+01	7.20E+01	7.70E+01	9.11E+01
	MB	**4.85E+01**	8.90E+01	5.24E+01	1.70E+02	6.04E+01	6.33E+01	6.90E+01	7.70E+01	6.60E+01	9.91E+01
	SD	2.46E+01	7.72E+00	1.06E+01	3.40E+01	1.33E+01	**2.13E+00**	9.26E+00	4.45E+00	5.24E+00	7.68E+00

Table 23. Optimal results of functions f_1, f_{17}, and f_{27} for $m = 10$ considering 50 dimensions.

		YSGA	GWO	PSO	SA-PSO	ABC	CSA	NAA	MFO	AFSA	WOA
$f_1(\mathbf{x})$	**AB**	**4.37E-99**	6.10E-64	1.23E+02	2.33E-03	8.28E-03	8.28E-03	2.80E-27	8.37E-01	1.76E-31	6.36E+01
	MB	**4.71E-118**	8.74E-65	1.00E+02	6.41E-04	7.56E-03	7.56E-03	1.97E-31	8.47E-01	1.65E-35	1.00E+02
	SD	**2.39E-98**	1.18E-63	9.71E+01	5.16E-03	3.19E-03	3.19E-03	1.52E-26	1.05E-01	9.00E-31	7.16E+01
$f_{17}(\mathbf{x})$	**AB**	**6.76E+01**	6.92E+01	3.64E+02	7.86E+01	7.56E+01	8.76E+01	1.06E+02	3.71E+02	6.83E+01	3.36E+02
	MB	**6.42E+01**	6.65E+01	3.50E+02	7.13E+01	8.21E+01	8.41E+01	8.21E+01	3.78E+02	7.12E+01	3.28E+02
	SD	4.12E+00	1.54E+01	5.45E+01	8.61E+00	1.25E+01	**2.52E+00**	7.16E+01	2.70E+01	1.50E+01	4.80E+01
$f_{27}(\mathbf{x})$	**AB**	**5.47E+01**	5.60E+01	5.91E+01	1.64E+02	5.90E+01	6.10E+01	6.00E+01	7.30E+01	6.00E+01	7.60E+01
	MB	**5.11E+01**	5.90E+01	6.04E+01	1.58E+02	6.70E+01	5.90E+01	5.90E+01	6.90E+01	6.10E+01	7.00E+01
	SD	2.23E+01	1.23E+00	9.72E+00	2.97E+01	7.71E+00	8.77E+00	**1.89E-02**	1.22E+01	1.78E+00	9.26E+00

Table 24. Optimal results of functions f_1, f_{17}, and f_{27} for $m = 20$ considering 50 dimensions.

		YSGA	GWO	PSO	SA-PSO	ABC	CSA	NAA	MFO	AFSA	WOA
$f_1(\mathbf{x})$	**AB**	**4.31E-172**	3.96E-87	2.80E+01	3.88E-04	4.20E+00	1.20E-04	1.29E-57	1.96E+01	4.01E-01	2.66E-22
	MB	**3.03E-185**	2.29E-88	2.62E+01	3.97E-04	4.23E+00	1.16E-04	2.75E-62	2.62E+01	4.03E-01	8.43E-28
	SD	**0.00E+00**	1.82E-86	1.94E+01	1.25E-04	9.17E-01	4.78E-05	7.04E-57	2.07E+01	6.20E-02	1.45E-21
$f_{17}(\mathbf{x})$	**AB**	**0.00E+00**	1.79E+00	3.33E+02	4.49E+01	4.73E+02	3.36E+01	4.58E+01	3.06E+02	3.15E+02	1.16E-08
	MB	**0.00E+00**	0.00E+00	3.27E+02	1.83E+01	4.71E+02	3.14E+01	4.34E+01	3.09E+02	3.18E+02	2.84E-14
	SD	**0.00E+00**	6.02E+00	5.69E+01	4.88E+01	1.97E+01	1.15E+01	3.10E+01	4.18E+01	2.40E+01	6.36E-08
$f_{27}(\mathbf{x})$	**AB**	**2.90E+01**	3.00E+01	1.02E+03	1.07E+02	1.36E+02	1.20E+02	3.28E+01	1.06E+03	1.12E+03	2.91E+01
	MB	**2.90E+01**	3.00E+01	9.68E+02	1.05E+02	1.36E+02	1.19E+02	2.90E+01	1.13E+03	9.81E+02	2.93E+01
	SD	**0.00E+00**	9.38E-15	5.24E+02	4.18E+01	3.57E+01	2.83E+01	2.06E+01	5.27E+02	4.97E+02	1.33E-04

Table 25. Optimal results of functions f_1, f_{17}, and f_{27} for $m = 50$ considering 50 dimensions.

		YSGA	GWO	PSO	SA-PSO	ABC	CSA	NAA	MFO	AFSA	WOA
$f_1(\mathbf{x})$	**AB**	**1.88E-204**	2.28E-64	8.00E+01	3.13E-03	3.78E-05	3.78E-05	2.75E-38	7.79E-01	4.66E-41	5.33E+01
	MB	**2.36E-187**	6.12E-65	1.00E+02	1.08E-03	3.24E-05	3.24E-05	8.93E-39	7.84E-01	1.17E-43	2.61E-05
	SD	**8.70E-146**	4.55E-10	8.05E+01	5.66E-03	1.76E-05	1.76E-05	5.73E-38	5.09E-02	1.55E-30	7.76E+01
$f_{17}(\mathbf{x})$	**AB**	**0.00E+00**	1.14E+02	2.91E+02	2.79E+01	2.77E+01	2.77E+01	1.82E+02	3.22E+02	1.23E+02	2.53E+02
	MB	**0.00E+00**	8.86E+01	2.89E+02	3.13E+01	2.24E+01	2.24E+01	1.71E+02	3.23E+02	1.08E+02	2.48E+02
	SD	**0.00E+00**	2.87E+01	4.96E+01	1.18E+01	1.44E+01	1.44E+01	9.95E+01	2.42E+01	2.31E+01	4.85E+01
$f_{27}(\mathbf{x})$	**AB**	**4.90E+01**	4.90E+01	5.01E+01	1.67E+02	4.90E+01	4.90E+01	4.90E+01	4.90E+01	4.90E+01	4.96E+01
	MB	**4.90E+01**	4.90E+01	5.01E+01	1.63E+02	4.90E+01	4.90E+01	4.90E+01	4.90E+01	4.90E+01	4.90E+01
	SD	**1.38E-05**	1.29E-04	9.12E-01	2.92E+01	4.48E-06	4.48E-06	4.10E-05	6.14E-03	1.03E-04	7.81E-01

Table 26. Optimal results of functions f_1, f_{17}, and f_{27} for $m = 100$ considering 50 dimensions.

As it can be seen from **Tables 23, 24, 25, and 26**, the experimental results demonstrate that the YSGA has a better performance than the other algorithms even when different population sizes are considered.

5.4 Convergence analysis

To visualize the evolution over the iterations of the proposed approach, the performance results of each algorithm in a single execution have been plotted to illustrate the speed of convergence. This test examines the evolution of the optimization process for each method on a particular benchmark function.

For the convergence analysis, six distinct benchmark functions were contemplated: f_4, f_5, f_{13}, f_8, f_{24} and f_{25}. The selection of the functions for the examination was made considering two unimodal functions (f_4 and f_5), two multimodal functions (f_{13} and f_8) and two composite functions (f_{24} and f_{25}). The test evaluates the performance of every algorithm when solving each function in 30 dimensions. The graphs with the results of the convergence test are illustrated in **Figure 8**, **9** and **10**.

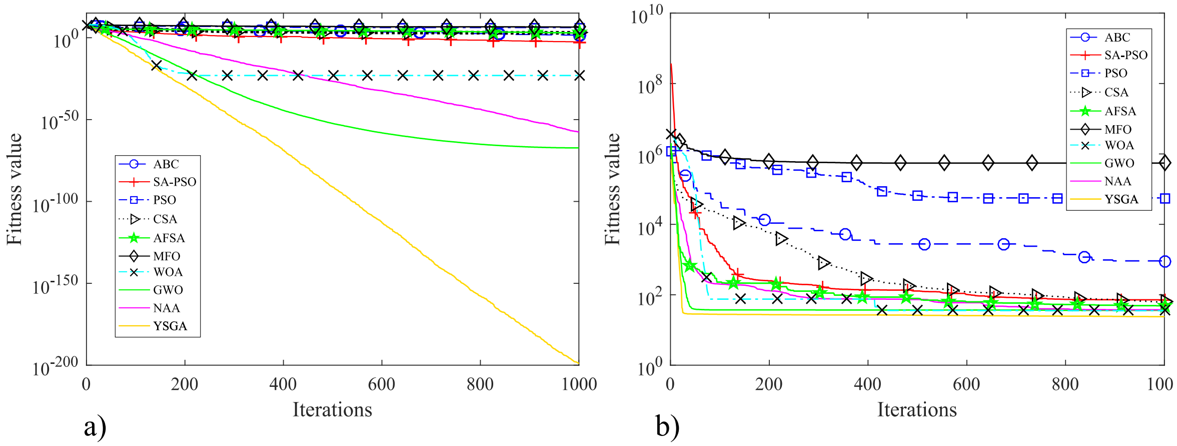

Figure 8. Convergence test for functions (a) f_4 and (b) f_5 in 30 dimensions.

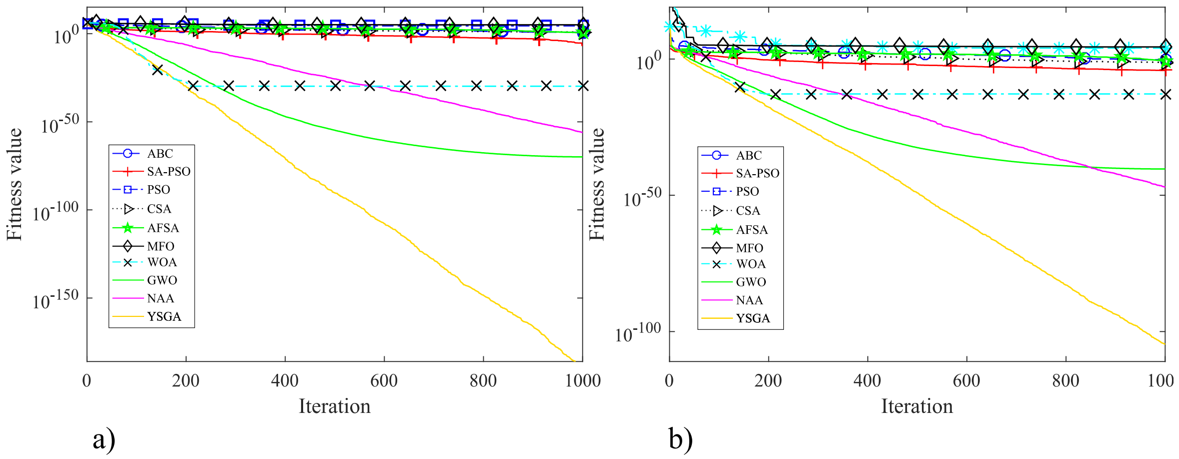

Figure 9. Convergence test for functions (c) f_{13} and (d) f_8 in 30 dimensions.

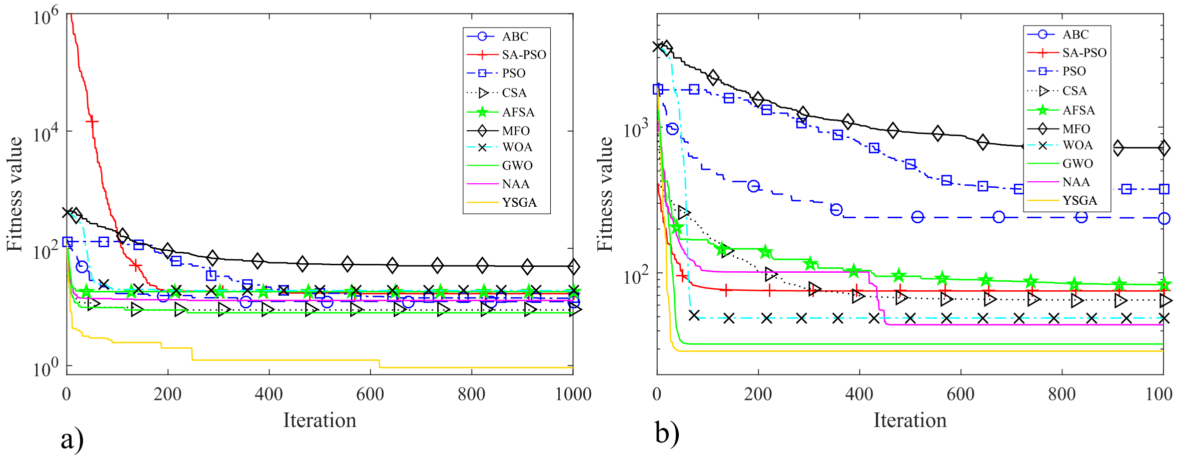

Figure 10. Convergence test for functions (e) f_{24} and (f) f_{25} in 30 dimensions.

In **Figure 8,** the convergence curves of the unimodal functions f_4 and f_5 are visualized in (a) and (b) respectively. Multimodal functions f_{13} and f_8 can be observed in (c) and (d) respectively from **Figure 9** while in **Figure 10** the convergence of the composite functions f_{24} and f_{25} are visualized in (e) and (f) respectively. It is clear that the proposed approach converges faster than the ABC, SA-PSO, PSO, CSA, GWO and NAA methods.

5.5 Engineering optimization problems

As a complimentary evaluation, the performance of the YSGA algorithm was tested in the following engineering optimization problems: the design of a gear train [58], the minimization of a spring´s tension-compression [59], the design of a pressure vessel [58], the parameter estimation for a FM synthesizer [60] and the design of a three-bar truss [61]. The cost functions of the engineering optimization problems are f_{B1}, f_{B2}, f_{B3}, f_{B4} and f_{B5} respectively. See **Appendix B** for a detailed explanation of the engineering problems.

To ensure the uniformity of the experiments, the algorithms implicated were the same as in sections 4.1, 4.2 and 4.3. Thus, the experimental results were compared with the performance of the GWO, PSO, SA-PSO, ABC, CSA, NAA, MFO, AFSA, and WOA algorithms. The parameters considered for the tests that are described in this section were taken from **Table 4**. The total population m has been set to 50 individuals and the maximum number of iterations t_{max} has been set to 1000 for all the algorithms involved. Each experiment is repeated considering 30 independent executions.

The performance of each algorithm for every engineering optimization problem can be analyzed in **Table 27**. The registered results are classified as the ***Worst*** and ***Best*** fitness value over the 30 independent runs. Likewise, the average ***Avg*** and the standard deviation ***Std*** are listed in the same table. The ***Avg*** and the ***Std*** are calculated considering the results collected from the 30 distinct executions. The best outcomes obtained by each engineering optimization problem are highlighted in boldface.

From **Table 27**, it can be noticed that the YSGA algorithm achieves better performance than its competitors for all the engineering optimization problems.

		YSGA	**GWO**	**PSO**	**SA-PSO**	**ABC**	**CSA**	**NAA**	**MFO**	**AFSA**	**WOA**
f_{B1}	***Worst***	**1.23E-11**	4.32E-10	6.93E-08	8.79E-09	1.90E-09	2.12E-09	1.32E-11	3.66E-09	2.01E-05	3.03E-08
	Best	**1.92E-12**	3.32E-12	2.31E-11	3.24E-12	2.81E-12	6.43E-11	4.14E-12	3.75E-10	5.98E-09	7.86E-10
	Avg	**1.97E-12**	4.46E-12	2.53E-11	7.22E-12	4.87E-12	3.51E-11	4.45E-12	2.86E-10	5.98E-07	3.75E-10
	Std	**4.20E-14**	7.82E-13	3.34E-12	1.68E-13	5.69E-13	1.20E-13	2.06E-12	9.00E-11	2.16E-08	6.22E-06
f_{B2}	***Worst***	**1.20E-02**	8.81E-02	1.32E-02	2.68E-02	2.06E-02	8.42E-01	1.53E-02	8.43E+00	1.12E-02	2.00E+00
	Best	**7.28E-03**	7.39E-03	7.30E-03	1.16E-02	7.39E-03	1.04E-02	7.68E-03	5.20E+00	7.45E-03	1.46E-01
	Avg	**7.77E-03**	7.90E-03	8.36E-03	1.17E-02	7.89E-03	8.20E-03	1.20E-02	6.90E+00	8.87E-03	1.96E-01
	Std	**2.70E-04**	3.58E-04	1.54E-03	1.32E-03	3.93E-04	2.95E-04	2.19E-03	3.19E+00	1.09E-03	1.26E-01
f_{B3}	***Worst***	**6.25E+03**	6.42E+03	7.30E+03	6.88E+03	6.46E+03	6.79E+03	6.71E+03	6.10E+05	2.42E+05	5.21E+05
	Best	**5.88E+03**	5.97E+03	5.89E+03	6.10E+03	5.98E+03	5.89E+03	5.89E+03	2.01E+04	1.79E+04	2.89E+04
	Avg	**6.10E+03**	6.11E+03	6.36E+03	6.11E+03	6.16E+03	6.17E+03	6.12E+03	1.89E+05	1.04E+05	1.68E+05
	Std	**5.30E+01**	1.04E+02	6.70E+02	1.07E+02	1.16E+02	1.89E+02	7.75E+01	1.56E+05	6.90E+04	1.35E+05
f_{B4}	***Worst***	**2.08E+01**	2.51E+01	2.61E+01	2.68E+01	2.13E+01	2.53E+01	2.36E+01	2.76E+01	2.55E+01	2.57E+01
	Best	**1.63E+01**	1.79E+01	1.75E+01	2.50E+01	1.70E+01	1.64E+01	1.68E+01	1.83E+01	1.80E+01	1.91E+01
	Avg	**1.72E+01**	1.89E+01	1.86E+01	2.60E+01	1.97E+01	2.06E+01	1.86E+01	2.37E+01	1.83E+01	2.03E+01
	Std	**1.17E+00**	4.80E+00	5.66E+00	1.63E+00	1.30E+00	4.35E+00	4.96E+00	2.44E+00	6.21E+00	4.52E+00

f_{B5}	***Worst***	**2.65E+02**	2.83E+02	2.83E+02	2.82E+02	2.83E+02	2.89E+02	2.82E+02	2.67E+02	2.80E+02	2.69E+02
	Best	**2.64E+02**	2.80E+02	2.80E+02	2.70E+02	2.80E+02	2.79E+02	2.79E+02	2.64E+02	2.74E+02	2.65E+02
	Avg	**2.64E+02**	2.80E+02	2.81E+02	2.80E+02	2.80E+02	2.80E+02	2.80E+02	2.65E+02	2.79E+02	2.66E+02
	Std	**3.95E-11**	9.51E-01	1.57E+00	4.57E-09	1.79E-03	1.49E-10	9.40E-08	8.59E-01	3.85E+00	4.45E+00

Table 27. Optimal results of the engineering optimization problems described in **Appendix B**.

Finally, the best set of design variables $\mathbf{x}$ are listed in **Table 28**. These results define the best parameter configuration for the best solutions founded by the YSGA approach over all 30 independent executions for every engineering optimization problems.

FUNCTION	x_1	x_2	x_3	x_4	x_5	x_6	$f_{Bi}(\mathbf{x})$ for $i = 1,\ldots, 5$
f_{B1}	1.20E+01	2.10E+01	5.90E+01	3.00E+01			1.94E-12
f_{B2}	5.00E-02	4.79E-01	4.07E+00				7.28E-03
f_{B3}	7.78E-01	3.84E-01	4.04E+01	1.99E+02			5.88E+03
f_{B4}	-5.35E-01	4.76E+00	3.36E+00	4.98E+00	3.28E-01	4.47E+00	1.63E+01
f_{B5}	7.88E-01	4.09E-01	7.88E-01				2.64E+02

Table 28. The best parameter configuration for the design variables obtained by the YSGA for each engineering optimization problem described in **Appendix B**.

6 Conclusions

In this paper, a hunting model of Yellow Saddle Goatfish has been developed. At some abstraction level, the approach can be characterized as a search strategy for optimization proposes. In the approach, individuals emulate a set of fish which experiment interaction based on the peculiar hunting behavior of the Yellow Saddle Goatfish. Initially, the proposed model designates a number of groups or sub-populations according to the spatial distribution of the individuals. The algorithm contemplates two distinct categories of search agents (fish): chasers and blockers. In each sub-population one fish assumes the role of chaser while the rest are considered blockers. Depending on the category, each element is undergone by a set of different evolutionary operations which emulate the different collaborative behaviors that are present in the natural hunting process. The fitness value represents the relative success that an element experiment during the hunting process. With the use of this biological model, the new search strategy improves the optimization results in terms of accuracy and convergence in comparison to other popular evolutionary techniques.

The performance of the proposed approach has been tested on a set of 27 well-known benchmark functions. The results were compared to other popular algorithms currently in use. They include the Gray Wolf Optimization (GWO), the Particle Swarm Optimization (PSO), the Synchronous-Asynchronous Particle Swarm Optimization (SA-PSO), the Artificial Bee Colony (ABC), the Crow Search Algorithm (CSA) and the Natural Aggregation Algorithm (NAA). Statistical validation of the results has been conducted by applying a non-parametric framework to ensure the consistency of the algorithm. The results demonstrate that they are not produced by a random effect. Additionally, the algorithm has been also tested over several engineering optimization problems and compared to the performance against its competitors. The experimental results demonstrate this technique is fast, accurate and robust.

References

[1] Predrag Slijepcevic, Evolutionary epistemology: Reviewing and reviving with new data the research programme for distributed biological intelligence, Biosystems, 163, (2018), 23-35.

[2] Belkacem Khaldi, Fouzi Harrou, Foudil Cherif, Ying Sun, Self-organization in aggregating robot swarms: A DW-KNN topological approach, Biosystems, 165, (2018), 106-121.

[3] Francesco Bianchini, Artificial intelligence and synthetic biology: A tri-temporal contribution, Biosystems, 148, (2016), 32-39.

[4] Arpan Kumar Kar, Bio inspired computing – A review of algorithms and scope of applications, Expert Systems with Applications, 59, (2016), 20-32.

[5] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1995, vol. 4, pp. 1942–1948.

[6] N. A. Ab Aziz, M. Mubin, M. S. Mohamad, and K. Ab Aziz, "A synchronous-asynchronous particle swarm optimisation algorithm.," *ScientificWorldJournal.*, vol. 2014, Arti, p. 17, 2014.

[7] D. Karaboga, "An idea based on Honey Bee Swarm for Numerical Optimization," *Tech. Rep. TR06, Erciyes Univ.*, no. TR06, p. 10, 2005.

[8] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm," *Comput. Struct.*, vol. 169, pp. 1–12, Jun. 2016.

[9] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014.

[10] F. Luo, J. Zhao, and Z. Y. Dong, "A new metaheuristic algorithm for real-parameter optimization: Natural aggregation algorithm," *2016 IEEE Congr. Evol. Comput. CEC 2016*, no. July, pp. 94–103, 2016.

[11] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.

[12] F. Fausto, E. Cuevas, A. Valdivia, and A. González, "A global optimization algorithm inspired in the behavior of selfish herds," *Biosystems*, vol. 160, pp. 39–55, Oct. 2017.

[13] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic

paradigm," *Knowledge-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.

[14] Gang Xu, Guosong Yu, On convergence analysis of particle swarm optimization algorithm, Journal of Computational and Applied Mathematics, 333, (2018), 65-73.

[15] Yukio-Pegio Gunji, Yoshiyuki Kusunoki, Nobuhide Kitabayashi, Toshinao Mochizuki, Masaki Ishikawa, Takanori Watanabe, Dual interaction producing both territorial and schooling behavior in fish, Biosystems, 50(1), (1999), 27-47.

[16] Debasis Manna, Alakes Maitib, G.P. Samanta, Analysis of a predator-prey model for exploited fish populations with schooling behavior, Applied Mathematics and Computation, 317, (2018), 35-48.

[17] Jacques Gautrais, Christian Jost and Guy Theraulaz, Key behavioural factors in a self-organised fish school model, Annales Zoologici Fennici, 45(5), (2008), 415-428.

[18] Daniel Calovi, Ugo Lopez, Sandrine Ngo, Clément Sire, Hugues Chaté and Guy Theraulaz, Swarming, schooling, milling: phase diagram of a data-driven fish school model, New Journal of Physics, 16, (2014), 1-15.

[19] J. E. Randall, *The goatfishes Parupeneus cyclostomus , P . macronemus and freeloaders*, vol. 20, no. 2. Aquaprint, 2014.

[20] C. Strübin, M. Steinegger, and R. Bshary, "On Group Living and Collaborative Hunting in the Yellow Saddle Goatfish (Parupeneus cyclostomus)1," *Ethology*, vol. 117, no. 11, pp. 961–969, Nov. 2011.

[21] M. Steinegger, D. G. Roche, and R. Bshary, "Simple decision rules underlie collaborative hunting in yellow saddle goatfish," 2018.

[22] M. I. McCormick, "Fish feeding on mobile benthic invertebrates: influence of spatial variability in habitat associations," *Mar. Biol.*, vol. 121, no. 4, pp. 627–637, Feb. 1995.

[23] J. E. Randall, *Red Sea reef fishes*. IMMEL, 1983.

[24] J. E. Randall and University of Hawaii at Manoa. Sea Grant College Program., *Reef and shore fishes of the Hawaiian Islands*. Sea Grant College Program, University of Hawai, 2007.

[25] C. Boesch and H. Boesch, "Hunting Behavior of Wild Chimpanzees in the Tai' National Park," *Am. J. Phys. Anthropol.*, pp. 78547–573, 1989.

[26] P. E. Stander, "Cooperative Hunting in Lions: The Role of the Individual," *Source Behav. Ecol. Sociobiol. Behav Ecol Sociobiol*, vol. 29, no. 6, pp. 445–454, 1992.

[27] D. Biro, T. Sasaki, and S. J. Portugal, "Bringing a Time–Depth Perspective to Collective Animal Behaviour," *Trends Ecol. Evol.*, vol. 31, no. 7, pp. 550–562, Jul.

2016.

[28] M. E. Arnegard and B. A. Carlson, “Electric organ discharge patterns during group hunting by a mormyrid fish.,” *Proceedings. Biol. Sci.*, vol. 272, no. 1570, pp. 1305–14, Jul. 2005.

[29] R. Bshary, A. Hohner, K. Ait-el-Djoudi, and H. Fricke, “Interspecific Communicative and Coordinated Hunting between Groupers and Giant Moray Eels in the Red Sea,” *PLoS Biol.*, vol. 4, no. 12, p. e431, Dec. 2006.

[30] S. Kalyani and K. S. Swarup, “Particle swarm optimization based K-means clustering approach for security assessment in power systems,” *Expert Syst. Appl.*, vol. 38, no. 9, pp. 10839–10846, Sep. 2011.

[31] S. H. Al-Harbi and V. J. Rayward-Smith, “Adapting k-means for supervised clustering,” *Appl. Intell.*, vol. 24, no. 3, pp. 219–226, Jun. 2006.

[32] L. Kaufman and P. J. Rousseeuw, Eds., *Finding Groups in Data*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 1990.

[33] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, 1967, pp. 281–297.

[34] A. K. Jain, “Data clustering: 50 years beyond K-means,” *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010.

[35] A. Likas, N. Vlassis, and J. J. Verbeek, “The global k-means clustering algorithm,” *Pattern Recognit.*, vol. 36, no. 2, pp. 451–461, Feb. 2003.

[36] J. A. HARTIGAN and M. A. WONG, “A K-Means Clustering Algorithm,” *Source J. R. Stat. Soc. Ser. C (Applied Stat.*, vol. 28, no. 1, pp. 100–108, 1979.

[37] E. Forgy, “Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of Classification,” *Biometrics*, vol. 21, no. 3, pp. 768–769, 1965.

[38] S. Lloyd, “Least squares quantization in PCM,” *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.

[39] S. J. Redmond and C. Heneghan, “A method for initialising the K-means clustering algorithm using kd-trees,” *Pattern Recognit. Lett.*, vol. 28, no. 8, pp. 965–973, Jun. 2007.

[40] A. Chechkin, R. Metzler, J. Klafter, and V. Gonchar, “Introduction to the Theory of Lévy Flights,” *Anomalous Transport: Foundations and Applications*. pp. 129–162, 2008.

[41] C.-Y. Lee, X. Yao, Evolutionary algorithms with adaptive Levy mutations, in: Proceedings of the 2001 Congress on Evolutionary Computation, Seoul, South Korea, 2001, pp. 568-575.

[42] H. Hakli and H. U□uz, "A novel particle swarm optimization algorithm with Levy flight," *Appl. Soft Comput. J.*, vol. 23, pp. 333–345, 2014.

[43] X. S. Yang and S. Deb, "Multiobjective cuckoo search for design optimization," *Comput. Oper. Res.*, vol. 40, no. 6, pp. 1616–1624, 2013.

[44] G. Wang *et al.*, "Lévy-flight krill herd algorithm," *Math. Probl. Eng.*, vol. 2013, 2013.

[45] X.-S. Yang, *Engineering Optimization: An Introduction with Metaheuristic Applications*, 1st ed. Wiley, 2010.

[46] Watkins WA, Schevill WE. Aerial observation of feeding behavior in four baleen whales: *Eubalaena glacialis*, *Balaenoptera borealis*, *Megaptera novaean- gliae*, and *Balaenoptera physalus*. J Mammal 1979, pp.155-163.

[47] Goldbogen JA, Friedlaender AS, Calambokidis J, Mckenna MF, Simon M, Nowacek DP. Integrative approaches to the study of baleen whale diving be- havior, feeding performance, and foraging ecology. BioScience 2013, vol 63, pp. 90-100.

[48] K.D. Frank, C. Rich, T. Longcore, Effects of artificial night lighting on moths, in: Ecological Consequences of Artificial Night Lighting, 2006, pp. 305-344.

[49] M. Neshat and G. Sepidnam, "Swallow swarm optimization algorithm : a new method to optimization," pp. 429–454, 2013.

[50] M. Neshat, G. Sepidnam, M. Sargolzaei, and A. N. Toosi, "Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications," *Artif. Intell. Rev.*, vol. 42, no. 4, pp. 965–997, Dec. 2014.

[51] M. E. Centre, "Selection of K in K -means clustering," vol. 219, no. September 2004, pp. 103–119, 2005.

[52] F. Marini and B. Walczak, "Particle swarm optimization (PSO). A tutorial," *Chemom. Intell. Lab. Syst.*, vol. 149, pp. 153–165, Dec. 2015.

[54] S. García, D. Molina, M. Lozano, and F. Herrera, "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*, vol. 15, no. 6, pp. 617–644, Dec. 2009.

[55] F. Wilcoxon, "Individual Comparisons by Ranking Methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945.

[56] Y. Hochberg, “A sharper bonferroni procedure for multiple tests of significance,” *Biometrika*, 1988.

[57] R. A. Armstrong, “When to use the Bonferroni correction,” *Ophthalmic Physiol. Opt.*, vol. 34, no. 5, pp. 502–508, Sep. 2014.

[58] E. Sandgren, “Nonlinear Integer and Discrete Programming in Mechanical Design Optimization,” *J. Mech. Des.*, vol. 112, no. 2, p. 223, Jun. 1990.

[59] J. S. Arora and J. S. Arora, “Chapter 12 – Numerical Methods for Constrained Optimum Design,” in *Introduction to Optimum Design*, 2012, pp. 491–531.

[60] S. Das and P. Suganthan, “Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems,” 2018.

[61] J. Koski, “Defectiveness of weighting method in multicriterion optimization of structures,” *Commun. Appl. Numer. Methods*, vol. 1, no. 6, pp. 333–337, Nov. 1985.

Appendix A

The set of benchmark test functions implemented in the experiments is described in **Table AI**, where $f(\mathbf{x}^*)$ is the optimum value of the function, $\mathbf{x}^*$ the optimum position and $\mathbf{S} \in \mathbb{R}^n$ the search space. The benchmark test functions are classified in unimodal, multimodal and composite.

Name	Function	S	Dim	Minimum
Sphere	$f_1(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$	$[-5, 5]^n$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = 0,\ldots, 0$
Sum Squares	$f_2(\mathbf{x}) = \sum_{i=1}^{n} ix_i^2$	$[-10, 10]^n$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0,\ldots, 0)$
Sum of Different Powers	$f_3(\mathbf{x}) = \sum_{i=1}^{n} \lvert x_i \rvert^{i+1}$	$[-1, 1]^n$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0,\ldots, 0)$
Schwefel 2	$f_4(\mathbf{x}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_i \right)^2$	$[-100, 100]$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0,\ldots, 0)$
Rotated Hyper-Ellipsoid	$f_5(\mathbf{x}) = \sum_{i=1}^{n} \sum_{j=1}^{i} x_j^2$	$[-65.5, 65.5]$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0,\ldots, 0)$

Table AI. Unimodal test benchmark functions considered in the experiments.

Name	Function	S	Dim	Minimum

Ackley	$f_6(\mathbf{x}) =- 20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)} + 20 + e$	$[-30, 30]^n$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0,\ldots, 0)$
Powell	$f_7(\mathbf{x}) = \sum_{i=1}^{n/4}\left[(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-}\right.$	$[-4, 5]^n$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0,\ldots, 0)$
Rosenbrock	$f_8(\mathbf{x}) = \sum_{i=1}^{n-1}\left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$	$[-5, 10]^n$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (1,\ldots, 1)$
Schwefel 22	$f_9(\mathbf{x}) = \sum_{i=1}^{n}\lvert x_i\rvert + \prod_{i=1}^{n}\lvert x_i\rvert$	$[-100, 100]^r$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0,\ldots, 0)$
Perm2	$f_{10}(\mathbf{x}) = \sum_{i=1}^{n}\left[\sum_{j=1}^{n}(j + \beta)\left({x_j}^i - \frac{1}{j}\right)\right]^2$	$[-n, n]^n$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = \left(1, \frac{1}{2},\ldots,\frac{1}{n}\right)$
Dixon Price	$f_{11}(\mathbf{x}) = (x_i - 1)^2 + \sum_{i=2}^{n} i(2x_i^2 - x_{i-1})^2$	$[-10, 10]^n$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = 2^{-\frac{2^i - 2}{2^i}}$ for $i = 1,\ldots,n$
Styblinski Tang	$f_{12}(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{n}\left(x_i^4 - 16x_i^2 + 5x_i\right)$	$[-5, 5]^n$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) =- 39.1659n;$ $\mathbf{x}^* = (-2.90,\ldots, 2.90)$
Quartic	$f_{13}(\mathbf{x}) = \sum_{i=1}^{n}\left\{(ix_i)^4 + rand[0,1)\right\}$	$[-1.28, 1.28]$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0,\ldots, 0)$
Salomon	$f_{14}(\mathbf{x}) = 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^{n} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{n} x_i^2}$	$[-100, 100]^n$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0,\ldots, 0)$
Penalty1	$f_{15}(\mathbf{x}) = \frac{\pi}{30}\left\{\begin{matrix} 10\sin^2(\pi y_1) \\ + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_i + 1)] \\ + (y_n - 1)^2 \end{matrix}\right\} + \sum_{i:}^{n}$ $y_i = 1 + \frac{x_i + 1}{4};$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \le x_i \le a \\ k(-x_i - a)^m, & x_i <- a \end{cases}$	$[-50, 50]^n$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (-1,\ldots, -1)$
Penalty2	$f_{16}(\mathbf{x}) = 0.1\left\{\begin{matrix} \sin^2(3\pi x_1) \\ + \sum_{i=1}^{n-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] \\ + (x_n - 1)^2[1 + \sin^2(2\pi x_n)] \end{matrix}\right\} + \sum_{i=1}^{n}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \le x_i \le a \\ k(-x_i - a)^m, & x_i <- a \end{cases}$	$[-50, 50]^n$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (1,\ldots, 1)$
Rastrigin	$f_{17}(\mathbf{x}) = 10n + \sum_{i=1}^{n}\left[{x_i}^2 - 10\cos(2\pi x_i)\right]$	$[-5.12, 5.12]$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0,\ldots, 0)$
Mishra1	$f_{18}(\mathbf{x}) = (1 + x_n)^{x_n}; \quad x_n = n - \sum_{i=1}^{n-1} x_i$	$[0, 1]^n$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = 2;$ $\mathbf{x}^* = (1,\ldots, 1)$
Mishra2	$f_{19}(\mathbf{x}) = (1 + x_n)^{x_n}; \quad x_n = n - \sum_{i=1}^{n-1}\frac{(x_i + x_{i+1})}{2}$	$[0, 1]^n$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = 2;$ $\mathbf{x}^* = (1,\ldots, 1)$

Mishra11	$f_{20}(\mathbf{x}) = \left[\frac{1}{n}\sum_{i=1}^{n} \lvert x_i \rvert - \left(\prod_{i=1}^{n} \lvert x_i \rvert\right)^{\frac{1}{n}}\right]^2$	$[-10, 10]^n$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0,\ldots,0)$
Vincent	$f_{21}(\mathbf{x}) = -\sum_{i=1}^{n} \sin(10\log x_i)$	$[0.25, 10]^n$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = -n;$ $\mathbf{x}^* = (7.70,\ldots,7.70)$
Infinity	$f_{22}(\mathbf{x}) = \sum_{i=1}^{n} x_i^6\left[\sin\left(\frac{1}{x_{\mathrm{i}}}\right) + 2\right]$	$[-1, 1]^n$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0,\ldots,0)$
Griewank	$f_{23}(\mathbf{x}) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^r$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0,\ldots,0)$

Table AII. Multimodal test benchmark functions considered in the experiments.

Name	**Function**	**S**	**Dim**	**Minimum**
Fx24	$f_{24}(\mathbf{x}) = f_1(\mathbf{x}) + f_9(\mathbf{x}) + f_{17}(\mathbf{x})$	$[-100, 100]^r$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0,\ldots,0)$
Fx25	$f_{25}(\mathbf{x}) = f_8(\mathbf{x}) + f_{17}(\mathbf{x}) + f_{23}(\mathbf{x})$	$[-100, 100]^r$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = n - 1;$ $\mathbf{x}^* = (0,\ldots,0)$
Fx26	$f_{26}(\mathbf{x}) = f_4(\mathbf{x}) + f_6(\mathbf{x}) + f_8(\mathbf{x}) + f_{16}(\mathbf{x})$	$[-100,100]^n$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = (1.1n) - 1;$ $\mathbf{x}^* = (0,\ldots,0)$
Fx27	$f_{27}(\mathbf{x}) = f_6(\mathbf{x}) + f_8(\mathbf{x}) + f_9(\mathbf{x}) + f_{17}(\mathbf{x}) + f_{23}(\mathbf{x})$	$[-100,100]^n$	$n = 30$ $n = 50$ $n = 100$	$f(\mathbf{x}^*) = n - 1;$ $\mathbf{x}^* = (0,\ldots,0)$

Table AIII. Composite test benchmark functions considered in the experiments.

Appendix B

In this section, the engineering optimization problems considered in the experiments are described in detail.

B.1 Gear train problem

Given a gear train, as shown in **Figure BI**, it is required to minimize the squared difference between the teeth ratio of the gear and a given scalar value. The decision variables are the number of teeth corresponding to each gear. Labels A, B, D and F are used to identified the gears. The decision variables are $x_1 = A$, $x_2 = B$, $x_3 = D$ and $x_4 = F$. The scalar value is $1/6.931$. The cost function and constraints are defined as follows:

Problem B1: gear train

Minimize:

$$f_{B1}(\mathbf{x}) = \left(\frac{1}{6.931} - \frac{x_3 x_2}{x_1 x_4}\right)^2$$

Subject to:

$0 \leq x_i \leq 600, \quad i = 1, 2, 3, 4$

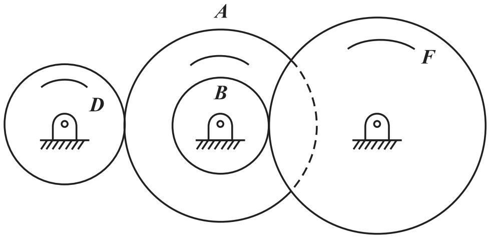

Figure BI. Gear train design.

B.2 Spring problem

The problem is to minimize the tension or compression experienced by a spring when a load P is applied. For optimization, the wire diameter d, the coil diameter D and the number of active coils n are considered. The decision variables are $x_1 = d$, $x_2 = D$ and $x_3 = n$, see **Figure BII**. The design problem is formulated as:

Problem B2: spring

Minimize:

$$f_{B2}(\mathbf{x}) = (x_3 + 2)x_2x_1^2$$

Subject to:

$$g_1(\mathbf{x}) = 1 - \frac{x_2^3x_3}{71{,}785x_1^4} \leq 0$$

$$g_2(\mathbf{x}) = \frac{4x_2^2 - x_1x_2}{12{,}566(x_2x_1^3 - x_1^4)} + \frac{1}{5{,}108x_1^2} - 1 \leq 0$$

$$g_3(\mathbf{x}) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0$$

$$g_4(\mathbf{x}) = \frac{x_1 + x_2}{1.5} \leq 0$$

$$0.5 \leq x_1 \leq 2$$

$$0.25 \leq x_2 \leq 1.3$$

$$2 \leq x_3 \leq 15$$

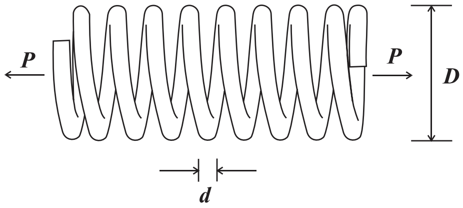

***Figure BII**:* Spring design.

B.3 Pressure vessel problem

The design of a pressure vessel is required to minimize the material used for its construction. The optimization problem must consider the thickness of the shell T_s, the thickness of the head T_h, internal radius of the vessel R, and the length of the vessel L, see **Figure BIII**. The decision variables are $\mathbf{x} = [x_1,x_2,x_3,x_4]$ were $x_1 = T_s$, $x_2 = T_h$, $x_3 = R$ and $x_4 = L$. The cost function and constraints are defined as follows:

Problem B3: pressure vessel

Minimize:

$$f_{B3}(\mathbf{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

Subject to:

$$g_1(\mathbf{x}) = -x_1 + 0.0193x_3 \leq 0$$

$$g_2(\mathbf{x}) = -x_2 + 0.00954x_3 \leq 0$$

$$g_3(\mathbf{x}) = -\pi x_3^2x_4 - (4/3)\pi x_3^2 + 1{,}296{,}000 \leq 0$$

$$g_4(\mathbf{x}) = x_4 - 240 \leq 0$$

$$0 \leq x_i \leq 100, \;\; i = 1{,}2$$

$$0 \leq x_i \leq 200, \;\; i = 3{,}4$$

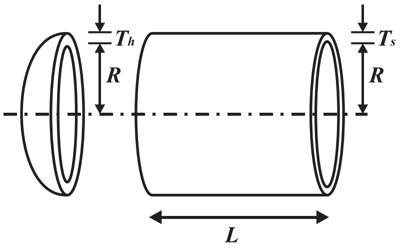

***Figure BIII**:* Pressure vessel design.

B.4 FM synthesizer problem

An FM synthesizer generates a signal $y(\mathbf{x},t)$ similar to a target signal $y_0(t)$. To minimize the error between the signal and the target signal, a parameter estimator for the FM synthesizer is designed considering a finite wave amplitude a_i and the frequency ω_i. The decision variables are $\mathbf{x} = [x_1 = a_1,x_2 = \omega_1,x_3 = a_2,x_4 = \omega_2,x_5 = a_3,x_6 = \omega_3]$. The cost function and constraints are defined as follows:

Problem B4: FM synthesizer

Minimize:

$$f_{B4}(\mathbf{x}) = \sum_{t=0}^{100} \left(y(\mathbf{x},t) - y_0(t)\right)^2$$

$y(\mathbf{x},t) = x_1\sin(x_2\theta t) + x_3\sin(x_4\theta t) + x_5\sin(x_6\theta t)$

Subject to:

$-6.4 \leq x_i \leq 6.35, \quad i = 1,2,3,4,5,6$

$\theta = \frac{2\pi}{100}$

B.5 Three bar truss problem

The manufacturing cost of a three-bar truss, subject to a load P and a stress σ, must be minimized. The decision variables are the cross-sectional areas x_1, x_2 and x_3 shown in **Figure BIII**, where $x_3 = x_1$ due to the design symmetry. The cost function and constraints are defined as follows:

Problem B5: Three bar truss

Minimize:

$f_{B5}(\mathbf{x}) = l\left(2\sqrt{2x_1} + x_2\right)$

Subject to:

$$g_1(\mathbf{x}) = \frac{\sqrt{2x_1} + x_2}{\sqrt{2x_1^2 + 2x_1x_2}}P - \sigma \leq 0$$

$$g_2(\mathbf{x}) = \frac{x_2}{\sqrt{2x_1^2 + 2x_1x_2}}P - \sigma \leq 0$$

$$g_3(\mathbf{x}) = \frac{1}{\sqrt{2x_2^2 + x_1}}P - \sigma \leq 0$$

$0 \leq x_i \leq 1, \quad i = 1,2$

$l = 100$ cm, $P = 2$ kN/cm^2, $\sigma = 2$ kN/cm^2

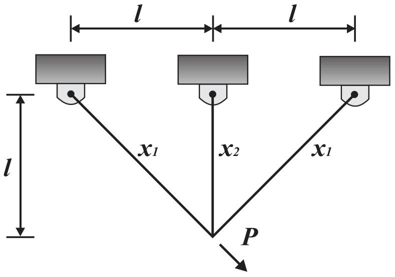

Figure BIV: Three-bar truss design.