Applied Soft Computing Journal 88 (2020) 106039

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Side-Blotched Lizard Algorithm: A polymorphic population approach

Oscar Maciel C. [a], Erik Cuevas [a], Mario A. Navarro [a], Daniel Zaldívar [a], Salvador Hinojosa [b,*]

[a] División de Electrónica y Computación, Universidad de Guadalajara, CUCEI, Av. Revolución 1500, Guadalajara, Jal, Mexico
[b] Dpto. Ingeniería del Software e Inteligencia Artificial, Facultad Informática, Universidad Complutense de Madrid, 28040 Madrid, Spain

ARTICLE INFO

Article history:
Received 21 June 2019
Received in revised form 15 December 2019
Accepted 16 December 2019
Available online 30 December 2019

Keywords:
Optimization
Metaheuristics
Bio-inspired computation

ABSTRACT

In metaheuristic algorithms, finding the optimal balance between exploration and exploitation is a key research topic that remains open. In the nature, a reptile called Side-Blotched Lizard has achieved an interesting dynamic balance over its population. Such lizards evolved with three morphs associated with distinctive mating strategies. The synergy between the morphs generates a polymorphic population, able to self-balance the subpopulations of each morph without depleting the weakest morph. This equilibrium is achieved as the most common morph becomes the weakest, and the smaller subpopulations increase their chances of mating. In this paper, the Side-Blotched Lizard Algorithm (SBLA) is proposed to emulate the polymorphic population of the lizard. For this purpose, three operators are used to guarantee a dynamic over the population that allows the coexistence of multiple morphs. From the computational point, SBLA uses a subpopulation managing strategy which emulates the sinusoidal distribution of the lizard population over time. Even more, the mating behavior of each morph is modeled with three concepts, defensive, expansive, and sneaky. The performance of SBLA is tested on a set of five unimodal, eighteen multimodal, four composite benchmark functions, and engineering problems like; the welded beam, FM synthesizer, and rolling element bearing. To validate the results, we compared them to ten well-established algorithms and using the Wilcoxon test and the Bonferroni correction. The examination of the experimental results exhibits the accuracy, robustness and unique problem-solving method of the proposed algorithm.

1. Introduction

Global optimization comprises a large collection of methods designed to find the global minima or maxima for a given problem, where an objective function is used to model the problem at hand. However, many real-life applications cannot be solved using classical optimization techniques due to their properties and complexity. Metaheuristic optimization algorithms are designed as stochastic methods devoted to identifying near-optimal solutions on complex problems; their stochastic and derivative-free nature often prevents them from being stuck in local optima [1,2].

In the last decade, different optimization algorithms based on metaheuristic principles have been proposed with interesting results. Such schemes consider models extracted from natural or social systems, which at some abstraction level can be conceived as search strategies. Nature-inspired algorithms are popular due to their success in problem-solving, however, one algorithm cannot solve all optimization problems, for this reason, scientists search inspiration to make new and improved algorithms, from small-sized invertebrates as the ant colony optimization [3], to the largest animal as the whale optimization algorithm [4], a standard classification is to divide them in evolution-based, physics-based, human-based and swarm-based optimization algorithms [5].

Evolution-based algorithms combine their particles to obtain similar solutions on the next generation, mutate to get a new solution and compares them to select the best individuals to keep evolving; the most popular evolutionary techniques are the evolutionary strategy [6], genetic algorithms [7], and differential evolution [8]; even if evolution-based algorithms have multiple particles, our proposal will fall in the classification of a population/swarm-based algorithms, these algorithms have techniques that draw inspiration of the collective behavior of one or more groups of species. Under these models, evolutionary mechanisms such mutation, reproduction, selection and recombination, are contemplated in this kind of approaches. Representative examples of evolution-inspired methods are Differential Evolution (DE) [8], Genetic Algorithms (GAs) [9], Evolutionary Strategies (ES) [6,10,11] and Self-Adaptive Differential Evolution (JADE) [12].

The physics-based approaches are methods designed to emulate physic phenomena by replicating the laws of physics. In this category, the Simulated Annealing algorithm can be found, which contributed to the concept of linking the temperature to how

* Corresponding author.
E-mail addresses: oscar.maciel@alumno.udg.mx (O. Maciel C.), erik.cuevas@cucei.udg.mx (E. Cuevas), marioa.navarro@alumno.udg.mx (M.A. Navarro), daniel.zaldivar@cucei.udg.mx (D. Zaldívar), salvahin@ucm.es (S. Hinojosa).

https://doi.org/10.1016/j.asoc.2019.106039
1568-4946/

wide a local search is conducted, while iteratively cooling down the system to refine the search [13]. However, one of the most notorious approaches in this categorify is the Electromagnetism-like optimizer, proposed by Birbil and Fang in 2003 [14] and inspired by the Coulomb's law generating attraction and repulsion forces among charged particles. Later, in 2009, Rashedi et al. [15] proposed the Gravitational Search Algorithm (GSA) inspired by the gravitational forces exerted by an object mass. Some other relevant approaches include the States of Matter Search (SMS) and the Sine Cosine Algorithm [16,17]. The most popular physics-inspired methods include [18–20], Gravitational Search Algorithm (GSA) [15], the Simulated Annealing (SA) method [18–20], Big Bang-Big Crunch (BB-BC) [21], Electromagnetism-like Mechanism (EM) [14], States of Matter Search (SMS) [16,22], Water Cycle Algorithm (WCA) [23] and Ray Optimization (RO) [24].

Human-based algorithms are classified as optimization methodologies whose design is inspired by diverse social phenomena related to human behavior and lifestyle; these techniques are based on various cognitive processes or methodologies used by humans to solve problems or perform everyday activities. The harmony search [25], proposed by Geem in 2001, is a metaheuristic approach inspired on the principles behind the process of harmony improvisation in Jazz. Here, musicians are said to compose a harmony by combining different music pitches stored in their memory representing candidate solutions, with the purpose of finding the perfect harmony that symbolizes the optimal solution. Atashpaz-Gargari and Lucas in 2007 proposed a novel population-based metaheuristic known as Imperialist Competitive Algorithm [26]. Said optimization technique is inspired by the actions taken by individual countries to extend their power through the acquisition of other territories. The most popular algorithms from this category also include League Championship Algorithm (LCA) [27], Social mimic optimization algorithm (SMO) [28], and Teaching Learning-Based Optimization (TLBO) [29].

Some examples of the swarm-based algorithms are; the particle swarm optimization by Kennedy and Eberhart in 1995 [30], which took inspiration on the behavior of flocking birds foraging for food sources; the ant colony optimization proposed by Dorigo and Blum in 2005 [3] that considers several ants moving through, based on a pheromone density associated to them, thus giving place to a backtracking method; cuckoo search by Yang and Deb in 2009 [31] which modeled the brood parasitism behavior of the cuckoo bird, the way it works is by considering the solution as a nest or eggs. The cuckoos have a probability to fly towards and lay their own eggs, then the solution can be abandoned to represent the behavior of other birds when discovering that the replacement took place, each of these are population-based algorithms. Thus, swarm-based methods correspond to schemes based on the cooperative behavior presented by animals or insects groups. Other relevant swarm-based methods include, Firefly Algorithm (FA) [32–34], Best-so-far Artificial Bee Colony (BSF ABC) [35], , Gray Wolf Optimizer (GWO) [36], Crow Search Algorithm (CSA) [37,38], Locust Search (LS) [39,40], Whale Optimization Algorithm (WOA) [4], Bat Algorithm (BA) [41], Ant Colony Optimization (ACO) [42] and Moth Flame Optimization (MFO) algorithm [43], to name a few.

Within the swarm-based algorithms, there are methods with multiple populations; for example, a method inspired by the behavior of bees named artificial bee colony was proposed by Karaboga and Basturk in 2008 [44], in this approach the search agents are divided in three groups, employed bees, onlooker bees and scout bees. Each group has their function and can share their information with other groups; other algorithm that uses multiple populations is the Social Spider Optimization proposed in 2013 [45], it divides the populations on female and male spiders, each with different behaviors.

As the no free lunch theorem states [46], there is not a perfect algorithm, even if A outperforms algorithm B in some problem, there will be another problem in which B could get better results than A. There has been several applications for metaheuristics with different algorithms applied for problem-solving, either being benchmark [47,48], binary [49], combinatorial [50], dynamic optimization [51], proving that it is necessary to keep making and improving algorithms to solve a broad scope of problems.

The general methodology behind the design of swarm-based methods is the encoding of an animal or insect mechanism into a highly abstracted search strategy. In spite of the diverse number of metaphors, the algorithmic schemes of some bio-inspired methods do not differ substantially among them [52]. Therefore, there exist some cases in which complete different metaphors correspond to metaheuristic schemes that diverge in only a few elements. On the other hand, there are other cases where the same metaphor is implemented in a completely different metaheuristic structure. Under such circumstances, however, it has been demonstrated that the inclusion of a new operation in certain existent metaphor can produce very different results [53]. Assuming such critics, it is also acknowledged that bio-inspiration represent one of the main progress in optimization algorithms. From it, several powerful search mechanisms have been proposed such as crossover and mutation operations from GA [9], swarm movement [30] from PSO, levy flight displacements [31] from CS, probabilistic path tracking [42] from ACO, to name a few.

Metaheuristic algorithms use a combination of operators designed to provide an effective search strategy. Most operators present on bio-inspired or even metaheuristic algorithms can fall into one of two categories; exploration and exploitation [5]. The operators of the first category aim to explore the search space to promote the diversity of the solutions, while the exploitation operators intensify the search on promising regions. The success of a metaheuristic algorithm is mainly dependent on the balance of exploration and exploitation [54]. Unfortunately, an optimal balance is not easy to find since it is problem-dependent and remains as an open question in the community. One way to address this issue is the use of dynamical subpopulations, where each subpopulation presents specific behaviors designed to intensify or diversify the exploration over promising areas [55–57]. During the iterative process the subpopulations can either compete or collaborate to find the optimum solution [58]. Furthermore, some authors have followed the differentiation of the subpopulations according to biological traits such as gender, size or rarity [59].

The peculiar conduct of animal groups in nature such as colonies of ants, flocks of birds, schools of fish, groups of lizards and swarms of bees have attracted the attention of researchers. Biologists have studied this cooperative phenomenon to model biological schemes while scientists have used these models as approaches for solving real-world problems. One of these models used to manage the prevalence of subpopulations of the same species can be found on the Side-Blotched Lizard. The most relevant trait of this kind of Lizards is the presence of three morphs in the same species [60]. Commonly, only the fittest morph survives and passes its genes to next generations. However, Side-Blotched Lizards have achieved an atypical equilibrium between morphs (subpopulations) in nature. The population of male lizards is divided into three subpopulations of blue, orange, and yellow lizards. Their mating behaviors differ significantly according to their morph. As a result, the number of lizards at every subpopulation vary according to the most effective mating strategy at the time following a cycle. The cycle starts with one morph successfully growing its subpopulation until it becomes vulnerable to the mating strategy of another morph. In this sense, all three variants of the lizard will become the most numerous subpopulations to later decrease its numbers to a minimum, and then the cycle starts over again. The number of lizards of each morph at every generation maintains a sinusoidal behavior while achieving a remarkable polymorphic population.

In this paper, a bio-inspired method called the Side-Blotched Lizards Algorithm (SBLA) is introduced. The SBLA uses as a metaphor the polymorphic changes over the population showed by the Side-Blotched lizards. Each morph emulates a specific mating behavior; with every morph, the ratio between exploration and exploitation is changed aiming to enhance the search strategy. The equilibrium between subpopulations is achieved by incorporating three basic operations over the population according to the number of elements of each morph. The search agents (lizards) can be either created, eliminated, or transformed following a set of simple rules. Moreover, following the metaphor three operators are used on the SBLA to move the search agents around the solution space. The adopted search strategies or behaviors can be summarized as expansive, defensive, and sneaky. The synergy between operators and the dynamic subpopulation control allows presenting a robust algorithm suitable for global optimization tasks. The performance of the proposed SBLA is compared against ten different metaheuristic algorithms which consider bio-inspired, evolution-inspired, physics-inspired and social-inspired techniques. They include the Differential evolution (DE) [8], Simulated annealing (SA) [61], Harmony search (HS) [25], artificial bee colony (ABC) [62], Particle search optimizer (PSO) [30], Bat algorithm (BA) [41], Firefly algorithm (FA) [63], Moth flame optimization (MFO) [43], Crow search algorithm (CSA) [64] and Gray Wolf optimizer (GWO) [36]. The comparison takes place over twenty-seven benchmark functions including unimodal, multimodal, composite functions over 30, 50, and 100 dimensions. Furthermore, to analyze the performance of the proposed approach over real-life optimization tasks, three interesting engineering problems are evaluated and compared such as the FM synthesizer design, rolling element bearing, and welded beam. The significance of the entire set of experiments is statistically analyzed using the Wilcoxon test and applying the Bonferroni correction. Experimental results demonstrate the high performance of the proposed approach in terms of accuracy and robustness.

This paper is organized as follows: In Section 2, details and preliminary concepts are introduced. In Section 3, the Side-Blotched Lizard Algorithm (SBLA) is exposed. In Section 4, the experimental results and statistical analysis over the benchmark functions are presented. In Section 5, the experimental results and the comparative study over a set of relevant engineering problems are presented. Finally, in Section 6, conclusions are drawn.

2. Preliminary concepts

The side-blotched lizard (*Uta stansburiana*), is a specie of little lizards located on different states west of United States and Mexico. They are a remarkable example of a stable evolutionary mating model with three morphs from the same species. In nature, many species display polymorphisms associated with their reproduction strategies; biologists compare the case of the side-blotched lizard with the game of rock–paper–scissors, where the rock crushes scissors, the scissors cut paper, and the paper wraps the rock generating the closed cycle that characterizes the game. This set of rules is similar to the roles of the three colors morphs, where each morph uses a different strategy to mate, competing through a mechanism where an individual has advantages over another, and the third outweighs the first. The interaction of this species is a challenge to the evolutionary theory, because the fittest strategy should prevail, unless the morphs within have exactly the same ability to perform adequately in a habitat. Fig. 1 shows the different morphs on the male lizards and their cycles, we can easily note a distinction of color looking at their throats and they seem to be able to differentiate between them, their defense mechanism varies depending on their color polymorphism.

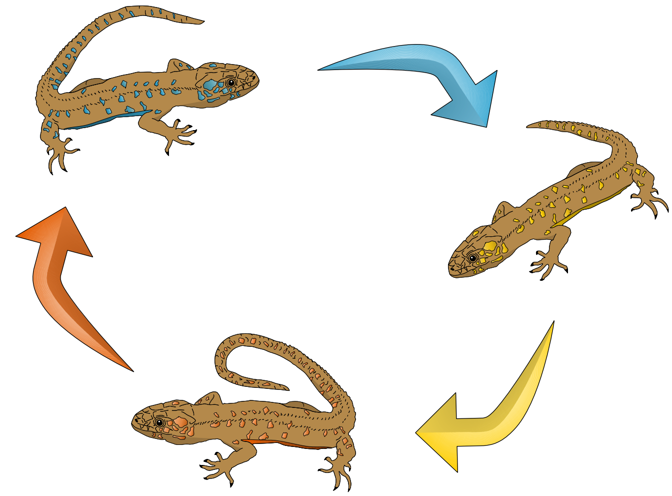

Fig. 1. Dynamic of the population.

The fascinating complexity of nature involves intricate details on real-life situations which are difficult to accurately model. Thus, a rather simple version of the general behavior is described as follows. Side-blotched lizards have three different mating behaviors depending on the distinct polymorphism in their throat colors. Orange males are the biggest and most aggressive. They hold vast territories with several females each and can force out the smaller and less aggressive blues. Blue males typically occupy smaller areas and are monogamous, focusing his interest in just one female. Yellow males do not establish exclusive territories; instead they stealth to find unaccompanied females with whom to mate, which makes them especially successful with females that live in territories held by the orange competitors as their attention spread among several females. And the blues males are more cautious with their female and can keep sneaky yellow males away. As described above, each morph has advantages and disadvantages on the mating war depending on who is leading, which induces changes in the prevailing class [60,65,66]. The result is a cycle with a dynamic balance on the mating strategies that has lasted many years. The dynamic balance on the population between the strategies and morphs creates a cycle where the least common morph of one breeding season has the biggest offspring in the next season.

The mating behavior can be summarized on the following points.

- Orange throated males are highly dominant, they are the largest and the most aggressive of them all; they defend large territories with lots of females, comparable to harems, they also steal the mate of the blue lizard imposing strength, but they are vulnerable to the slippery yellow lizards.
- The blue throated males are dominant, they have and intermediate size and guard smaller territories containing a single female; they are better protecting them from yellow throated lizards, but are susceptible to having their females stolen by the orange lizards.
- The males with a yellow hue, imitates behaviors of the females to go unnoticed to the orange lizards, in contrast to the others morphs, this morph does no possess defined territories, they locate in the suburbs of orange lizards to steal neglected females.

3. Side-Blotched Lizard Algorithm (SBLA)

In this paper, the stability of the population of the side-blotched lizard, and fluctuation of each color polymorphism is used as a guidance to develop a novel bio-inspired subpopulation-based optimization algorithm. Said algorithm emulates the total and partial populations of lizards, taking into consideration how the distribution of each morph gradually changes as time goes by. The strategy takes into consideration three polymorphism or subpopulations, blue, orange, and yellow lizards, but they only take place after it generates the initial population. Each morph has their own set of rules, examples are, the rule to add a particle and the strategy to search the space, this is in behalf of their nature, since each color has their own behavior and way of surviving. In each iteration, the number of individuals in each subpopulation will change, but the general population of lizards will stay consistent through the iterative process (see Fig. 2).

3.1. Initialization

A population P of TPOP lizards (individuals) $\{x_1, x_2, \ldots, x_n\}$ generates randomly and uniformly distributed solutions between the lower LB^d and upper UB^d boundaries for each dimension d in the D-dimensional search space, where $TPOP$ is population size and $x_1 \in P$ is a vector of decision variables defined as $x_i = \{x_i^1, x_i^2, \ldots, x_i^D\}$ being D the number of dimensions of the problem. The initialization formula is:

$$x_i^j = rand * \left(UB^j - LB^j\right) + LB^j \tag{1}$$

$$i = 1, 2, \ldots, TPOP; \quad j = 1, 2, \ldots, d. \tag{2}$$

where rand is a uniformly distributed random number between [0,1], this is the first step, which makes the initial population of lizards to have a position in the search space which is defined as a vector that goes from $\left[LB^1, LB^2 \ldots LB^D\right]$ *to* $[UB^1, UB^2 \ldots UB^D]$ in which each dimension $1 \ldots D$ has an upper and lower limit.

The lizard's rock–paper–scissors evolutionary game serves to maintain its biological diversity [67], and the SBLA takes it as base to make a subpopulations model that reflects the changes on each polymorphism while keeping the general population of the species.

3.2. Subpopulation model

The function takes as input the total population $TPOP$, the number of seasons S and the number of iterations $iter$, and returns the quantity of individuals of each subpopulation through every iteration PB, PO *and* PY (blue, orange, and yellow populations). To avoid the extinction of a morph, a minimum percentage of the population is preserved ($MinP$ of 10%).

To represent a full cycle, we generate a frequency of $\pi * 2$, the maximum population possible for a subpopulation is $MaxP = (TPOP * 2/subpop) - MinP$ where $subpop = 3$ and represents the number of subpopulations. To emulate the cyclic behavior of the polymorphic populations a sine function represent the population through time, generated using a vector of points X that goes from 0 to the max iteration with a spacing given by $Freq * S$, also two variables given by $Aux1 = \left(\frac{MaxP}{2}\right) + \left(\frac{MinP}{2}\right)$ and $Aux2 = \left(\frac{MaxP}{2}\right) - \left(\frac{MinP}{2}\right)$.

Fig. 3 shows a cycle in each subpopulation, represented by a full sine, the purple line represents the sum of populations, notice that it is the same at every iteration, and the green line represent the minimum population allowed.

Then to generate every subpopulation obtaining their population each iteration ($PIter$):

$$PIter = Aux1 + Aux2 * \sin\left(X + \left(\frac{Freq}{subpop} * i\right)\right) \tag{3}$$

where i is an index that goes from 1 to $subpop$, in this case from 1 to 3, rounded to represent a full particle or lizard accurately, once the populations are generated, they are added on a variable that will be used to control that in every iteration the sum of populations is the same as the total population $TPOP$, this is where the iterative process begins.

In Fig. 4 it can be observed how the frequency on the algorithm makes each cycle repeat regularity, ending in the same place where it begins, this allows each subpopulation to show their proprieties periodically, it also shows the pink line which represent the population that the algorithm must maintain. If the sum of the populations is higher than TPOP, it will subtract one particle from the subpopulation with the most particles, if the sum is smaller, it will add a particle from the subpopulation with least particles, if there is a tie the subpopulations will be selected at random.

To promote the diversity of solutions, a random shift is applied in the sines allowing to start the iteration process with any of the morphs dominating, like showed in Fig. 5, where the displacement is of 48 iterations to the right, and those are the new values used in each iteration, providing a natural start to the populations instead of their static counterpart. Once gotten each value of the subpopulations for each iteration, we generate the first population colorless and depending on their fitness value; then, the algorithm assigns them color. To accelerate the convergence of the algorithm, no matter the number of lizards of each morph, blues will initialize with the best fitness, then orange, and finally yellow.

3.3. Population changes

Once the initial population is generated with their respective fitness and the subpopulation colors are assigned, the iterative search process can start; this will go from 1 to the maximum iteration number.

The first step is to get the period of the population and the changes of population made on the iteration by taking the next point on the sine, by checking the current iteration in each vector of subpopulations. There are *three* types of seasons depending on who is the predominant morph, and this let us perform the rock–paper–scissors behavior of the lizards.

To get the changes on the population ΔP the algorithm subtracts from each vector of subpopulation PIter^i the value on the last element PIter^{i-1} for each color and save it in a population changes vector.

$$\Delta P = P^i - P^{i-1} \tag{4}$$

Once known the period and changes of population, it is time to perform them, this thanks to the delete, transform and add lizard functions explained below.

(A) Delete function

The algorithm calls the Delete lizard function when there is a negative change on the population of a morph, and there is no positive change on the one that has an advantage over it.

It gets every lizard in the population that matches the color to eliminate, once got, it sorts them by fitness in an ascending order, and it sets the worst value corresponding to the first index to null on the population array, once deleted the variable of population changes on that specific color will decrease by one and the process will continue until the variable reaches zero,

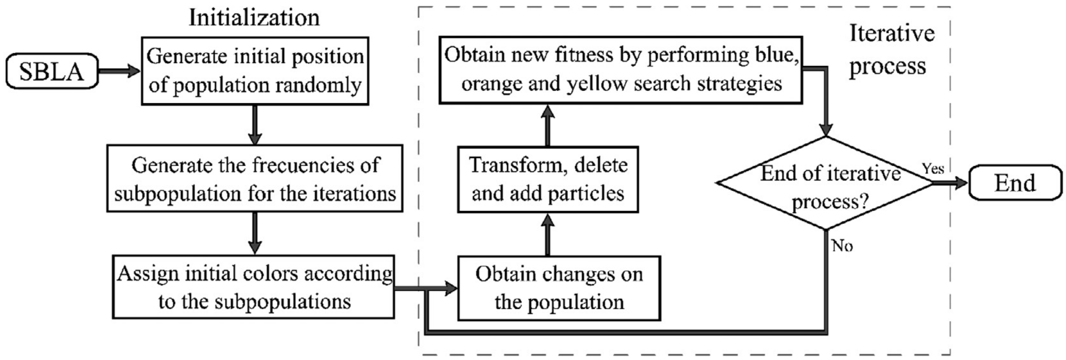

Fig. 2. Flowchart of SBLA.

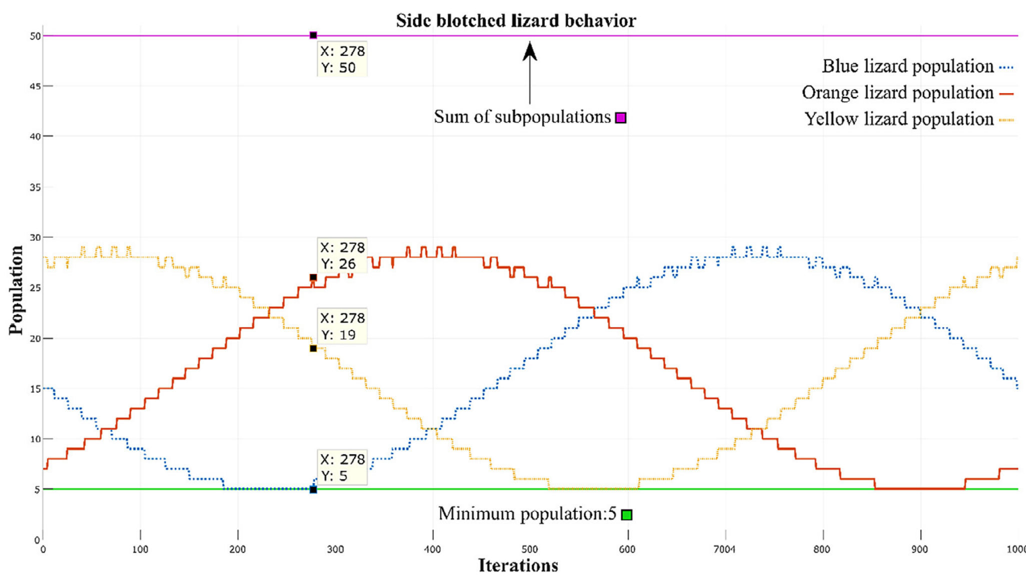

Fig. 3. Representation of the population.

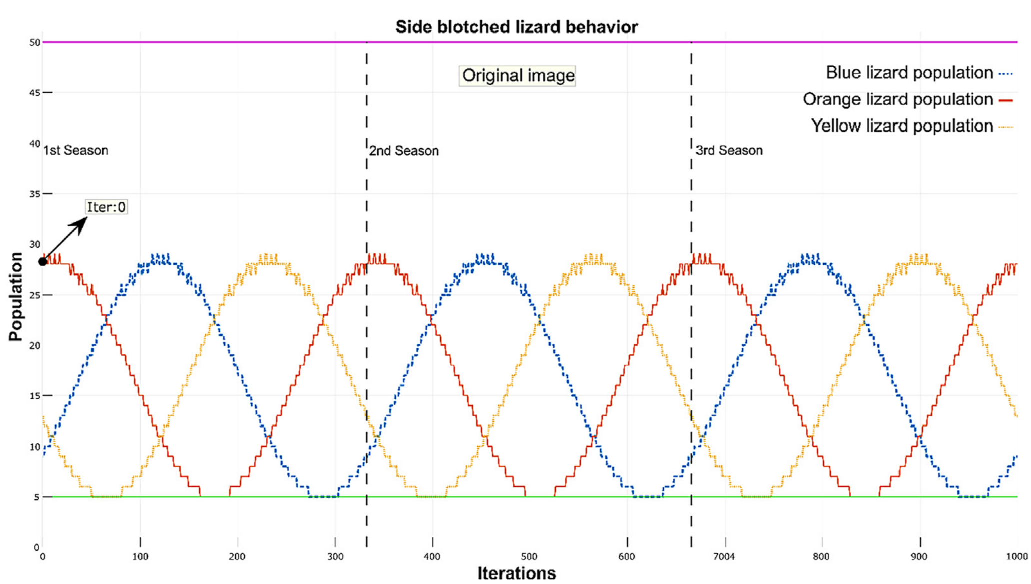

Fig. 4. Population with a frequency of 3.

Fig. 6 shows the delete lizard function, in which a yellow lizard disappear from the population.

(B) Transform function

When the changes in the population of the color with the predominant population has a positive change and the one afflicted by it has a negative one the algorithm performs the transform

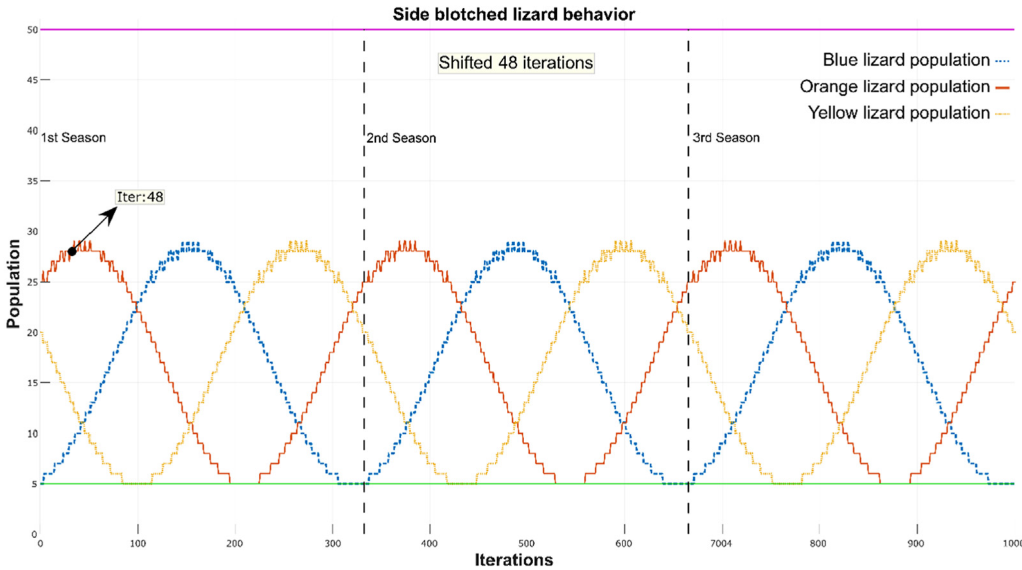

Fig. 5. The population shifted by 48 iterations.

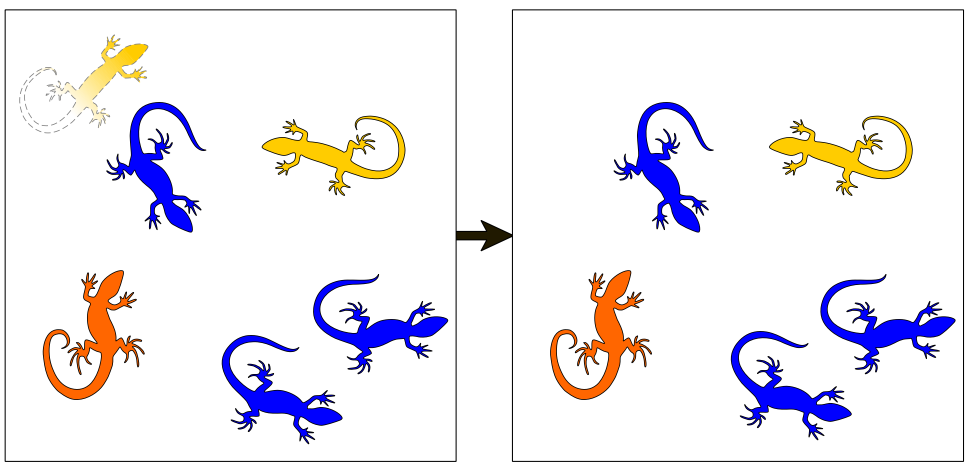

Fig. 6. Delete Lizard Function.

lizard function, for example, when is the orange season, the blue lizards will use the transform function each time their population decreases, and orange population increases.

The index of the worst particle of the affected population must be found and their color must be changed for the one that is affecting it, once done, the array of population changes increase by one on the population that grew and the other one will decrease.

Fig. 7 represents the transform lizard function, where a blue lizard changes color to orange saving its previous position.

(C) Add function

If the changes on the population index are positive for one population that does not prevail in the ongoing season or if the one affected by it does not have a negative change a new particle, then the algorithm will add it in the population, this method is different for each morph or subpopulation and detailed below.

The first requirement to add a lizard is to create a new position in the array with the color matching the morph of the lizard. As an example, with an orange lizard, it is necessary to seek for two blue lizards, that has the first and second best fitness, then we choose at random a proportion from 0 to 1 *randP*, this represent a position between the first lizard and the second lizard, blue lizards use the yellow instead, and to add a yellow lizard we need to select a position between the first and second best orange lizards, detailed below is the general formula to add a new lizard *NL*:

$$NL = 1BL * randP + 2BL * (1 - randP) \qquad (5)$$

This will leave the new lizard in any point between the ones described above, the vector of population changes increasing in one for the new lizard, and it will help to make more diverse the positions of every lizard avoiding the stagnation for having only one type of movement, Fig. 8 add an orange lizard between the blue ones.

3.4. Search strategies

Once obtained the correct amount of particles in the iteration, the process continues, by applying search strategies according to each of the morphs, inspired by each of the lizards behaviors; the blue lizard is monogamous and tries to stay near of more blue lizards to protect each other, orange lizards behave more aggressively and territorial, even more against each other, and yellow lizards do not care about territory at all, they are opportunistic and tries to steal females of other morphs, this is most effective

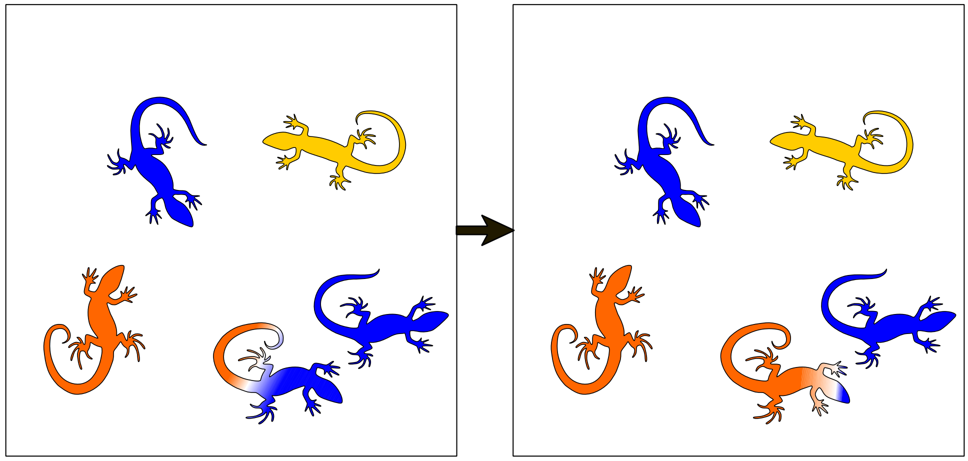

Fig. 7. Transform lizard function.

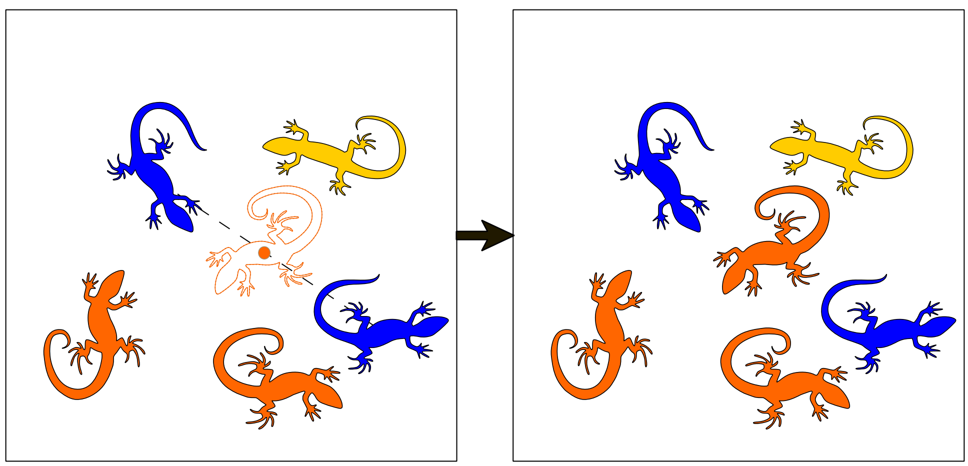

Fig. 8. Add lizard function.

on the orange lizards because they have large territories and it becomes difficult to protect it.

(A) Blue defensive strategy

Blue strategy revolves around getting near themselves, without sharing the same spot, for this we made a formula that helps reduce movement according to how close they are, the closer, the smaller the movement will be, we modeled according to the formula of the golden ratio ϕ [68] which has been used in several engineering and biological works [69–72].

$$\varphi = \frac{\left(1+\sqrt{5}\right)}{2} \tag{6}$$

Initially, we need to get the best lizard in the total population *BLP* and the best of the blue lizards *BLB*, both sorted according to their fitness, if it is the same lizard it will need to get the second-best blue lizard instead, to make an absolute difference between them and to get a delta distance Δ.

$$\Delta = absolute(BLP - BLB) \tag{7}$$

The next step is to move the lizards one by one, a new lower and upper bound will limit them, the formula to calculate them is below:

$$LB = BLP - \Delta \tag{8}$$

$$UB = BLP + \Delta \tag{9}$$

The algorithm randomly selects a dimension in which the new point will move between the upper and lower limit, for the other dimensions a random number that goes from 0 to 1 is used for each dimension, if the number is under 0.5 the position will be equal to the upper limit, elsewhere the position for that dimension will be the lower limit, the next blue lizard of the population will have the same rules of movement as the previous, but with a reduced delta distance calculated by dividing it by the golden ratio previously calculated, and will be used to calculate the new limits.

$$\Delta = \Delta/\varphi \tag{10}$$

This movement will lead to an exploitation of the search space in various levels, simulating an arrangement on the edges of a hypercube, the higher the number of blue lizards in the population in that specific iteration, the deeper the exploitation will get.

In Fig. 9 it can be seen at the left side that there are three lizards, the L^1 being the lizard with the better fitness, it select the nearest lizard and the distance between them is taken as Δ, after this the distance is added to the position of L^1 in order to obtain the new upper limit, and subtracted to obtain the lower limit, every other lizard will move towards the best one, the next blue lizard in the population will be positioned in the outer square, which represents the first hypercube (2 dimensional in this case), the third one will be in the second outer edges calculated by

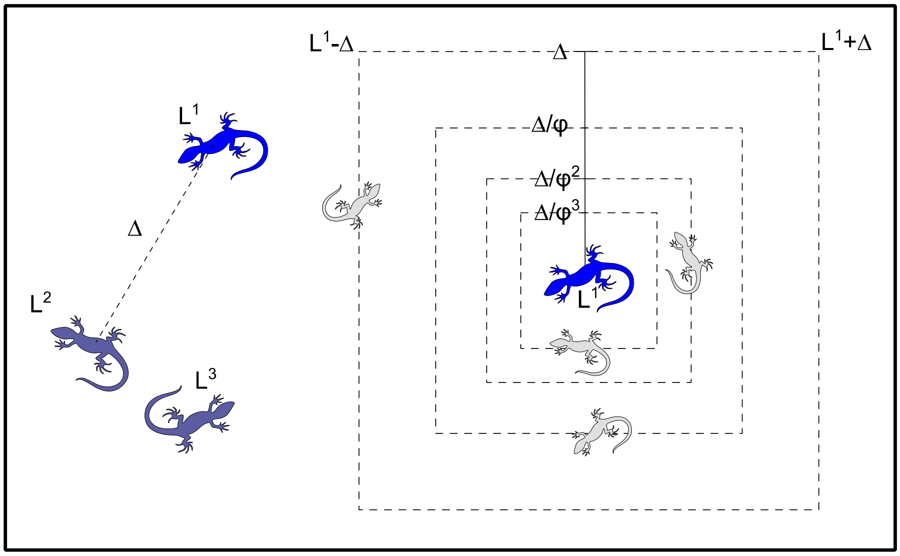

Fig. 9. Visual representation of the blue search strategy in 2 dimensions.

the previous hypercube divided by φ and so on, this generate a controlled exploitation that goes deeper each time.

(B) Orange dominant expansion strategy

Orange lizards have vast territories with many females, they are ultra-dominant, and other morphs cannot endure their attacks. Their ultra-dominant nature makes them move away from each other, thus expanding the orange territories, we take this into consideration and identify it as an explore movement.

The first step to make the search process is to obtain the best orange lizard according to fitness, and for the other orange lizards, we have to get the delta distance between each one and the best.

$$\Delta^n = BLO - O^n \tag{11}$$

Once acquiring the distance, a gain must be calculated by dividing the upper limit *UpLim* minus the lower limit *LowLim* over the number of orange lizards *#O*, the gain serves to move them according to the delta distance multiplied for a random between 0 and 1.

$$Gain = \frac{UpLim - LowLim}{\#O} \tag{12}$$

$$\mathrm{m}^n = 1/\Delta^n * rand * Gain \tag{13}$$

Once getting the desired movement m^n, the selected orange O^n can move away from the best orange.

$$O^n = BLO - \mathrm{m}^n \tag{14}$$

This new position leads to the exploration; it makes the particles to move further away from the nearest they are from the best individual in the orange population, if it is the same particle it will not move at all and will not access to the cost function.

(C) Yellow sneaky strategy

For the yellow search strategy, we focused on the sneaking side of the lizards, as they do not have territories; they enter to the orange territories to steal their females, said areas are large, making hard to defend them from yellow lizards.

This behavior works by choosing an orange lizard by the roulette method [73], this is to give them the chance to get onto any lizard territories with an advantage to those with the best fitness. Once the orange lizard is selected O^m, the yellow lizard Y^n will move towards it, the operator is:

$$Y^n = Y^n + rand * (O^m - Y^n) \tag{15}$$

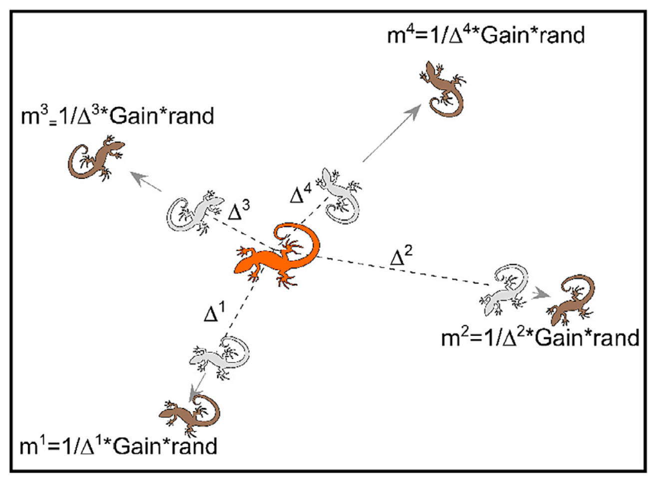

Fig. 10. Visual representation of the orange search strategy.

The roulette represents the opportunistic behavior by allowing to choose another lizard to invade in each iteration, this will occur for each yellow lizard in the population. This allows exploiting the explored areas previously found by orange lizards.

In Fig. 11 it can be seen that two yellow lizards are moving towards the orange lizard, L^7 selected the orange lizard L^6 which had only 11% of probability to be selected, and L^2 selected L^5 which is had the most probably to be selected with a 46%, it must be said that both lizards could have chosen L5, but the roulette method gives everyone a chance to be selected as the winner, the yellow lizards then proceeded to move towards their selected orange using Eq. (15).

3.5. End of the iterative process

After the movements are over, the population will have better positions and fitness, and then the iterative process is repeated to improve even more those values.

In one run of the algorithm, the subpopulations number of individuals will fluctuate at constant rates, from having the minimum number of particles to end up with over two-thirds of the maximum population and keep oscillating, but at any time

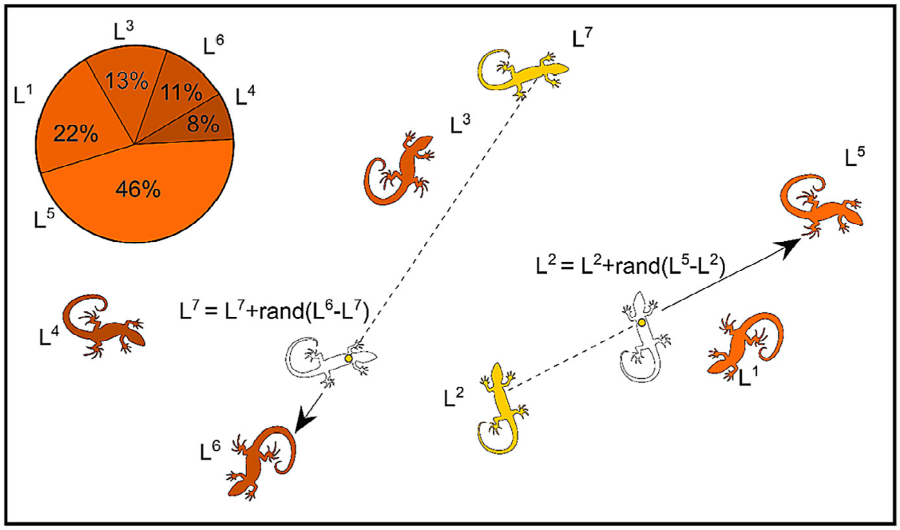

Fig. 11. Visual representation of the yellow search strategy.

of the iterative process the total population number of lizards will stay the same, allowing to continually change the balance of exploration and exploitation of the search space, searching for new solutions and refining the best ones from the beginning until the end of the iterative process.

4. Experimental results

The objective of this section is to evaluate the performance of the SBLA method through several optimization problems and to compare them with some similar optimization algorithms. These results are validated statically using the Wilcoxon rank sum [74], which compares two samples and measures their differences to determine whether two independent samples have the same distribution or if they differ from each other.

The average best solution (AB), the median best (MB), and the standard deviation (SD) are the indicators used on the experiment. Each of the functions is tested 30 times for each algorithm, and the best values of each test are used to obtain the AB, MB, and SD. The indicators AB and MB, evaluate how accurate the solutions were, and the SD evaluates the dispersion and reliability of the results.

The parameters used on SBLA are the common on the area, population *P* and maximum iterations *Iter*, population was treated as 50 for every experiment and it is significant to express that a minimum population of 30 is needed to show diversity from every lizard subpopulation, and the frequency of the season *S* is obtained automatically by the algorithm, it divides the maximum number of iterations over 12 to get them, this number was obtained via experimentation and it is automatically calculated by the algorithm.

The algorithms that were compared against this approach contain well-known heuristics and the majority are population-based algorithms; Differential evolution (DE), Simulated annealing (SA), Harmony search (HS), artificial bee colony (ABC), Particle search optimizer (PSO), Bat algorithm (BA), Firefly algorithm (FA), Moth flame optimization (MFO), Crow search algorithm (CSA) and Gray Wolf optimizer (GWO). These algorithms were compared alongside a benchmark of 5 unimodal, 18 multimodal and four composite functions, with the parameters listed in Appendix Table 1.

To ensure the repeatability of the test results, the parameters used in each method have been configured according to the reported values, below is the configuration of these settings, every algorithm was tested using 50 particles of population, and the stop criteria are 50,000 function access for 30D, 80,000 for 50D and 160,000 for 100D.

The experimental results are divided into four subsections. 1.-Unimodal, 2.-Multimodal, 3-Composite, and 4.-Real problems, the functions above have their details explained in Appendix Tables A.1–A.3.

4.1. Unimodal functions

The functions from $f1$ to $f5$ are unimodal functions and they are described in Appendix Table A.1, they have a single optimum, those function aid to estimate the exploit capacity of the algorithm. Experimental results are listed in Tables 2–4, displaying the performance of every algorithm for each function in 30, 50, and 100 dimensions. Indicators AB, MB and SD represent the average, medium and standard deviation of the best's individual through the 30 tests of each function. The best outcomes of each parameter are highlighted in boldface.

It is noticeable from Table 2 that SBLA gets the best AB, MD, and SD in every function, GWO also gets the best SD on f_5, however, the proposed method gets a better result in AB and MD, this proves that the exploitation on the algorithm is superior than its competitors.

To demonstrate the scalability of the proposed algorithm, the same functions were performed at 50 and 100 dimensions, Tables 3 and 4 shows their results. From Table 3 SBLA gets the best AB, MD, and SD in every function, it is matched by GWO in f_5, this is due to both algorithms finding the best solution in each of the 30 repetitions.

In Table 4 it can be observed how SBLA keeps the best AB, MD and SD even if the scalability of the problem grew, GWO gets the same values on f_5, as result of both obtaining the global best each time the experiment was executed, but the proposed algorithm gets better results from f_1 to f_4 demonstrating that it can solve unimodal functions better than its competitors.

Results from Tables 4–6 are validated statistically by the non-parametric Wilcoxon test. The test compares the difference between two methods and measures their differences. For the algorithms in comparison, a 5% (5.0E−2) of significance level is considered for the best results in the 30 separate runs of each algorithm. The proposed method is compared to the other algorithms individually. The configuration on the tables is set: SBLA vs. ABC, SBLA vs. BA, SBLA vs. DE, SBLA vs. FA, SBLA vs. HS, SBLA

Table 1
Parameter settings.

Setting configuration	
SBLA	Every parameter is obtained automatically depending on the population and iterations, it does not require parameter tuning.
[36] ABC	Onlooker 50%, employees 50%, acceleration coefficient upper bound = 1, abandonment limit L = round (0.6*dimensions*population)
[75] BA	The parameters where set as follows: Initial loudness rate $A = 2$, pulse emission rate $r = 0.9$, minimum frequency $f_{\min} = 0$ and maximum frequency $f_{\max} = 1$, respectively
[76] DE	The crossover rate is set to $CR = 0.5$, while the differential weight is given as $F = 0.2$
[77] FA	The randomness factor and the light absorption coefficient are set to $\alpha = 0.2$ and $\gamma = 1.0$, respectively
[64] HS	$HMCR = 0.7$ Harmony Memory Considering Rate finds notes randomly within the possible playable range. $PAR = 0.3$ Pitch Adjusting Rate.
[78] MFO	The number of flames is set as $N_{\text{flames}} = \text{round}\left((N_{\text{pop}} - k) * (N_{\text{pop}} - 1)/k_{\max}\right)$, where N_{pop} is the population, $k_{\max}$ the maximum number of iterations and k the actual iteration.
[78] SA	The algorithm's initial temperature is set to $T^0 = 1$, while the cooling schedule employed correspond to a geometrical cooling scheme considering a cooling rate of β calculated to obtain in the last iteration a temperature of 0.
[78] CSA	The awareness probability is set to $AP = 0.1$, while the flight length is given as $fl = 2$
[78] PSO	The cognitive and social coefficients are set to $c_1 = 2.0$ and $c_2 = 2.0$, respectively. Also, the inertia weight factor ω is set to decreases linearly from 0.9 to 0.2 as the search process evolves
[78] GWO	The algorithm's parameter a is set to decrease linearly from 2 to 0

Table 2
Results of unimodal functions 30D.

		SBLA	ABC	BA	DE	FA	HS	MFO	SA	CSA	PSO	GWO
$f_1(x)$	AB	**2.16E−141**	9.32E+04	2.44E+05	2.74E+04	1.60E+03	2.03E+05	1.40E+04	5.56E+05	4.94E+05	2.30E+04	3.36E−50
	MD	**1.46E−150**	9.31E+04	2.34E+05	2.70E+04	1.56E+03	2.02E+05	4.29E+03	5.41E+05	4.97E+05	1.50E+04	6.94E−51
	SD	**1.18E−140**	1.63E+04	1.27E+05	3.19E+03	3.54E+02	1.65E+04	1.99E+04	9.66E+04	5.23E+04	2.43E+04	9.42E−50
$f_2(x)$	AB	**3.69E−137**	4.97E+06	1.66E+07	8.45E+05	8.51E+04	1.04E+07	7.55E+05	2.71E+07	3.18E+07	2.43E+06	7.31E−49
	MD	**2.31E−148**	5.03E+06	1.60E+07	8.28E+05	8.39E+04	1.05E+07	2.50E+05	2.71E+07	3.25E+07	1.77E+06	1.53E−49
	SD	**2.02E−136**	8.83E+05	6.60E+06	1.21E+05	1.37E+04	1.23E+06	1.09E+06	4.47E+06	3.80E+06	2.12E+06	1.68E−48
$f_3(x)$	AB	**9.10E−149**	3.01E+01	4.41E−05	1.24E+01	4.65E−01	5.81E+01	2.62E+00	2.34E+02	1.24E+02	8.35E−06	1.94E−54
	MD	**1.42E−156**	2.90E+01	4.40E−05	1.26E+01	4.61E−01	5.92E+01	5.92E−08	2.36E+02	1.26E+02	2.66E−06	1.68E−54
	SD	**2.81E−148**	5.21E+00	5.12E−06	1.76E+00	8.95E−02	5.77E+00	8.00E+00	3.54E+01	1.31E+01	1.88E−05	1.77E−54
$f_4(x)$	AB	**5.01E−143**	2.08E+03	1.34E+02	6.56E+02	3.60E+01	4.51E+03	4.03E+02	1.37E+04	1.10E+04	5.27E+02	5.46E−52
	MD	**7.46E−150**	2.07E+03	1.77E+01	6.59E+02	3.46E+01	4.58E+03	1.00E+02	1.39E+04	1.11E+04	4.00E+02	9.55E−53
	SD	**2.67E−142**	3.01E+02	2.35E+02	8.81E+01	6.97E+00	4.38E+02	4.94E+02	2.21E+03	1.30E+03	4.30E+02	1.95E−51
$f_5(x)$	AB	**0.00E+00**	4.45E−01	1.12E−08	2.59E−03	3.56E−07	3.10E−04	2.25E−20	1.88E+00	2.29E−02	6.14E−15	2.16E−218
	MD	**0.00E+00**	4.44E−01	1.07E−08	2.31E−03	3.09E−07	2.55E−04	1.75E−22	1.84E+00	2.09E−02	3.03E−17	2.26E−225
	SD	**0.00E+00**	2.44E−01	4.90E−09	1.29E−03	2.28E−07	1.49E−04	7.58E−20	7.81E−01	1.32E−02	2.72E−14	**0.00E+00**

vs. MFO, SBLA vs. SA, SBLA vs. CSA, SBLA vs. PSO, and SBLA vs. GWO.

Tables 5 to 7 represent the visualization of p-values resulting from the Wilcoxon test, considering the Bonferroni correction [79], the tables represent the results on 50D, as they are very similar, any potential difference is treated in the description. In the Wilcoxon test, a null hypothesis is established in which the difference between two approaches is imperceptible, and if the hypothesis is rejected, it means that the variance among them is significant. The Bonferroni correction is used to avoid falling into a Type 1 Error because as the number of comparisons increases the possibility of this error also rises; the significance level is recalculated by dividing the 5.0E−2 of significance between the number of tests, which is 30 and this results on a value of 1.7E−3.

The p-values from the Wilcoxon test are obtained by comparing the best value of each run between the proposed method against every other algorithm; the null hypothesis is rejected for all values lower than 1.7E−3, which means that the methods compared are significantly different from each other. A simplified form to identify the outcomes between the algorithms is used, the symbols ▲, ▼, and ► represent the performance of the SBLA algorithm against the compared technique, the symbol ▲ means that SBLA had a better performance, ▼ shows that SBLA performed worse, and ► specifies that the analysis did not identify any difference between SBLA and its competitors.

Table 5 shows the Wilcoxon table for 50D, in which the proposed method gets the best results in the five unimodal functions, and it gets a lower value than 1.7E−3, this means that the proposed method is significantly different from the others, but in f_5 when growing the dimensional value, Wilcoxon found similarities in 50D and it did not find any difference in 100D obtaining a NaN "Not a number" value, however, this was generated since both methods obtained the best solution for f_5 in the 30 independent runs, showing that both competitors are fully competent.

4.2. Multimodal functions

Appendix Table A.2 presents the multimodal functions comprised from f_6 to f_{23} and are characterized by having multiple local optima that increase exponentially as the number of dimensions gets bigger. Therefore, this kind of functions represent the most difficult task of optimization; these functions challenges the algorithms to demonstrate their ability to find the global optimum among all the local optimums. The experimental results are shown in Tables 6–8 considering the performance of each algorithm for each test function in 30, 50, and 100 dimensions.

In Table 6 it can be observed that SBLA gets the best AB, MD, and SD in most functions, in f_7, f_{10}, f_{21} and f_{22} GWO gets the best values, but SBLA gets the second best solutions, in f_{13} GWO gets the best solution, and SBLA is tied against FA in the second place,

Table 3
Results of unimodal functions 50D.

		SBLA	ABC	BA	DE	FA	HS	MFO	SA	CSA	PSO	GWO
$f_1(x)$	AB	**2.72E−223**	2.03E+04	1.44E+05	4.58E+03	2.15E+02	1.76E+05	9.15E+04	1.67E+06	4.85E+05	2.73E+04	2.77E−84
	MD	**2.82E−240**	1.98E+04	1.14E+05	4.48E+03	2.12E+02	1.78E+05	7.09E+04	1.66E+06	4.83E+05	1.93E+04	4.30E−85
	SD	**0.00E+00**	3.99E+03	1.36E+05	6.55E+02	2.61E+01	1.49E+04	7.34E+04	2.43E+05	5.43E+04	2.27E+04	5.16E−84
$f_2(x)$	AB	**3.45E−223**	1.08E+06	1.66E+07	1.25E+05	1.53E+04	9.06E+06	6.98E+06	1.21E+08	3.17E+07	2.46E+06	2.63E−82
	MD	**1.45E−240**	9.93E+05	1.58E+07	1.24E+05	1.45E+04	9.29E+06	5.24E+06	1.18E+08	3.28E+07	2.10E+06	3.55E−83
	SD	**0.00E+00**	2.37E+05	6.92E+06	1.45E+04	3.10E+03	1.31E+06	6.24E+06	2.10E+07	4.20E+06	1.73E+06	6.45E−82
$f_3(x)$	AB	**1.41E−232**	6.49E+00	3.93E−05	2.05E+00	5.81E−02	5.23E+01	1.22E+01	4.18E+02	1.28E+02	7.81E−10	6.41E−88
	MD	**2.96E−240**	6.55E+00	4.04E−05	2.02E+00	5.89E−02	5.38E+01	3.79E−05	4.25E+02	1.28E+02	3.26E−10	2.02E−88
	SD	**0.00E+00**	1.17E+00	3.52E−06	3.32E−01	8.03E−03	5.85E+00	1.91E+01	5.13E+01	1.24E+01	1.60E−09	1.01E−87
$f_4(x)$	AB	**1.75E−227**	4.58E+02	3.73E+00	1.10E+02	5.24E+00	3.98E+03	2.64E+03	3.83E+04	1.10E+04	4.13E+02	6.80E−86
	MD	**1.19E−237**	4.44E+02	4.65E−03	1.14E+02	5.30E+00	3.99E+03	2.55E+03	3.79E+04	1.11E+04	3.00E+02	9.37E−87
	SD	**0.00E+00**	9.24E+01	1.54E+01	1.53E+01	7.12E−01	4.15E+02	1.89E+03	5.64E+03	1.05E+03	4.24E+02	1.26E−85
$f_5(x)$	AB	**0.00E+00**	3.38E−01	5.22E−09	7.38E−05	5.14E−08	1.39E−04	1.16E−15	2.40E+00	2.16E−02	4.05E−21	**0.00E+00**
	MD	**0.00E+00**	3.56E−01	4.95E−09	4.48E−05	3.93E−08	1.22E−04	3.04E−18	2.15E+00	1.92E−02	2.49E−22	**0.00E+00**
	SD	**0.00E+00**	1.86E−01	1.80E−09	6.32E−05	4.37E−08	6.72E−05	6.04E−15	9.65E−01	1.47E−02	1.09E−20	**0.00E+00**

		SBLA	ABC	BA	DE	FA	HS	MFO	SA	CSA	PSO	GWO
$f_1(x)$	AB	**2.72E−223**	2.03E+04	1.44E+05	4.58E+03	2.15E+02	1.76E+05	9.15E+04	1.67E+06	4.85E+05	2.73E+04	2.77E−84
	MD	**2.82E−240**	1.98E+04	1.14E+05	4.48E+03	2.12E+02	1.78E+05	7.09E+04	1.66E+06	4.83E+05	1.93E+04	4.30E−85
	SD	**0.00E+00**	3.99E+03	1.36E+05	6.55E+02	2.61E+01	1.49E+04	7.34E+04	2.43E+05	5.43E+04	2.27E+04	5.16E−84
$f_2(x)$	AB	**3.45E−223**	1.08E+06	1.66E+07	1.25E+05	1.53E+04	9.06E+06	6.98E+06	1.21E+08	3.17E+07	2.46E+06	2.63E−82
	MD	**1.45E−240**	9.93E+05	1.58E+07	1.24E+05	1.45E+04	9.29E+06	5.24E+06	1.18E+08	3.28E+07	2.10E+06	3.55E−83
	SD	**0.00E+00**	2.37E+05	6.92E+06	1.45E+04	3.10E+03	1.31E+06	6.24E+06	2.10E+07	4.20E+06	1.73E+06	6.45E−82
$f_3(x)$	AB	**1.41E−232**	6.49E+00	3.93E−05	2.05E+00	5.81E−02	5.23E+01	1.22E+01	4.18E+02	1.28E+02	7.81E−10	6.41E−88
	MD	**2.96E−240**	6.55E+00	4.04E−05	2.02E+00	5.89E−02	5.38E+01	3.79E−05	4.25E+02	1.28E+02	3.26E−10	2.02E−88
	SD	**0.00E+00**	1.17E+00	3.52E−06	3.32E−01	8.03E−03	5.85E+00	1.91E+01	5.13E+01	1.24E+01	1.60E−09	1.01E−87
$f_4(x)$	AB	**1.75E−227**	4.58E+02	3.73E+00	1.10E+02	5.24E+00	3.98E+03	2.64E+03	3.83E+04	1.10E+04	4.13E+02	6.80E−86
	MD	**1.19E−237**	4.44E+02	4.65E−03	1.14E+02	5.30E+00	3.99E+03	2.55E+03	3.79E+04	1.11E+04	3.00E+02	9.37E−87
	SD	**0.00E+00**	9.24E+01	1.54E+01	1.53E+01	7.12E−01	4.15E+02	1.89E+03	5.64E+03	1.05E+03	4.24E+02	1.26E−85
$f_5(x)$	AB	**0.00E+00**	3.38E−01	5.22E−09	7.38E−05	5.14E−08	1.39E−04	1.16E−15	2.40E+00	2.16E−02	4.05E−21	**0.00E+00**
	MD	**0.00E+00**	3.56E−01	4.95E−09	4.48E−05	3.93E−08	1.22E−04	3.04E−18	2.15E+00	1.92E−02	2.49E−22	**0.00E+00**
	SD	**0.00E+00**	1.86E−01	1.80E−09	6.32E−05	4.37E−08	6.72E−05	6.04E−15	9.65E−01	1.47E−02	1.09E−20	**0.00E+00**

Table 4
Results of unimodal functions 100D.

		SBLA	ABC	BA	DE	FA	HS	MFO	SA	CSA	PSO	GWO
$f_1(x)$	AB	**0.00E+00**	3.42E+06	4.50E+05	6.31E+05	8.27E+02	1.52E+06	7.11E+05	6.92E+06	2.36E+06	2.07E+05	4.94E−118
	MD	**0.00E+00**	3.44E+06	4.48E+05	6.40E+05	7.73E+02	1.52E+06	6.83E+05	6.86E+06	2.39E+06	1.87E+05	8.49E−119
	SD	**0.00E+00**	3.56E+05	3.55E+05	3.90E+04	1.95E+02	6.54E+04	3.86E+05	6.71E+05	1.63E+05	1.25E+05	8.08E−118
$f_2(x)$	AB	**0.00E+00**	3.65E+08	1.01E+08	3.79E+07	1.13E+05	1.79E+08	9.39E+07	1.08E+09	3.41E+08	3.66E+07	2.12E−116
	MD	**0.00E+00**	3.68E+08	1.00E+08	3.83E+07	1.10E+05	1.84E+08	8.20E+07	1.10E+09	3.43E+08	2.99E+07	4.60E−117
	SD	**0.00E+00**	3.50E+07	7.87E+07	2.94E+06	2.49E+04	1.38E+07	5.39E+07	1.42E+08	2.45E+07	2.00E+07	4.08E−116
$f_3(x)$	AB	**0.00E+00**	5.73E+02	1.57E−04	1.49E+02	1.10E−01	2.17E+02	5.41E+01	8.33E+02	2.97E+02	9.61E+00	1.55E−122
	MD	**0.00E+00**	5.78E+02	1.55E−04	1.50E+02	1.01E−01	2.21E+02	5.25E+01	8.19E+02	2.96E+02	2.36E−06	4.74E−123
	SD	**0.00E+00**	4.72E+01	1.25E−05	8.23E+00	2.84E−02	1.29E+01	3.12E+01	9.95E+01	1.94E+01	1.61E+01	2.63E−122
$f_4(x)$	AB	**0.00E+00**	8.17E+04	6.71E−02	1.46E+04	1.88E+01	3.53E+04	1.17E+04	1.58E+05	5.53E+04	4.67E+03	7.75E−120
	MD	**0.00E+00**	8.19E+04	6.28E−02	1.46E+04	1.93E+01	3.51E+04	1.08E+04	1.60E+05	5.57E+04	4.55E+03	1.95E−120
	SD	**0.00E+00**	8.14E+03	3.27E−02	1.07E+03	4.27E+00	2.08E+03	6.31E+03	1.65E+04	3.35E+03	2.46E+03	1.67E−119
$f_5(x)$	AB	**0.00E+00**	1.31E+00	1.77E−09	5.91E−01	4.56E−08	5.97E−04	2.62E−07	3.25E+00	2.72E−02	1.11E−14	**0.00E+00**
	MD	**0.00E+00**	1.32E+00	1.70E−09	6.00E−01	3.67E−08	6.24E−04	2.87E−14	3.34E+00	2.67E−02	1.02E−16	**0.00E+00**
	SD	**0.00E+00**	3.11E−01	7.20E−10	1.98E−01	3.32E−08	2.25E−04	1.44E−06	9.44E−01	1.29E−02	4.65E−14	**0.00E+00**

Table 5
Wilcoxon of unimodal functions 50D.

	SBLA VS ABC	SBLA VS BA	SBLA VS DE	SBLA VS FA	SBLA VS HS	SBLA VS MFO	SBLA VS SA	SBLA VS CSA	SBLA VS PSO	SBLA VS GWO
$f_1(x)$	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲
$f_2(x)$	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲
$f_3(x)$	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲
$f_4(x)$	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲
$f_5(x)$	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	2.73E−03►

however, the proposed method gets a better SD, making it more reliable, and in f_{15} SBLA gets the third place after MFO and PSO, the overall results shows that SBLA is more than capable in the task of solving multimodal functions.

Tables 7 and 8 shows the results from multimodal functions in 50 and 100 dimensions, the complexity of these functions increases exponentially as the dimensionality rises, this is caused by the local optimums that exist in each dimension, if the algorithm get stuck into any of them, it will be less likely to obtain the global solution.

Table 6
Results of multimodal functions 30D.

		SBLA	ABC	BA	DE	FA	HS	MFO	SA	CSA	PSO	GWO
$f_6(x)$	AB	**0.00E+00**	1.68E+01	1.54E+01	1.42E+01	3.96E+00	1.61E+01	1.80E+01	9.71E−01	1.85E+01	9.86E−01	2.58E−14
	MD	**0.00E+00**	1.67E+01	1.55E+01	1.39E+01	3.96E+00	1.61E+01	1.87E+01	0.00E+00	1.84E+01	1.04E+00	2.84E−14
	SD	**0.00E+00**	8.31E−01	9.28E−01	1.11E+00	2.14E−01	3.25E−01	3.11E+00	2.74E+00	2.04E−01	7.40E−01	3.84E−15
$f_7(x)$	AB	8.94E−01	1.43E+06	6.54E+02	6.52E+04	4.00E+02	3.49E+05	1.84E+05	3.13E+06	1.02E+06	1.03E+04	**6.67E−01**
	MD	9.12E−01	1.43E+06	3.18E+01	6.19E+04	3.87E+02	3.51E+05	8.05E+04	3.00E+06	1.01E+06	4.87E+01	**6.67E−01**
	SD	9.86E−02	4.27E+05	1.85E+03	1.68E+04	1.60E+02	5.68E+04	3.10E+05	9.76E+05	1.49E+05	3.18E+04	**2.71E−06**
$f_8(x)$	AB	**0.00E+00**	1.07E+02	2.55E+02	4.24E+01	2.65E+00	2.02E+02	7.28E+01	6.24E+02	4.37E+02	1.74E−02	2.10E−03
	MD	0.00E+00	1.09E+02	2.40E+02	4.33E+01	2.62E+00	2.06E+02	4.57E+01	5.98E+02	4.39E+02	1.60E−02	**0.00E+00**
	SD	0.00E+00	1.71E+01	6.13E+01	5.48E+00	3.26E−01	1.72E+01	9.28E+01	8.96E+01	3.11E+01	1.36E−02	5.10E−03
$f_9(x)$	AB	0.00E+00	2.75E+00	3.55E−06	7.76E−02	6.07E−05	2.21E−01	7.72E−02	5.13E+00	6.58E−01	5.24E−08	1.56E−110
	MD	0.00E+00	2.72E+00	3.62E−06	7.92E−02	4.10E−05	2.22E−01	7.63E−07	4.63E+00	6.23E−01	1.39E−08	2.03E−115
	SD	0.00E+00	7.95E−01	1.54E−06	1.83E−02	4.86E−05	4.48E−02	2.94E−01	2.19E+00	1.55E−01	1.07E−07	7.85E−110
$f_{10}(x)$	AB	2.62E+00	2.02E+02	2.38E+01	6.36E+01	6.45E+00	6.23E+01	5.88E+01	3.08E+02	1.54E+02	4.53E+00	**2.34E+00**
	MD	2.65E+00	2.02E+02	2.43E+01	6.38E+01	5.86E+00	6.28E+01	5.66E+01	3.16E+02	1.49E+02	3.32E+00	**2.36E+00**
	SD	**2.43E−01**	1.96E+01	7.65E+00	8.50E+00	2.85E+00	8.13E+00	1.73E+01	6.97E+01	1.72E+01	4.30E+00	2.44E−01
$f_{11}(x)$	AB	**0.00E+00**	4.33E−01	1.59E−04	**0.00E+00**	1.64E−05	1.34E−02	5.37E−06	2.20E−08	2.25E−01	**0.00E+00**	1.10E−10
	MD	**0.00E+00**	4.41E−01	2.04E−08	**0.00E+00**	1.64E−05	1.33E−02	**0.00E+00**	0.00E+00	2.28E−01	**0.00E+00**	1.49E−11
	SD	**0.00E+00**	7.72E−02	3.81E−04	**0.00E+00**	8.94E−06	3.15E−03	2.52E−05	9.81E−08	4.77E−02	**0.00E+00**	2.34E−10
$f_{12}(x)$	AB	**0.00E+00**	6.36E+10	2.15E+08	3.98E−29	1.09E−37	5.97E−35	1.03E−213	6.18E−23	6.75E+07	**0.00E+00**	**0.00E+00**
	MD	**0.00E+00**	2.63E+08	1.73E+03	1.28E−33	8.58E−47	6.11E−40	2.81E−228	7.71E−293	1.51E+06	**0.00E+00**	**0.00E+00**
	SD	**0.00E+00**	1.72E+11	9.77E+08	2.17E−28	5.98E−37	3.26E−34	**0.00E+00**	3.33E−22	3.20E+08	**0.00E+00**	**0.00E+00**
$f_{13}(x)$	AB	3.01E+01	4.23E+01	7.65E+01	3.81E+01	3.01E+01	5.59E+01	3.30E+01	9.56E+01	9.02E+01	3.15E+01	**3.00E+01**
	MD	**3.00E+01**	4.25E+01	7.50E+01	3.80E+01	**3.00E+01**	5.60E+01	**3.00E+01**	9.50E+01	9.00E+01	**3.00E+01**	**3.00E+01**
	SD	3.05E−01	1.80E+00	8.63E+00	1.22E+00	3.46E−01	2.31E+00	4.68E+00	7.68E+00	3.38E+00	3.26E+00	**0.00E+00**
$f_{14}(x)$	AB	**1.02E−110**	1.02E+04	1.63E+00	7.04E+03	6.45E+01	3.75E+03	2.81E+02	4.26E+03	7.18E+03	6.40E+02	5.42E−06
	MD	**7.34E−126**	1.05E+04	4.74E−02	6.82E+03	6.68E+01	3.70E+03	1.81E+02	3.02E+03	7.41E+03	4.38E+02	3.48E−06
	SD	**5.56E−110**	2.15E+03	6.30E+00	9.86E+02	2.88E+01	6.29E+02	3.79E+02	4.49E+03	1.28E+03	8.71E+02	7.04E−06
$f_{15}(x)$	AB	4.86E+03	1.31E+11	3.44E+10	6.61E+09	2.14E+07	3.00E+10	**6.25E−01**	3.28E+11	7.53E+10	8.09E+02	8.46E+03
	MD	4.81E+03	1.27E+11	3.13E+10	6.61E+09	1.91E+07	2.97E+10	**4.12E−01**	3.22E+11	7.34E+10	3.40E+02	8.80E+03
	SD	1.09E+03	3.58E+10	2.29E+10	1.01E+09	1.24E+07	4.62E+09	**6.34E−01**	9.64E+10	1.24E+10	1.02E+03	2.13E+03
$f_{16}(x)$	AB	**8.55E+00**	1.13E+02	5.12E+01	3.04E+01	2.04E+01	4.91E+01	1.45E+01	2.19E+02	9.12E+01	2.20E+01	1.73E+01
	MD	**8.65E+00**	1.10E+02	4.98E+01	3.02E+01	2.04E+01	4.93E+01	1.15E+01	2.07E+02	9.26E+01	2.17E+01	1.74E+01
	SD	**4.33E−01**	2.16E+01	1.65E+01	1.63E+00	9.76E−01	3.74E+00	6.45E+00	5.78E+01	1.31E+01	1.83E+00	5.29E−01
$f_{17}(x)$	AB	**0.00E+00**	5.57E+02	1.29E+02	2.79E+02	1.56E+02	2.84E+02	1.49E+02	3.98E+02	5.05E+02	1.80E+02	6.88E−02
	MD	**0.00E+00**	5.59E+02	1.35E+02	2.79E+02	1.51E+02	2.86E+02	1.57E+02	4.07E+02	5.12E+02	1.76E+02	**0.00E+00**
	SD	**0.00E+00**	2.39E+01	4.09E+01	1.83E+01	2.70E+01	1.59E+01	3.61E+01	5.79E+01	2.59E+01	3.84E+01	3.77E−01
$f_{18}(x)$	AB	**2.89E+01**	5.52E+05	2.84E+02	2.38E+04	7.38E+02	1.75E+05	8.88E+04	6.00E+05	9.85E+05	1.96E+05	4.66E+01
	MD	**2.89E+01**	5.63E+05	1.06E+02	2.29E+04	6.70E+02	1.84E+05	5.65E+04	6.06E+05	9.78E+05	1.73E+05	4.62E+01
	SD	**3.82E−02**	1.77E+05	5.78E+02	4.37E+03	2.81E+02	2.57E+04	7.77E+04	2.25E+05	2.26E+05	9.80E+04	7.17E−01
$f_{19}(x)$	AB	**3.76E−65**	8.79E+01	5.04E+01	8.01E+01	3.83E+01	6.00E+01	5.51E+01	4.17E+01	6.50E+01	3.12E+01	2.67E−11
	MD	**6.21E−71**	8.77E+01	5.16E+01	8.06E+01	3.85E+01	6.05E+01	5.58E+01	4.49E+01	6.54E+01	3.14E+01	1.91E−11
	SD	**1.73E−64**	3.36E+00	7.18E+00	2.98E+00	6.70E+00	2.15E+00	9.79E+00	1.59E+01	3.05E+00	3.52E+00	2.74E−11
$f_{20}(x)$	AB	**3.38E−75**	1.90E+59	1.07E+56	6.38E+19	4.44E+05	9.43E+10	3.83E+02	1.35E+03	1.89E+57	7.38E+02	2.54E−29
	MD	**1.20E−78**	5.35E+56	3.00E+50	1.70E+15	1.09E+03	1.33E+07	4.00E+02	1.36E+03	4.81E+54	7.07E+02	1.67E−29
	SD	**1.08E−74**	6.85E+59	4.99E+56	2.96E+20	2.08E+06	4.89E+11	2.07E+02	1.67E+02	9.59E+57	2.14E+02	2.75E−29
$f_{21}(x)$	AB	5.00E−01	1.13E+04	3.37E+04	4.61E+03	1.92E+02	2.17E+04	2.67E+03	9.30E+04	4.73E+04	3.17E+00	**0.00E+00**
	MD	0.00E+00	1.12E+04	3.40E+04	4.65E+03	1.94E+02	2.22E+04	**0.00E+00**	9.30E+04	4.79E+04	3.00E+00	**0.00E+00**
	SD	5.72E−01	1.72E+03	8.42E+03	4.12E+02	3.93E+01	2.31E+03	4.50E+03	1.60E+04	4.17E+03	1.74E+00	**0.00E+00**
$f_{22}(x)$	AB	1.65E+01	2.17E+07	1.93E+07	2.37E+07	7.00E+05	1.59E+07	1.48E+05	2.52E+05	2.39E+07	9.63E+05	**−2.39E+02**
	MD	1.69E+01	2.16E+07	1.70E+07	2.39E+07	6.52E+05	1.57E+07	1.11E+05	4.28E+04	2.43E+07	6.20E+05	**−2.07E+02**
	SD	**5.28E+00**	4.12E+06	6.71E+06	2.42E+06	2.96E+05	1.86E+06	1.48E+05	4.61E+05	3.66E+06	1.04E+06	1.46E+02
$f_{23}(x)$	AB	**6.83E−118**	5.03E+03	2.35E+03	9.70E+02	2.70E+02	5.14E+02	3.04E+02	4.23E+02	1.08E+04	9.70E+02	2.17E−13
	MD	**1.17E−131**	4.83E+03	1.90E+03	9.61E+02	2.69E+02	5.22E+02	3.14E+02	4.39E+02	2.91E+03	9.28E+02	4.31E−14
	SD	**3.14E−117**	2.13E+03	1.74E+03	6.47E+01	3.65E+01	5.55E+01	9.22E+01	1.13E+02	4.45E+04	2.73E+02	7.85E−13

From Table 7 SBLA gets the best results in the majority of the multimodal functions, best values on f_7, f_{10}, f_{13}, f_{18} and f_{21} are obtained by GWO, f_{18} by DE and GWO and f_{15} by PSO. However, in f_7, f_{10}, and f_{18} SBLA gets the second-best result, also for f_{21} it gets the third best result, obtaining an overall best solution on multimodal functions at 50 dimensions.

In Table 8 there are some changes on the functions, it gets the best result in 11 out of 18 functions, in $f_{12}, f_{17}, f_{18}, f_{20}$ to f_{24} GWO gets the best result, sharing f_{17} first place with SBLA. In f_{21} and f_{22}, the proposed algorithm gets the second best solution, being suitable for every multimodal function in 100-dimensional space and shows the capacity of exploration that the algorithm has.

The results are validated by the Wilcoxon test, which compares the proposed method with every other algorithm. The significance level considered is $1.7E{-}3$, this includes the Bonferroni correction. The configuration on the tables is set: SBLA vs. ABC, SBLA vs. BA, SBLA vs. DE, SBLA vs. FA, SBLA vs. HS, SBLA vs. MFO, SBLA vs. SA, SBLA vs. CSA, SBLA vs. PSO, and SBLA vs. GWO. These results are visualized in Table 9.

The symbols ▲, ▼, and ► represent the performance of SBLA algorithm against the other algorithms, symbols are used to represent the performance; ▲ means that SBLA had a better performance, ▼ shows a worse one and ► specifies that the analysis did not identify any difference between the methods.

Table 7
Results of multimodal functions 50D.

		SBLA	ABC	BA	DE	FA	HS	MFO	SA	CSA	PSO	GWO
$f_6(x)$	AB	**4.74E−16**	1.19E+01	1.57E+01	1.03E+01	1.92E+00	1.57E+01	1.70E+01	2.98E−01	1.83E+01	1.57E−01	1.47E−14
	MD	**0.00E+00**	1.19E+01	1.60E+01	8.28E+00	1.94E+00	1.57E+01	1.85E+01	**0.00E+00**	1.83E+01	6.62E−04	1.42E−14
	SD	**1.23E−15**	1.28E+00	1.19E+00	3.48E+00	1.19E−01	3.46E−01	4.34E+00	1.00E+00	2.46E−01	4.36E−01	2.59E−15
$f_7(x)$	AB	9.64E−01	5.30E+05	2.29E+00	8.06E+03	2.35E+01	2.93E+05	1.97E+05	8.93E+06	1.06E+06	1.12E+02	**6.67E−01**
	MD	9.92E−01	4.83E+05	6.77E−01	8.30E+03	2.17E+01	2.95E+05	9.43E+04	8.86E+06	1.08E+06	8.27E+01	**6.67E−01**
	SD	6.16E−02	1.88E+05	7.16E+00	1.68E+03	7.31E+00	4.78E+04	2.36E+05	1.94E+06	1.88E+05	1.24E+02	**7.83E−06**
$f_8(x)$	AB	**0.00E+00**	2.20E+01	2.74E+02	7.88E+00	1.20E+00	1.81E+02	5.41E+01	1.17E+03	4.42E+02	6.49E−03	5.16E−04
	MD	**0.00E+00**	2.19E+01	2.86E+02	7.89E+00	1.20E+00	1.82E+02	3.59E−02	1.20E+03	4.40E+02	3.70E−03	**0.00E+00**
	SD	**0.00E+00**	3.85E+00	8.49E+01	1.07E+00	3.24E−02	1.64E+01	9.65E+01	1.44E+02	2.67E+01	7.96E−03	2.82E−03
$f_9(x)$	AB	**0.00E+00**	1.80E+00	3.71E−06	7.71E−03	1.73E−07	1.65E−01	1.93E−01	8.70E+00	7.08E−01	2.92E−09	2.02E−178
	MD	**0.00E+00**	1.75E+00	3.51E−06	7.78E−03	1.05E−07	1.65E−01	1.61E−07	8.68E+00	6.79E−01	1.34E−11	1.46E−182
	SD	**0.00E+00**	4.79E−01	1.43E−06	2.70E−03	1.56E−07	3.49E−02	4.39E−01	3.09E+00	2.08E−01	9.04E−09	**0.00E+00**
$f_{10}(x)$	AB	4.52E+00	1.55E+02	2.17E+01	3.15E+01	5.48E+00	5.56E+01	6.42E+01	5.63E+02	1.53E+02	5.77E+00	**2.23E+00**
	MD	4.55E+00	1.55E+02	2.04E+01	3.15E+01	3.57E+00	5.63E+01	5.91E+01	5.53E+02	1.53E+02	3.14E+00	**2.19E+00**
	SD	2.67E−01	2.19E+01	7.40E+00	5.82E+00	4.10E+00	5.67E+00	2.66E+01	8.43E+01	1.62E+01	6.10E+00	**2.40E−01**
$f_{11}(x)$	AB	**0.00E+00**	4.01E−01	8.24E−15	**0.00E+00**	1.24E−07	1.05E−02	2.37E−06	**0.00E+00**	2.14E−01	**0.00E+00**	1.64E−11
	MD	**0.00E+00**	4.04E−01	6.20E−15	**0.00E+00**	1.20E−07	9.93E−03	**0.00E+00**	**0.00E+00**	2.19E−01	**0.00E+00**	2.75E−12
	SD	**0.00E+00**	9.25E−02	5.25E−15	**0.00E+00**	5.15E−08	2.07E−03	1.30E−05	**0.00E+00**	3.82E−02	**0.00E+00**	4.57E−11
$f_{12}(x)$	AB	**0.00E+00**	5.33E+08	3.13E+08	3.06E−49	6.08E−61	8.02E−43	**0.00E+00**	6.08E−27	4.01E+06	**0.00E+00**	**0.00E+00**
	MD	**0.00E+00**	2.48E+05	9.32E+02	9.39E−53	1.01E−64	3.37E−47	**0.00E+00**	1.34E−280	5.16E+05	**0.00E+00**	**0.00E+00**
	SD	**0.00E+00**	2.49E+09	1.27E+09	1.52E−48	3.09E−60	3.24E−42	**0.00E+00**	3.33E−26	8.52E+06	**0.00E+00**	**0.00E+00**
$f_{13}(x)$	AB	3.80E+01	3.26E+01	7.76E+01	**3.00E+01**	3.02E+01	5.24E+01	3.98E+01	1.41E+02	8.97E+01	3.10E+01	**3.00E+01**
	MD	3.80E+01	3.20E+01	7.60E+01	**3.00E+01**	**3.00E+01**	5.30E+01	4.00E+01	1.42E+02	8.95E+01	**3.00E+01**	**3.00E+01**
	SD	2.08E+00	1.07E+00	9.21E+00	**0.00E+00**	5.04E−01	1.87E+00	7.75E+00	9.95E+00	3.09E+00	2.37E+00	**0.00E+00**
$f_{14}(x)$	AB	**2.56E−117**	8.25E+03	1.75E−02	4.75E+03	8.98E+00	3.29E+03	2.45E+03	6.48E+03	7.43E+03	6.11E+02	1.61E−06
	MD	**9.47E−186**	7.96E+03	1.77E−02	4.85E+03	8.68E+00	3.25E+03	2.15E+03	4.50E+03	7.46E+03	4.85E+02	7.20E−07
	SD	**1.40E−116**	1.95E+03	3.83E−03	8.78E+02	4.97E+00	5.18E+02	2.06E+03	5.36E+03	1.38E+03	4.92E+02	2.64E−06
$f_{15}(x)$	AB	2.65E+04	7.46E+10	2.66E+10	8.61E+08	1.86E+05	2.47E+10	1.04E+10	5.75E+11	7.24E+10	**1.33E−01**	8.00E+03
	MD	2.61E+04	7.00E+10	2.18E+10	8.92E+08	1.29E+05	2.52E+10	1.25E+02	5.41E+11	7.23E+10	**4.48E−02**	7.44E+03
	SD	3.49E+03	2.70E+10	1.57E+10	1.52E+08	1.21E+05	2.78E+09	2.37E+10	1.14E+11	1.23E+10	**2.27E−01**	2.28E+03
$f_{16}(x)$	AB	**1.66E+01**	7.04E+01	5.26E+01	2.59E+01	1.93E+01	4.36E+01	3.32E+01	5.86E+02	9.28E+01	2.19E+01	1.68E+01
	MD	**1.67E+01**	6.96E+01	4.74E+01	2.59E+01	1.93E+01	4.42E+01	2.96E+01	5.80E+02	9.38E+01	2.14E+01	1.69E+01
	SD	6.84E−01	1.41E+01	2.32E+01	9.55E−01	7.40E−01	3.41E+00	1.56E+01	1.20E+02	1.07E+01	1.84E+00	**5.05E−01**
$f_{17}(x)$	AB	**0.00E+00**	5.21E+02	1.43E+02	2.16E+02	1.26E+02	2.58E+02	2.80E+02	7.10E+02	5.13E+02	1.57E+02	**0.00E+00**
	MD	**0.00E+00**	5.20E+02	1.35E+02	2.20E+02	1.18E+02	2.60E+02	2.77E+02	7.05E+02	5.13E+02	1.50E+02	**0.00E+00**
	SD	**0.00E+00**	2.09E+01	4.43E+01	1.20E+01	3.65E+01	1.54E+01	4.42E+01	8.51E+01	2.00E+01	3.58E+01	**0.00E+00**
$f_{18}(x)$	AB	4.89E+01	2.50E+05	8.71E+01	6.24E+03	2.25E+02	1.35E+05	2.45E+05	7.82E+05	9.21E+05	1.88E+05	**4.66E+01**
	MD	4.89E+01	2.55E+05	6.84E+01	6.24E+03	2.13E+02	1.39E+05	2.23E+05	7.89E+05	9.39E+05	1.68E+05	**4.63E+01**
	SD	**3.85E−02**	7.27E+04	5.69E+01	1.18E+03	9.16E+01	2.49E+04	1.70E+05	3.66E+05	1.88E+05	1.00E+05	8.53E−01
$f_{19}(x)$	AB	**2.39E−86**	8.50E+01	4.58E+01	7.44E+01	2.91E+01	5.74E+01	7.70E+01	3.70E+01	6.53E+01	2.56E+01	1.05E−18
	MD	**2.23E−95**	8.61E+01	4.53E+01	7.44E+01	2.92E+01	5.74E+01	7.95E+01	3.89E+01	6.67E+01	2.64E+01	5.84E−19
	SD	**1.21E−85**	4.68E+00	7.60E+00	2.93E+00	5.48E+00	1.61E+00	7.15E+00	1.39E+01	3.94E+00	4.13E+00	2.23E−18
$f_{20}(x)$	AB	**1.53E−93**	2.09E+56	9.12E+57	2.10E+02	5.33E+01	6.21E+04	9.07E+02	2.30E+03	1.03E+55	5.91E+02	5.99E−49
	MD	**1.61E−98**	2.70E+54	4.15E+50	1.96E+02	4.18E+01	1.26E+03	9.00E+02	2.25E+03	1.87E+54	5.92E+02	4.67E−49
	SD	**8.34E−93**	6.96E+56	4.93E+58	4.53E+01	3.28E+01	1.05E+05	3.32E+02	3.31E+02	1.94E+55	1.59E+02	4.24E−49
$f_{21}(x)$	AB	8.20E+00	2.37E+03	3.45E+04	7.79E+02	2.62E+01	2.03E+04	6.66E+03	1.49E+05	4.77E+04	8.67E−01	**6.67E−02**
	MD	8.00E+00	2.42E+03	3.47E+04	7.81E+02	2.60E+01	2.02E+04	5.00E+00	1.49E+05	4.79E+04	1.00E+00	**0.00E+00**
	SD	1.88E+00	4.76E+02	1.01E+04	1.19E+02	2.74E+00	2.51E+03	1.06E+04	1.75E+04	4.86E+03	9.00E−01	**2.54E−01**
$f_{22}(x)$	AB	3.57E+01	1.34E+07	1.89E+07	1.86E+07	2.73E+05	1.39E+07	3.71E+06	1.10E+06	2.60E+07	1.57E+06	**−2.62E+02**
	MD	3.80E+01	1.26E+07	1.85E+07	1.92E+07	2.86E+05	1.38E+07	2.61E+06	**−2.45E+03**	2.64E+07	1.20E+06	−2.52E+02
	SD	**6.32E+00**	2.60E+06	5.91E+06	3.31E+06	1.39E+05	1.40E+06	3.13E+06	2.20E+06	2.73E+06	1.57E+06	1.23E+02
$f_{23}(x)$	AB	**7.31E−135**	5.13E+03	1.26E+03	9.06E+02	1.69E+02	4.45E+02	7.45E+02	6.95E+02	2.52E+03	8.81E+02	1.23E−23
	MD	**2.08E−150**	5.17E+03	8.42E+02	9.11E+02	1.75E+02	4.45E+02	7.35E+02	7.21E+02	2.27E+03	8.81E+02	1.75E−25
	SD	**3.97E−134**	1.87E+03	1.35E+03	7.81E+01	3.27E+01	3.92E+01	1.35E+02	1.79E+02	1.60E+03	2.93E+02	3.79E−23

Below in Table 9, it can be observed that SBLA gets better results than ABC, BA, HS, MFO, SA, and CSA in every multimodal function at 50 dimensions, NaN is used when both algorithms found the best solution for the 30 individual runs, the values that were consistent in every dimensional test were only DE and SA in f_{11}, also with PSO and GWO in f_{12}, MFO and PSO gets similar results to SBLA in f_{11} and GWO in f_8, even if those values were lower than the validation of Wilcoxon SBLA obtained better results and seems like the similitude was function dependent, these results shows that our proposed algorithm solves multimodal functions adequately even in higher dimensions.

4.3. Composite functions

Details of composite functions can be found in Appendix Table A.3; there are four functions, that cover from f_{24} to f_{26}, these functions have the characteristic of having to solve multiple functions at the same time, even if they do not have the global optimum on the same values, the algorithm must be able to find the balance between all the functions to obtain the best solution. The results are shown in Tables 10–12 for 30, 50 and 100 dimensions.

Table 8
Results of multimodal functions 100D.

		SBLA	ABC	BA	DE	FA	HS	MFO	SA	CSA	PSO	GWO
$f_6(x)$	AB	**3.55E−16**	2.06E+01	1.56E+01	1.90E+01	2.35E+00	1.81E+01	1.93E+01	7.24E−02	1.90E+01	4.89E+00	1.86E−14
	MD	**0.00E+00**	2.06E+01	1.56E+01	1.90E+01	2.35E+00	1.81E+01	1.92E+01	0.00E+00	1.90E+01	2.07E+00	1.78E−14
	SD	**1.08E−15**	1.24E−01	8.45E−01	1.98E−01	2.00E−01	1.64E−01	4.66E−01	3.46E−01	1.50E−01	4.42E+00	4.14E−15
$f_7(x)$	AB	9.93E−01	2.60E+07	1.17E+00	3.39E+06	3.81E+02	4.09E+06	1.72E+06	3.56E+07	6.16E+06	1.97E+05	**6.67E−01**
	MD	1.00E+00	2.63E+07	6.89E−01	3.39E+06	3.56E+02	4.11E+06	1.11E+06	3.47E+07	6.31E+06	9.45E+04	**6.67E−01**
	SD	1.74E−02	3.37E+06	1.80E+00	3.49E+05	1.39E+02	3.23E+05	1.68E+06	6.15E+06	9.30E+05	2.83E+05	**3.44E−08**
$f_8(x)$	AB	**0.00E+00**	1.99E+03	5.33E+02	5.07E+02	1.35E+00	7.41E+02	1.96E+02	2.56E+03	1.04E+03	3.32E+01	2.67E−04
	MD	**0.00E+00**	2.02E+03	5.44E+02	5.13E+02	1.35E+00	7.39E+02	1.81E+02	2.54E+03	1.03E+03	1.52E−02	**0.00E+00**
	SD	**0.00E+00**	1.54E+02	1.55E+02	3.19E+01	7.72E−02	4.01E+01	1.30E+02	1.67E+02	5.95E+01	5.02E+01	1.46E−03
$f_9(x)$	AB	**0.00E+00**	1.76E+01	4.59E−06	3.58E+00	3.09E−05	1.78E+00	1.93E−01	1.83E+01	2.40E+00	5.66E−07	1.97E−238
	MD	**0.00E+00**	1.79E+01	4.48E−06	3.47E+00	2.76E−05	1.76E+00	1.54E−06	1.89E+01	2.45E+00	5.07E−07	8.65E−246
	SD	**0.00E+00**	1.95E+00	1.38E−06	5.79E−01	1.92E−05	1.97E−01	4.39E−01	4.02E+00	5.02E−01	4.27E−07	**0.00E+00**
$f_{10}(x)$	AB	9.15E+00	8.39E+02	4.30E+01	4.05E+02	4.47E+01	2.52E+02	1.59E+02	1.15E+03	4.03E+02	4.98E+01	**6.13E+00**
	MD	9.20E+00	8.44E+02	4.01E+01	4.06E+02	4.15E+01	2.53E+02	1.50E+02	1.19E+03	4.07E+02	4.85E+01	**6.08E+00**
	SD	2.53E−01	6.24E+01	1.22E+01	2.24E+01	1.27E+01	1.51E+01	3.93E+01	1.60E+02	2.15E+01	1.99E+01	3.28E−01
$f_{11}(x)$	AB	**0.00E+00**	7.85E−01	4.10E−14	**0.00E+00**	1.11E−07	5.61E−02	1.70E−06	**0.00E+00**	3.78E−01	9.77E−06	7.33E−15
	MD	**0.00E+00**	8.12E−01	2.81E−14	**0.00E+00**	1.03E−07	5.62E−02	**0.00E+00**	**0.00E+00**	3.80E−01	**0.00E+00**	5.95E−22
	SD	**0.00E+00**	1.03E−01	2.78E−14	**0.00E+00**	6.16E−08	8.04E−03	9.33E−06	**0.00E+00**	3.44E−02	5.35E−05	4.01E−14
$f_{12}(x)$	AB	**0.00E+00**	1.45E+36	1.70E+20	1.78E−23	3.72E−108	1.73E−39	**0.00E+00**	5.10E−64	1.86E+20	**0.00E+00**	**0.00E+00**
	MD	**0.00E+00**	1.42E+32	1.28E+10	3.97E−29	4.05E−116	3.63E−42	**0.00E+00**	7.0E−255	1.23E+19	**0.00E+00**	**0.00E+00**
	SD	**0.00E+00**	6.18E+36	6.85E+20	8.49E−23	1.94E−107	5.17E−39	**0.00E+00**	2.79E−63	3.66E+20	**0.00E+00**	**0.00E+00**
$f_{13}(x)$	AB	3.67E+01	1.47E+02	1.18E+02	9.90E+01	3.53E+01	1.19E+02	5.94E+01	2.69E+02	1.66E+02	4.67E+01	**3.00E+01**
	MD	3.60E+01	1.47E+02	1.18E+02	9.90E+01	3.40E+01	1.20E+02	5.70E+01	2.67E+02	1.67E+02	4.60E+01	**3.00E+01**
	SD	2.26E+00	6.16E+00	1.29E+01	2.86E+00	3.26E+00	3.80E+00	1.37E+01	1.24E+01	6.84E+00	7.98E+00	**1.83E−01**
$f_{14}(x)$	AB	**0.00E+00**	7.53E+04	6.55E−02	3.84E+04	7.40E+01	2.07E+04	1.14E+04	1.38E+04	2.38E+04	5.72E+03	1.06E−06
	MD	**0.00E+00**	7.80E+04	6.57E−02	3.87E+04	7.05E+01	2.09E+04	1.08E+04	1.22E+04	2.50E+04	4.89E+03	2.47E−07
	SD	**0.00E+00**	9.93E+03	1.05E−02	2.95E+03	3.04E+01	1.84E+03	6.00E+03	1.24E+04	3.46E+03	3.66E+03	3.05E−06
$f_{15}(x)$	AB	2.61E+05	8.63E+11	6.29E+10	1.72E+11	5.13E+06	1.55E+11	2.93E+10	1.12E+12	2.12E+11	**1.96E+03**	1.24E+05
	MD	2.65E+05	8.57E+11	5.98E+10	1.70E+11	4.81E+06	1.57E+11	4.14E+08	1.13E+12	2.10E+11	**1.02E+03**	1.27E+05
	SD	1.35E+04	7.69E+10	3.64E+10	1.90E+10	2.06E+06	1.27E+10	3.92E+10	1.59E+11	2.45E+10	**2.38E+03**	1.46E+04
$f_{16}(x)$	AB	**3.77E+01**	1.83E+03	1.90E+02	2.79E+02	4.76E+01	3.25E+02	2.05E+02	2.56E+03	4.68E+02	7.22E+01	3.83E+01
	MD	**3.78E+01**	1.87E+03	1.92E+02	2.81E+02	4.76E+01	3.28E+02	1.61E+02	2.48E+03	4.71E+02	6.38E+01	3.84E+01
	SD	**7.08E−01**	2.09E+02	7.31E+01	2.14E+01	1.75E+00	2.62E+01	1.08E+02	3.41E+02	5.33E+01	2.13E+01	7.44E−01
$f_{17}(x)$	AB	**0.00E+00**	1.55E+03	2.35E+02	9.04E+02	3.75E+02	7.97E+02	6.06E+02	1.45E+03	1.14E+03	4.41E+02	**0.00E+00**
	MD	**0.00E+00**	1.56E+03	2.36E+02	9.08E+02	3.75E+02	7.95E+02	5.92E+02	1.50E+03	1.14E+03	4.38E+02	**0.00E+00**
	SD	**0.00E+00**	5.26E+01	5.34E+01	1.95E+01	4.99E+01	2.71E+01	9.90E+01	1.67E+02	3.31E+01	7.56E+01	**0.00E+00**
$f_{18}(x)$	AB	9.89E+01	7.28E+06	1.55E+02	4.11E+05	6.98E+02	1.05E+06	7.60E+05	1.79E+06	2.66E+06	1.28E+06	**9.70E+01**
	MD	9.89E+01	7.32E+06	1.45E+02	4.21E+05	6.68E+02	1.04E+06	6.80E+05	1.66E+06	2.65E+06	1.28E+06	**9.69E+01**
	SD	**3.63E−02**	8.21E+05	4.90E+01	4.45E+04	1.51E+02	8.38E+04	3.37E+05	6.09E+05	3.33E+05	3.68E+05	8.59E−01
$f_{19}(x)$	AB	**2.04E−210**	9.58E+01	4.49E+01	9.45E+01	7.93E+01	7.50E+01	9.21E+01	1.81E+01	7.37E+01	5.67E+01	1.00E−18
	MD	**8.83E−222**	9.59E+01	4.39E+01	9.46E+01	7.79E+01	7.49E+01	9.22E+01	1.81E+01	7.47E+01	5.63E+01	3.15E−21
	SD	**0.00E+00**	9.27E−01	6.97E+00	1.34E+00	1.08E+01	9.69E−01	2.02E+00	1.28E+00	2.93E+00	4.23E+00	4.03E−18
$f_{20}(x)$	AB	**3.82E−225**	1.19E+135	1.81E+119	1.98E+73	7.80E+02	8.77E+60	1.67E+03	5.06E+03	4.98E+119	1.94E+03	1.93E−69
	MD	**2.05E−232**	1.51E+125	3.22E+105	1.16E+66	7.59E+02	1.04E+56	1.55E+03	5.08E+03	6.32E+116	1.88E+03	1.23E−69
	SD	**0.00E+00**	5.70E+135	9.92E+119	8.38E+73	3.85E+02	3.31E+61	4.32E+02	2.91E+02	2.69E+120	2.95E+02	1.53E−69
$f_{21}(x)$	AB	6.87E+00	2.23E+05	6.23E+04	5.67E+04	6.04E+01	8.24E+04	1.97E+04	3.22E+05	1.14E+05	4.68E+03	**0.00E+00**
	MD	7.00E+00	2.27E+05	5.64E+04	5.65E+04	5.75E+01	8.25E+04	2.02E+04	3.22E+05	1.16E+05	3.25E+01	**0.00E+00**
	SD	1.81E+00	1.97E+04	1.84E+04	3.18E+03	1.32E+01	4.78E+03	1.60E+04	3.31E+04	6.12E+03	5.71E+03	**0.00E+00**
$f_{22}(x)$	AB	8.69E+01	1.95E+09	6.65E+08	1.30E+09	1.41E+07	8.90E+08	1.94E+08	3.31E+09	1.03E+09	6.44E+07	**−1.95E+02**
	MD	8.71E+01	1.98E+09	5.96E+08	1.31E+09	1.37E+07	9.13E+08	1.72E+08	3.22E+09	1.03E+09	6.02E+07	**−1.88E+02**
	SD	**4.29E+00**	1.68E+08	2.54E+08	6.40E+07	4.37E+06	7.29E+07	1.12E+08	4.68E+08	8.35E+07	4.04E+07	9.89E+01
$f_{23}(x)$	AB	**0.00E+00**	2.57E+04	4.18E+03	2.40E+03	7.76E+02	1.35E+03	1.98E+03	1.43E+03	1.54E+11	3.05E+03	9.20E−21
	MD	**0.00E+00**	2.45E+04	2.83E+03	2.40E+03	7.64E+02	1.37E+03	2.00E+03	1.46E+03	5.30E+05	3.06E+03	1.64E−22
	SD	**0.00E+00**	1.12E+04	2.96E+03	7.96E+01	8.35E+01	8.03E+01	2.12E+02	1.85E+02	4.12E+11	3.86E+02	3.22E−20

Table 10 express the results from 30-dimensional composite functions, in which SBLA gets the best results against every other algorithm; this is represented in boldface.

In Table 11 SBLA gets the best results for f_{24} and f_{27}, GWO gets the best solutions to f_{25} and f_{26} by a small margin and SBLA gets the second-place, both algorithms can solve composite functions at 50 dimensions adequately.

The results for composite functions at 100 dimensions are shown in Table 12, the best results for f_{24} and f_{27} are obtained by the proposed algorithm. GWO gets the best AB and SD in functions f_{25} and f_{26}, though SBLA gets a better SD, this can be interpreted as SBLA not getting the best results but being more reliable and GWO finding better solutions but with the probability of getting worst solutions that SBLA occasionally.

The results from Tables 10 to 12 are validated using the Wilcoxon test and Bonferroni correction, which compares the proposed method with every other algorithm with a significance level of $5.0E{-}2$ and it goes down to $1.7E{-}3$ applying the Bonferroni correction. The configuration on the tables is set: SBLA vs. ABC, SBLA vs. BA, SBLA vs. DE, SBLA vs. FA, SBLA vs. HS, SBLA vs. MFO, SBLA vs. SA, SBLA vs. CSA, SBLA vs. PSO, and SBLA vs. GWO. These results are visualized in Table 13.

Table 9
Wilcoxon of multimodal functions 50D.

	SBLA VS ABC	SBLA VS BA	SBLA VS DE	SBLA VS FA	SBLA VS HS	SBLA VS MFO	SBLA VS SA	SBLA VS CSA	SBLA VS PSO	SBLA VS GWO
$f_6(x)$	3.15E−12▲	3.15E−12▲	3.15E−12▲	3.15E−12▲	3.15E−12▲	3.15E−12▲	8.22E−1▲	3.15E−12▲	3.15E−12▲	2.76E−13▲
$f_7(x)$	3.02E−11▲	6.28E−06▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	6.52E−09▲	3.02E−11▼
$f_8(x)$	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	3.34E−01▲
$f_9(x)$	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲
$f_{10}(x)$	3.02E−11▲	3.02E−11▲	3.02E−11▲	4.92E−01▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	1.02E−01▲	3.02E−11▼
$f_{11}(x)$	1.21E−12▲	1.21E−12▲	NaN▶	1.21E−12▲	1.21E−12▲	8.15E−02▲	NaN▶	1.21E−12▲	NaN▶	1.21E−12▲
$f_{12}(x)$	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	NaN▶	1.21E−12▶	1.21E−12▲	NaN▶	NaN▶
$f_{13}(x)$	6.84E−02▶	2.25E−11▲	1.18E−11▼	2.39E−09▼	2.03E−11▲	2.75E−04▲	2.73E−11▲	2.21E−11▲	1.10E−05▶	1.18E−11▼
$f_{14}(x)$	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲
$f_{15}(x)$	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	9.51E−06▲	3.02E−11▲	3.02E−11▲	3.02E−11▼	3.02E−11▼
$f_{16}(x)$	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	1.33E−02▶
$f_{17}(x)$	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	NaN▶
$f_{18}(x)$	3.02E−11▲	3.79E−01▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▶
$f_{19}(x)$	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲
$f_{20}(x)$	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲
$f_{21}(x)$	2.62E−11▲	2.63E−11▲	2.62E−11▲	2.49E−11▲	2.63E−11▲	2.15E−01▲	2.66E−11▲	2.63E−11▲	1.10E−05▼	6.51E−11▼
$f_{22}(x)$	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	6.63E−01▲	3.02E−11▲	5.57E−10▲	3.02E−11▼
$f_{23}(x)$	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲

Table 10
Results of composite functions 30D.

		SBLA	ABC	BA	DE	FA	HS	MFO	SA	CSA	PSO	GWO
$f_{24}(x)$	AB	**1.48E−76**	2.85E+08	8.71E+07	5.44E+03	2.44E+02	2.12E+04	2.51E+04	9.69E+04	2.83E+05	8.37E+03	3.87E−30
	MD	**4.90E−79**	2.72E+06	4.32E+04	5.33E+03	2.35E+02	2.13E+04	2.01E+04	9.49E+04	8.37E+04	1.00E+04	2.92E−30
	SD	**4.64E−76**	1.12E+09	3.92E+08	7.24E+02	4.57E+01	2.08E+03	2.14E+04	2.20E+04	4.45E+05	8.78E+03	2.86E−30
$f_{25}(x)$	AB	**2.90E+01**	1.36E+03	9.31E+02	6.68E+02	2.43E+02	1.01E+03	1.55E+02	2.14E+03	1.62E+03	1.52E+02	4.90E+01
	MD	**2.90E+01**	1.37E+03	8.89E+02	6.78E+02	2.43E+02	1.02E+03	9.92E+01	2.23E+03	1.63E+03	1.48E+02	4.90E+01
	SD	**1.27E−04**	1.54E+02	1.52E+02	5.04E+01	2.38E+01	6.53E+01	1.32E+02	5.54E+02	1.22E+02	2.81E+01	1.58E−04
$f_{26}(x)$	AB	**3.20E+01**	8.35E+08	7.73E+07	2.53E+07	1.23E+04	1.06E+08	7.24E+01	1.72E+09	3.34E+08	8.90E+02	5.40E+01
	MD	**3.20E+01**	8.40E+08	5.96E+07	2.46E+07	9.09E+03	1.03E+08	7.21E+01	1.52E+09	3.36E+08	8.44E+02	5.40E+01
	SD	**2.41E−05**	1.84E+08	6.41E+07	6.75E+06	1.19E+04	2.05E+07	1.68E+01	7.08E+08	6.95E+07	2.25E+02	3.07E−05
$f_{27}(x)$	AB	**2.90E+01**	1.06E+10	1.23E+06	6.56E+02	2.41E+02	1.00E+03	7.09E+02	2.90E+03	2.52E+05	4.34E+02	4.90E+01
	MD	**2.90E+01**	2.76E+06	1.75E+03	6.54E+02	2.36E+02	1.02E+03	6.71E+02	2.90E+03	2.57E+04	3.99E+02	4.90E+01
	SD	**0.00E+00**	4.65E+10	5.87E+06	3.20E+01	3.51E+01	9.19E+01	4.52E+02	8.03E+02	7.61E+05	3.33E+02	2.64E−14

Table 11
Results of composite functions 50D.

		SBLA	ABC	BA	DE	FA	HS	MFO	SA	CSA	PSO	GWO
$f_{24}(x)$	AB	**1.55E−116**	5.77E+06	1.02E+10	1.02E+03	3.44E+01	1.91E+04	6.06E+04	1.61E+05	2.42E+05	8.37E+03	2.01E−49
	MD	**1.32E−120**	5.00E+05	4.48E+04	1.04E+03	3.47E+01	1.91E+04	5.02E+04	1.60E+05	6.82E+04	1.00E+04	1.13E−49
	SD	**7.47E−116**	2.51E+07	5.60E+10	1.51E+02	4.63E+00	1.57E+03	3.16E+04	3.46E+04	6.07E+05	8.78E+03	2.55E−49
$f_{25}(x)$	AB	4.90E+01	8.08E+02	8.64E+02	4.57E+02	1.87E+02	9.18E+02	4.19E+02	3.64E+03	1.57E+03	1.46E+02	**4.90E+01**
	MD	4.90E+01	8.09E+02	8.25E+02	4.56E+02	1.84E+02	9.26E+02	3.90E+02	3.98E+03	1.56E+03	1.42E+02	**4.90E+01**
	SD	1.92E−04	5.38E+01	2.37E+02	1.74E+01	3.35E+01	7.12E+01	2.56E+02	1.07E+03	1.41E+02	2.44E+01	**1.43E−04**
$f_{26}(x)$	AB	5.40E+01	4.72E+08	4.32E+07	2.13E+06	9.69E+02	8.97E+07	6.83E+07	3.37E+09	3.26E+08	5.33E+02	**5.40E+01**
	MD	5.40E+01	4.47E+08	2.90E+07	2.14E+06	9.17E+02	9.28E+07	3.93E+02	3.29E+09	3.25E+08	5.01E+02	**5.40E+01**
	SD	3.02E−05	1.37E+08	4.76E+07	7.04E+05	2.07E+02	1.47E+07	1.55E+08	5.39E+08	7.80E+07	1.32E+02	**2.61E−05**
$f_{27}(x)$	AB	**4.90E+01**	2.95E+06	7.03E+05	4.42E+02	1.75E+02	9.37E+02	1.75E+03	5.70E+03	6.32E+05	4.41E+02	4.90E+01
	MD	**4.90E+01**	9.56E+04	1.57E+03	4.40E+02	1.62E+02	9.32E+02	1.80E+03	4.94E+03	1.48E+04	4.08E+02	4.90E+01
	SD	**0.00E+00**	9.37E+06	3.58E+06	2.36E+01	3.54E+01	7.07E+01	9.50E+02	2.24E+03	1.87E+06	3.32E+02	2.25E−14

Table 12
Results of composite functions 100D.

		SBLA	ABC	BA	DE	FA	HS	MFO	SA	CSA	PSO	GWO
$f_{24}(x)$	AB	**1.49E−224**	1.73E+32	1.10E+18	6.99E+04	6.56E+01	8.16E+04	1.50E+05	3.79E+05	2.88E+19	5.42E+04	5.87E−70
	MD	**1.71E−233**	8.58E+27	2.61E+07	6.99E+04	6.49E+01	8.20E+04	1.41E+05	3.64E+05	1.60E+17	5.02E+04	3.62E−70
	SD	**0.00E+00**	7.90E+32	6.04E+18	6.69E+03	1.75E+01	4.39E+03	4.99E+04	7.61E+04	1.01E+20	2.42E+04	7.64E−70
$f_{25}(x)$	AB	9.90E+01	7.69E+03	1.56E+03	3.19E+03	5.59E+02	3.10E+03	1.11E+03	7.71E+03	3.78E+03	4.34E+02	**9.90E+01**
	MD	9.90E+01	7.85E+03	1.51E+03	3.18E+03	5.53E+02	3.11E+03	1.16E+03	7.84E+03	3.79E+03	3.82E+02	**9.90E+01**
	SD	**3.16E−04**	5.68E+02	4.44E+02	1.51E+02	5.93E+01	1.06E+02	3.75E+02	1.47E+03	1.65E+02	1.28E+02	3.27E−04
$f_{26}(x)$	AB	1.09E+02	5.22E+09	5.47E+07	9.53E+08	8.30E+03	7.29E+08	1.19E+08	6.94E+09	1.04E+09	2.89E+03	**1.09E+02**
	MD	1.09E+02	5.34E+09	2.97E+07	9.60E+08	7.29E+03	7.27E+08	1.46E+04	7.16E+09	1.02E+09	2.72E+03	**1.09E+02**
	SD	**2.74E−05**	5.69E+08	7.80E+07	1.40E+08	2.93E+03	8.71E+07	1.86E+08	1.09E+09	1.50E+08	4.62E+02	5.88E−05
$f_{27}(x)$	AB	**9.90E+01**	2.28E+32	1.59E+21	3.33E+03	4.90E+02	3.12E+03	4.48E+03	1.19E+04	2.57E+19	2.82E+03	9.90E+01
	MD	**9.90E+01**	3.09E+26	1.04E+10	3.31E+03	4.79E+02	3.15E+03	4.07E+03	1.05E+04	1.94E+17	2.96E+03	9.90E+01
	SD	**0.00E+00**	8.82E+32	8.71E+21	2.41E+02	6.58E+01	1.86E+02	1.84E+03	4.04E+03	5.98E+19	1.25E+03	4.78E−14

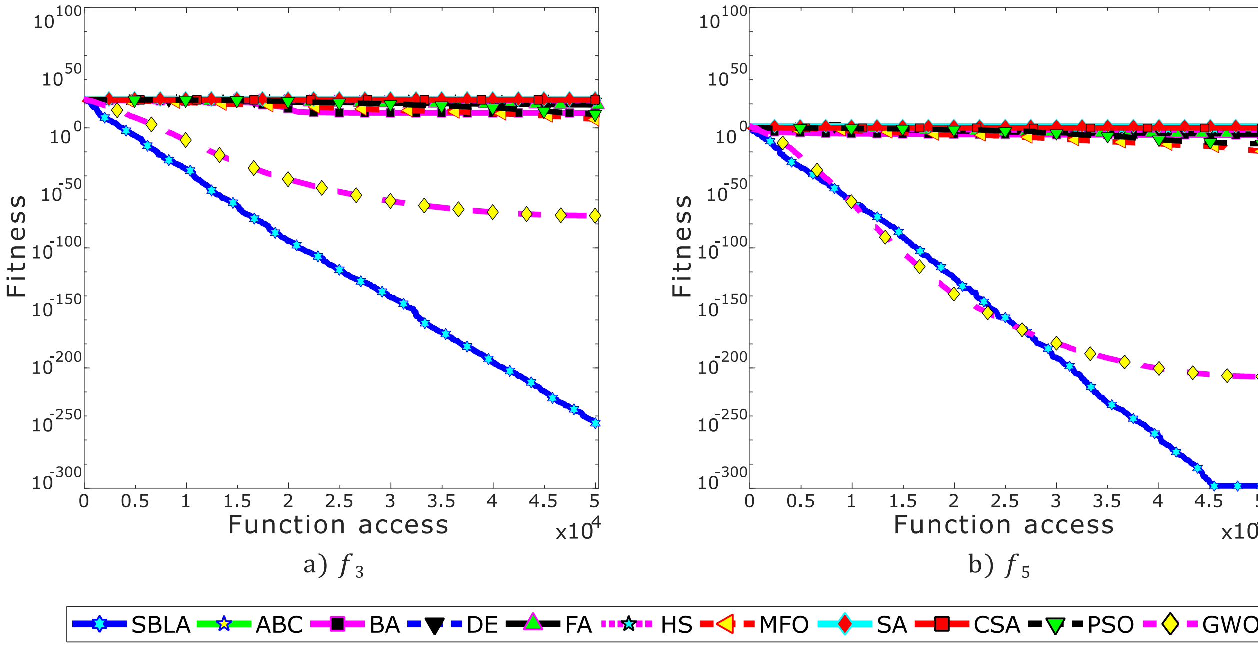

Fig. 12. Unimodal convergence graph for (a) f_3 and (b) f_5 in 30 dimensions.

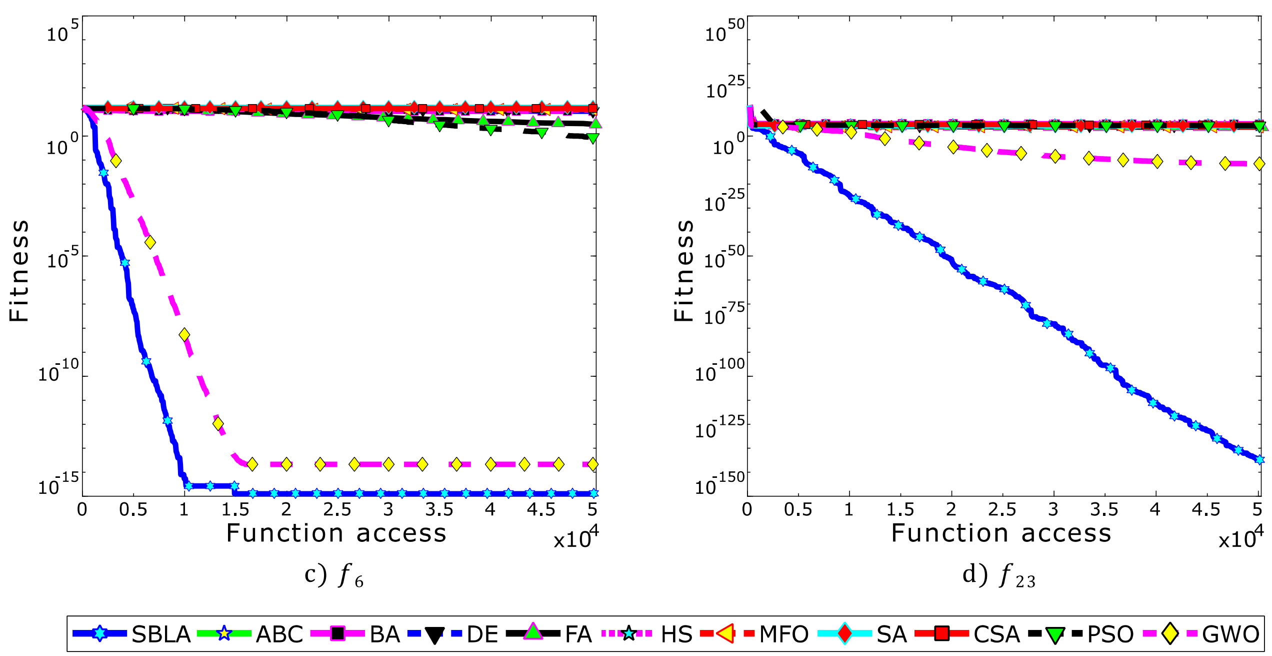

Fig. 13. Multimodal convergence graph for (c) f_6 and (d) f_{23} in 30 dimensions.

The symbols ▲, ▼, and ► represent the performance of SBLA algorithm against the other algorithms, symbols are used to represent the difference obtained from the AB of the algorithms; ▲ means that SBLA had a better performance, ▼ shows a worse performance, and ► specifies that the solutions are not meaningfully different between the methods.

In Table 13 it can be observed the Wilcoxon of 50-dimensional composite functions, in which SBLA obtained the best results through all the functions, and every value on the table is lower than $1.7E-3$, these values were similar in 30 and 100 dimensions, proving than even in high dimension, the proposed algorithm solves the functions using his own methods, and it cannot be compared to any of the algorithms listed in the table.

4.4. Convergence analysis

The algorithms improve their solutions through the iterative process, the convergence analysis has the values for the best solution each iteration, the performance results of each algorithm in a single execution have been plotted to illustrate the speed of convergence. The axis of the graph represents the fitness obtained in the function and the function access employed.

The convergence analysis considers six functions: f_3, f_5, f_6, f_{23}, f_{24}, and f_{26}. The selection of the functions was made considering two unimodal functions (f_3 and f_5), two multimodal functions (f_6 and f_{23}) and two composite functions (f_{24} and f_{26}), the test

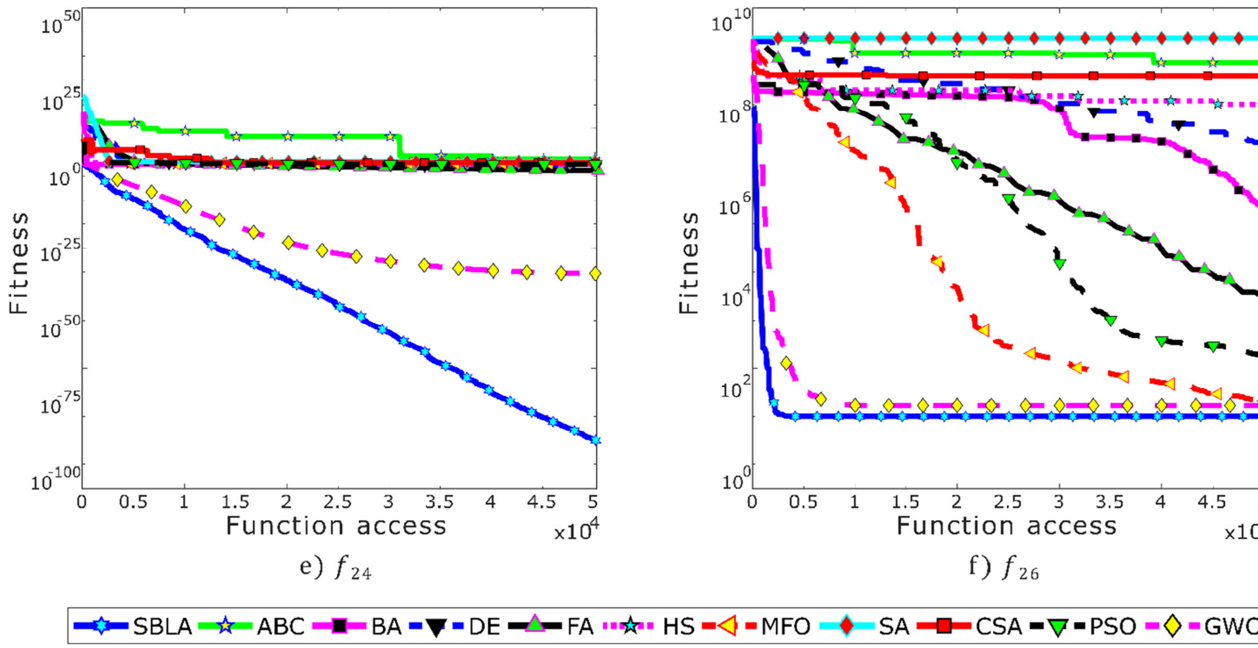

Fig. 14. Composite convergence graph for (e) f_{24} and (f) f_{26} in 30 dimensions.

Table 13
Wilcoxon of composite functions 50D.

	SBLA VS ABC	SBLA VS BA	SBLA VS DE	SBLA VS FA	SBLA VS HS	SBLA VS MFO	SBLA VS SA	SBLA VS CSA	SBLA VS PSO	SBLA VS GWO
$f_{24}(x)$	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲
$f_{25}(x)$	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▶
$f_{26}(x)$	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▲	3.02E−11▶
$f_{27}(x)$	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	1.21E−12▲	3.13E−13▶

evaluates the performance of algorithms through time in 30 dimensions. The graphs with the results of convergence are shown in Figs. 8–10.

In Fig. 12, the convergence curves of the unimodal functions f_3 and f_5 are visualized in (a) and (b) respectively. Multimodal functions f_6 and f_{23} can be observed in (c) and (d) respectively from Fig. 13 while in Fig. 14 the convergence of the composite functions f_{24} and f_{26} are visualized in (e) and (f) respectively. It is clear that the proposed approach converges faster than their competitors in every function, and it should be noted that SBLA constantly decreases until it finds the solution.

5. Engineering problems

We show in this section the ability to solve real-world problems of the proposed algorithm; we implemented three well-known engineering optimization problems to test against recognized optimization techniques; ABC, BA, DE, FA, HS, MFO, SA, CSA, PSO and GWO through 30 runs, to make a fair comparison, each of the algorithms had a limit of 50,000 function access, in Table 14 to 16 the value AB represent the average best, BST is the best solution and WST is the worst solution got, the x_n are the positions of the best solution for each of the algorithms.

Said problems are the welded beam design problem [80] in which the algorithm must minimize the cost, the FM synthesizer [81] that must be optimized to optimize the signal, and lastly the rolling element bearing [82] where the algorithm selects the best values of 10 variables with the purpose of maximizing the load carrying capacity of the bearing. In these optimization problems, SBLA algorithm gets the best average values; this makes SBLA a reliable algorithm to use in real-world problems.

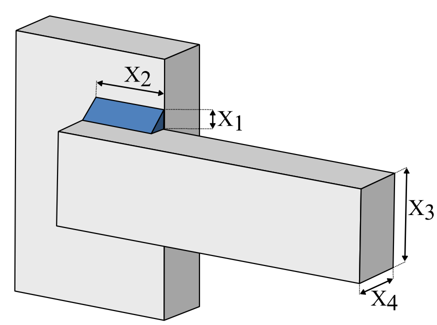

Fig. 15. Welded beam design problem.

5.1. Welded beam

The welded beam is a problem designed for the minimum cost in the function of the weld dimensions, weld labor, and beam material. It is an optimization problem for minimum cost by varying the weld and member dimensions; there are four decision variables; x_1 and x_2 represents the weld dimensions, x_3 and x_4 denotes the dimensions of the bar to weld (see Fig. 15).

$$P = 6000 \tag{16}$$

$$LL = 14 \tag{17}$$

$$E = 30e6 \tag{18}$$

$$G = 12e6 \tag{19}$$

$$\tau_{max} = 13600 \tag{20}$$

$$\sigma_{max} = 30000 \tag{21}$$

$$\delta_{max} = 0.25 \quad (22)$$

$$M = P * \left(LL + \frac{x_2}{2}\right) \quad (23)$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{(x_1 + x_3)}{2}\right)^2} \quad (24)$$

$$J = 2 * \left(\sqrt{2} * x_1 * x_2 * \left(\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right)\right) \quad (25)$$

$$\sigma(X) = \frac{6 * P * LL}{(x_3^2) * x_4} \quad (26)$$

$$\delta(X) = \frac{4 * P * (LL^3)}{E * x_3^3 * x_4} \quad (27)$$

$$Pc(X) = \frac{4.013 * E * \sqrt{\frac{x_3^2 * x_4^6}{36}}}{LL^2} * \left(1 - \left(\frac{x_3}{2 * LL}\right) * \sqrt{\frac{E}{4 * G}}\right) \quad (28)$$

$$\tau_1 = \frac{P}{\sqrt{2} * x_1 * x_2} \quad (29)$$

$$\tau_2 = \frac{M * R}{J} \quad (30)$$

$$\tau(X) = \sqrt{\tau_1^2 + \tau_2^2 + 2 * \tau_1 * \tau_2 * \frac{x_2}{2 * R}} \quad (31)$$

There are nine constrains, which are;
the shear stress in weld cannot exceed the maximum allowable for the material

$$g_1(X) = \tau(X) - \tau_{max} \leq 0 \quad (32)$$

the bending stress in the beam cannot exceed a maximum yield strength

$$g_2(X) = \sigma(X) - \sigma_{max} \leq 0 \quad (33)$$

the bar thickness must be greater than the weld thickness

$$g_3(X) = x_1 - x_4 \leq 0 \quad (34)$$

the cost of the weld material and the bar cannot exceed five dollars

$$g_4(X) = 1.10471 x_1^2 * x_2 + 0.04811 x_3 * x_4 * (14 + x_2) - 5 \leq 0 \quad (35)$$

the weld thickness must be larger than a defined minimum of 0.125

$$g_5(X) = 0.125 - x_1 \leq 0 \quad (36)$$

the end deflection of the beam

$$g_6(X) = \delta(X) - \delta_{max} \leq 0 \quad (37)$$

the buckling load on the bar

$$g_7(X) = P - P_c(X) \leq 0 \quad (38)$$

lower bound constraints

$$g_8(X) = x_i < low_i \quad (39)$$

upper bound constraints.

$$g_9(X) = x_i > up_i \quad (40)$$

Where:

Decision variable	x_1	x_2	x_3	x_4
Upper bound	2	10	10	2
Lower bound	0.1	0.1	0.1	0.1

If the algorithm met any of the constraints it must be punished with a large fitness value, the value used was 1*E+300, and in the other case, the fitness formula is applied.

$$minimize\, f(X) \quad \begin{cases} 1.10471 x_1^2 * x_2 + 0.04811 x_3 * x_4 * (14 + x_2) \\ \quad \vec{g}(X) \leq 0 \\ 1 * E + 300 \quad \vec{g}(X) > 0 \end{cases} \quad (41)$$

The results for the welded beam problem are shown in Table 14 in which MFO gets the best solution. However, SBLA is more reliable since its average solutions performed better than MFO, it also obtained a better result than MFO in his worst solution.

5.2. FM synthesizer

A frequency-modulated sound wave synthesis is essential in several modern music systems. The FM synthesizer is a minimization problem with six decision variables; x_1 *and* x_2 applied to the carrier signal and x_{3-6} to the modulator signals.

The problem has the same side constraints for the bounds of every variable, Eqs. (42) and (43) represent the lower and upper bounds.

$$g_1(X) = x_i < -6.4 \quad (42)$$

$$g_2(X) = x_i > 6.35 \quad (43)$$

If the algorithm met any of the constraints it must be punished with a huge fitness value, the value used was 1*E+300, and in the other case, the fitness formula is applied.

$$t = 100 \quad (44)$$

$$theta = 2^{*}\mathrm{pi}/100 \quad (45)$$

$$y\_t = x_1 * \sin(x_2 * t * theta + x_3 * \sin(x_4 * t * theta + x_5 * \sin(x_6 * t * theta))) \quad (46)$$

$$y\_0\_t = 1 * \sin(5 * t * theta - 1.5 * \sin(4.8 * t * theta + 2 * \sin(4.9 * t * theta))) \quad (47)$$

The outer sine is the carrier, and the inner sines are the modulator signals, x_1, x_3 *and* x_5 represents the amplitude of the signals while x_2, x_4 *and* x_6 denotes the angular frequency.

$$minimize\, f(X) \quad \begin{cases} \sum_0^t y\_t - y\_0\_t & \vec{g}(X) \leq 0 \\ 1 * E + 300 & \vec{g}(X) > 0 \end{cases} \quad (48)$$

In Table 15 it can be observed that SBLA gets the best results in the AB, BST and WST, even if six other algorithms can obtain the best solution for the problem, SBLA obtained most times in the 30 runs being the steadiest algorithm in this problem.

5.3. Rolling element bearing

The design of a rolling element bearing is a problem in which the criterion is to maximize the load carrying capacity of the bearing by optimizing ten decision variables; x_1 is the diameter of balls, x_2 is the mean diameter, x_3 as the number of balls, x_4 and x_5 are the inner and outer ring groove curvatures, x_6 *and* x_7 decide the minimum and the maximum diameter of the rolling element, x_8 defines the ring thickness, x_9 ensures the pitch diameter to be greater than the mean diameter and lastly x_{10} defines the ratio of diameter of the ball to the width of the bearing.

Table 14
Results of welded beam.

	SBLA	ABC	BA	DE	FA	HS	MFO	SA	CSA	PSO	GWO
AB	**1.94E+00**	6.67E+298	2.33E+299	1.00E+300	1.00E+300	1.00E+300	1.94E+00	1.00E+300	4.33E+299	1.00E+300	1.33E+299
BST	1.76E+00	2.63E+00	1.75E+00	1.00E+300	1.00E+300	1.00E+300	**1.72E+00**	1.00E+300	5.02E+00	1.00E+300	1.73E+00
WST	**2.20E+00**	1.00E+300	1.00E+300	1.00E+300	1.00E+300	1.00E+300	3.12E+00	1.00E+300	1.00E+300	1.00E+300	1.00E+300
x_1	0.1916	0.2561	0.1930	0.2328	0.3807	0.4152	0.2057	0.4320	1.5033	1.1128	0.2056
x_2	3.8308	3.4493	3.7761	1.4542	1.9803	1.2243	3.4705	1.4309	0.9104	1.2281	3.4742
x_3	9.1330	7.0043	9.0163	0.8241	1.0449	0.9787	9.0366	1.5894	0.1328	0.5128	9.0355
x_4	0.2053	0.4056	0.2067	0.5657	1.3028	1.6618	0.2057	1.6871	1.4057	0.9384	0.2058

Table 15
Results of FM synthesizer.

	SBLA	ABC	BA	DE	FA	HS	MFO	SA	CSA	PSO	GWO
AB	**3.08E+01**	3.10E+01	3.10E+01	3.10E+01	3.10E+01	3.09E+01	3.10E+01	3.10E+01	3.09E+01	3.09E+01	3.08E+01
BST	**3.01E+01**	**3.01E+01**	3.10E+01	3.10E+01	**3.01E+01**	**3.01E+01**	3.10E+01	3.10E+01	**3.01E+01**	**3.01E+01**	**3.01E+01**
WST	**3.10E+01**	**3.10E+01**	**3.10E+01**	**3.10E+01**	**3.10E+01**	**3.10E+01**	**3.10E+01**	**3.10E+01**	**3.10E+01**	**3.10E+01**	**3.10E+01**
x_1	−1.4334	0.5796	−0.0050	−0.2063	−1.2337	−1.1554	−0.1464	0.0685	−3.4581	−0.5456	0.8415
x_2	−5.3833	4.5397	6.4169	0.2270	−4.7352	−4.6659	3.1733	0.5398	6.1003	−4.8584	5.3735
x_3	1.0351	−1.4852	2.6245	0.4182	−0.8813	0.7538	−6.4000	−1.4881	3.1945	0.7613	−0.8951
x_4	4.9298	4.7347	4.5230	−2.3539	−4.5931	4.7022	−2.5246	2.1923	−3.0062	5.0798	5.0878
x_5	1.8167	−2.4917	1.9966	−4.9969	−2.1744	−2.2140	6.3500	−4.2723	2.4621	−1.7075	−2.4793
x_6	5.0435	−4.9397	−0.2842	−3.7315	5.4857	−5.0265	2.0319	−3.3162	−4.7758	−4.5850	−5.3513

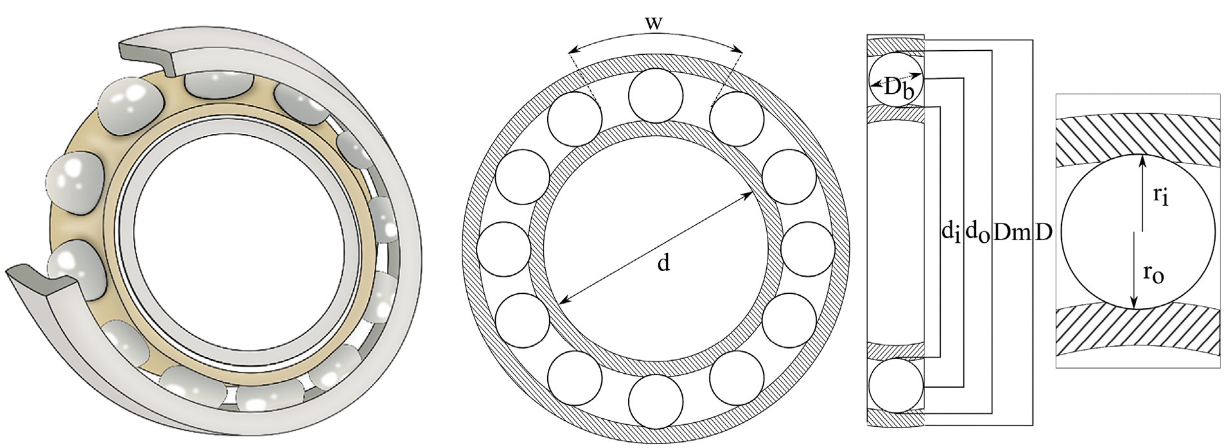

Fig. 16. Rolling element bearing design problem.

In Fig. 16 from left to right there is a rolling element bearing, then is the diagram with d representing the inner diameter and w is the spacing taken by one ball, the next image is a cut from the middle of the rolling element bearing where Db is the diameter of the balls, Dm is the mean diameter, do and di are the inner and outer raceway diameters at the grooves and the last image represent one ball, where ri and ro are the inner and outer ring groove curvature.

There are 12 constraints including the bound constraints of variables; x_1 is the diameter of balls (Db), x_2 the mean diameter (Dm), x_3 represent the number of balls, x_4 and x_5 are the inner and outer ring grooves curvatures (r_i and r_o), x_{6-10} are constrain constants.

$$f_i = \frac{x_4}{x_1} \tag{49}$$

$$f_o = \frac{x_5}{x_1} \tag{50}$$

$$\gamma = \frac{x_1}{x_2} \tag{51}$$

$$fc_1 = 37.91 * \left(1 + \left(1.04 * \left(\frac{1-\gamma}{1+\gamma}\right)^{1.72} * \left(\left(f_i * \frac{2*f_o - 1}{f_o} * (2*f_i - 1)\right)^{0.41}\right)^{\frac{10}{3}}\right)^{-0.3} \tag{52}$$

$$fc_2 = \frac{\gamma^{0.3} * (1-\gamma)^{1.39}}{(1+\gamma)^{\frac{1}{3}}} \tag{53}$$

$$fc_3 = \left(\frac{2*f_i}{2*f_i - 1}\right)^{0.41} \tag{54}$$

$$fc = fc_1 + fc_2 + fc_3 \tag{55}$$

$$D = 80 \tag{56}$$

$$d = 40 \tag{57}$$

$$w = 18 \tag{58}$$

$$T = \frac{D - d - 2*x_1}{4} \tag{59}$$

$$U = \frac{D-d}{2} - 3*T \tag{60}$$

$$\phi = 2*\pi - 2*\cos^{-1}\left(\frac{U^2 + \left(\frac{D}{2} - T - x_1\right)^2 - \left(\frac{d}{2} + T\right)^2}{2*U*\left(\frac{D}{2} - T - x_1\right)}\right) \tag{61}$$

$$d_i = \left(\frac{D-d}{2} - x_1\right) * 2 + d \tag{62}$$

Table 16
Results of the rolling element bearing.

	SBLA	ABC	BA	DE	FA	HS	MFO	SA	CSA	PSO	GWO
AB	**1.68E+04**	−7.00E+299	−8.00E+299	−1.00E+300	−1.00E+300	−1.00E+300	−3.33E+298	−1.00E+300	−1.00E+300	−1.00E+300	−8.67E+299
BST	1.85E+04	9.54E+03	1.86E+04	−1.00E+300	−1.00E+300	−1.00E+300	**1.96E+04**	−1.00E+300	−1.00E+300	−1.00E+300	1.95E+04
WST	**1.53E+04**	−1.00E+300	−1.00E+300	−1.00E+300	−1.00E+300	−1.00E+300	−1.00E+300	−1.00E+300	−1.00E+300	−1.00E+300	−1.00E+300
x_1	13.2319	11.5322	13.1075	12.4002	12.1990	12.3502	13.5922	10.8205	0.7969	10.8297	13.5914
x_2	56.6067	51.0599	57.9063	8.6276	9.7699	11.0008	58.2524	8.7703	0.7750	9.1499	57.9156
x_3	8.4608	4.9096	8.7876	9.2120	10.1952	11.9213	8.6286	9.1855	0.7690	11.1372	8.5650
x_4	6.8178	6.7536	6.7504	10.6186	10.2179	9.5248	7.0000	8.5607	0.8210	10.0418	7.0000
x_5	6.9422	6.8460	6.7512	13.1083	11.7295	9.5914	7.0000	13.8828	0.7301	10.5524	7.0000
x_6	0.4895	0.4193	0.4264	8.3513	8.4150	11.5617	0.5000	11.3739	0.8075	10.0935	0.4602
x_7	0.6899	0.6890	0.6558	11.3542	9.8338	13.3258	0.7000	8.6359	0.7278	8.2930	0.6844
x_8	0.3315	0.3028	0.3404	12.9801	13.4398	8.8945	0.3000	8.6321	0.8013	12.8706	0.3122
x_9	0.0638	0.0782	0.0469	10.9416	13.9183	10.3114	0.0304	10.3144	0.7792	11.4687	0.0774
x_{10}	0.8077	0.7156	0.8320	9.8717	8.2924	10.0742	0.8500	8.0045	0.7837	11.2211	0.8313

Table A.1
Unimodal test benchmark functions considered in the experiments.

Name	Function	S	Minimum
f_1-Rotated hyper-ellipsoid	$f_1(\mathbf{x}) = \sum_{i=1}^{n}\sum_{j=1}^{i} x_j^2$	$[-65.5, 65.5]^n$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0, \ldots, 0)$
f_2-Schwefel 2	$f_2(\mathbf{x}) = \sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_i\right)^2$	$[-100, 100]^n$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0, \ldots, 0)$
f_3-Sphere	$f_3(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$	$[-5, 5]^n$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = 0, \ldots, 0$
f_4-Sum squares	$f_4(\mathbf{x}) = \sum_{i=1}^{n} i x_i^2$	$[-10, 10]^n$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0, \ldots, 0)$
f_5-Sum of different powers	$f_5(\mathbf{x}) = \sum_{i=1}^{n} \lvert x_i\rvert^{i+1}$	$[-1, 1]^n$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0, \ldots, 0)$

$$d_o = d_i + x_1 * 2 \tag{63}$$

There are 12 constraints proposed for this problem, including side constraints for upper and lower bounds of the variables.

$$g_1(X) = \frac{\phi}{2 * \sin^{-1}\left(\frac{x_1}{x_2}\right)} - x_3 + 1 < 0 \tag{64}$$

$$g_2(X) = 2 * x_1 - x_6 * (D - d) < 0 \tag{65}$$

$$g_3(X) = x_7 * (D - d) - 2 * x_1 < 0 \tag{66}$$

$$g_4(X) = x_2 - (0.5 - x_9) * (D + d) < 0 \tag{67}$$

$$g_5(X) = (0.5 + x_9) * (D + d) - x_2 < 0 \tag{68}$$

$$g_6(X) = \left(\frac{di - d}{2}\right) - \left(\frac{D - do}{2}\right) < 0 \tag{69}$$

$$g_7(X) = 0.5 * (D - x_2 - x_1) - x_8 * x_1 < 0 \tag{70}$$

$$g_8(X) = x_{10} * w - x_1 < 0 \tag{71}$$

$$g_9(X) = f_i < 0.515 \tag{72}$$

$$g_{10}(X) = f_o < 0.515 \tag{73}$$

Also, lower and upper bound constrains

$$g_{11}(X) = x_i < low_i \tag{74}$$

upper bound constrains.

$$g_{12}(X) = x_i > up_i \tag{75}$$

Where:

Decision variable	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
Upper bound	14	73	31	7	7	0.5	0.7	0.35	0.08	0.85
Lower bound	8	47	4	0	0	0.4	0.6	0.3	0.3	0.7

If the algorithm met any of the constraints it must be punished with a negative substantial fitness value, the value used was −1*E+300, and in the other case, the fitness formula is applied

$$maximize\, f(X) \begin{cases} fc * x_3^{\frac{2}{3}} * x_1^{1.8} & \vec{g}(X) \le 0 \;\&\; x_1 \le 25.4 \\ 3.647 * fc * x_3^{\frac{2}{3}} * x_1^{1.4} & \vec{g}(X) \le 0 \;\&\; x_1 > 25.4 \\ -1 * E + 300 & \vec{g}(X) > 0 \end{cases} \tag{76}$$

Table 16 shows the results from the rolling element bearing problem, in which SBLA gets the best AB and WST, even if MFO gets a better solution, the average quality of solutions of SBLA makes it a safe choice to solve the problem, it must be noted that the worst solution of MFO is −1.00E+300, this value shows that it was not possible to find a solution in that run and SBLA gets a solution in each of the 30 runs.

6. Conclusions

We coded an algorithm modeling the behavior of the side-blotched lizards (SBLA), and their particular rock–paper–scissors strategy (RPS). In this approach the population is divided into three subpopulations, which has their unique characteristics and weakness, blue gets overpowered by orange, orange is not capable of protecting against yellow, and yellow cannot do anything against blue, this makes the subpopulations to increase and constantly decrease, similar to having three sines displaced between them, and we modeled the population with this in mind, the blue lizards are monogamous and they protect each other from the other morphs, in the code they move towards the blue lizard that has the best fitness at the moment, in the other hand the orange morph has an aggressive and ultra-dominant, the coded movement makes them get far away from each other, and lastly the lizards with the yellow tint are opportunistic and they try to steal the other morphs females, in nature yellow lizards have success against orange ones, this is because the orange morphs

Table A.2
Multimodal test benchmark functions considered in the experiments.

Name	Function	S	Minimum
f_6-Ackley	$f_6(\mathbf{x}) = -20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)} + 20 + e$	$[-30, 30]^n$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0, \ldots, 0)$
f_7-Dixon	$f_7(\mathbf{x}) = (x_i - 1)^2 + \sum_{i=2}^{n} i(2x_i^2 - x_{i-1})^2$	$[-10, 10]^n$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = 2^{-\frac{2^i-2}{2^i}}$ for $i = 1, \ldots, n$
f_8-Griewank	$f_8(\mathbf{x}) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^n$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0, \ldots, 0)$
f_9-Infinity	$f_9(\mathbf{x}) = \sum_{i=1}^{n} x_i^6\left[\sin\left(\frac{1}{x_i}\right) + 2\right]$	$[-1, 1]^n$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0, \ldots, 0)$
f_{10}-Levy	$f_{10}(\mathbf{x}) = \sin^2(\pi\omega_1) + \sum_{i=1}^{n-1}(\omega_i - 1)^2 \times \left[1 + 10\sin^2(\pi\omega_i + 1)\right] + (\omega_d - 1)^2\left[1 + \sin^2(2\pi\omega_d)\right]$ $\omega_i = 1 + \frac{x_i - 1}{4}$	$[-10, 10]^n$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (1, \ldots, 1)$
f_{11}-Mishra11	$f_{11}(\mathbf{x}) = \left[\frac{1}{n}\sum_{i=1}^{n}\lvert x_i\rvert - \left(\prod_{i=1}^{n}\lvert x_i\rvert\right)^{\frac{1}{n}}\right]^2$	$[-10, 10]^n$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0, \ldots, 0)$
f_{12}-MultiModal	$f_{12}(\boldsymbol{x}) = \sum_{i=1}^{n}\lvert x_i\rvert\prod_{i=1}^{n}\lvert x_i\rvert$	$[-10, 10]^n$	$f(\boldsymbol{x}^*) = 0;$ $\boldsymbol{x}^* = (0, \ldots, 0)$
f_{13}-Plateau	$f_{13}(\boldsymbol{x}) = 30 + \sum_{i=1}^{n}\lfloor x_i\rfloor$	$[-5.12, 5.12]^n$	$f(\boldsymbol{x}^*) = 30;$ $\boldsymbol{x}^* = (0, \ldots, 0)$
f_{14}-Powell	$f_{14}(\mathbf{x}) = \sum_{i=1}^{n/4}\left[(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4\right]$	$[-4, 5]^n$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0, \ldots, 0)$
f_{15}-Qing	$f_{15}(\boldsymbol{x}) = \sum_{i=1}^{n}(x_i^2 - i)^2$	$[-500, 500]^n$	$f(\boldsymbol{x}^*) = 0;$ $\boldsymbol{x}^* = \left(\pm\sqrt{i}, \ldots, \pm\sqrt{i}\right)$
f_{16}-Quartic	$f_{16}(\mathbf{x}) = \sum_{i=1}^{n}\left\{(ix_i)^4 + rand[0, 1)\right\}$	$[-1.28, 1.28]^n$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0, \ldots, 0)$
f_{17}-Rastringin	$f_{17}(\mathbf{x}) = 10n + \sum_{i=1}^{n}\left[x_i^2 - 10\cos(2\pi x_i)\right]$	$[-5.12, 5.12]^n$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (0, \ldots, 0)$
f_{18}-Rosenbrock	$f_{18}(\mathbf{x}) = \sum_{i=1}^{n-1}\left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$	$[-5, 10]^n$	$f(\mathbf{x}^*) = 0;$ $\mathbf{x}^* = (1, \ldots, 1)$
f_{19}-Schwefel21	$f_{19}(\mathbf{x}) = \sum_{i=1}^{n}\lvert x_i\rvert$	$[-100, 100]^n$	$f(\boldsymbol{x}^*) = 0;$ $\boldsymbol{x}^* = (0, \ldots, 0)$
f_{20}-Schwefel22	$f_{20}(\mathbf{x}) = \sum_{i=1}^{n}\lvert x_i\rvert + \prod_{i=1}^{n}\lvert x_i\rvert$	$[-100, 100]^n$	$f(\boldsymbol{x}^*) = 0;$ $\mathbf{x}^* = (0, \ldots, 0)$
f_{21}-Step	$f_{21}(\boldsymbol{x}) = \sum_{i=1}^{n}(\lfloor x_i\rfloor + 0.5)^2$	$[-100, 100]^n$	$f(\boldsymbol{x}^*) = 0;$ $\boldsymbol{x}^* = (0.5, \ldots, 0.5)$
f_{22}-Trid	$f_{22}(\boldsymbol{x}) = \sum_{i=1}^{n}(x_i - 1)^2 - \sum_{i=2}^{n} x_i x_i - 1$	$[-n^2, n^2]^n$	$f(\boldsymbol{x}^*) = -n(n+4)(n-1)/6$ $\boldsymbol{x}^* = i(n + 1 - i)$ for i = 1,2 …n
f_{23}-Zakharov	$f_{23}(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{n} 0.5ix_i\right)^4$	$[-5, 10]^n$	$f(\boldsymbol{x}^*) = 0;$ $\boldsymbol{x}^* = (0, \ldots, 0)$

have bigger territories and cannot protect all their females, the movement programmed for the yellow lizard, is to move towards one orange territory selected via roulette according to their fitness. Besides movements of the lizard, their population can fluctuate, the add function serves to represent a newborn lizard, the delete function denotes a natural death, and the transform lizard function mimics a lizard being attacked by other.

We tested the performance of the proposed algorithm on a set of 5 unimodal, 18 multimodal and four composite benchmark functions; we compared their results to some well-established algorithms. The algorithms compared are the artificial bee colony (ABC), bat algorithm (BA), differential evolution (DE), firefly algorithm (FA), harmony search (HS), moth flame optimization (MFO),

Table A.3
Composite test benchmark functions considered in the experiments.

Name	Function	S	Minimum
f_{24}-Composite 1	$f_{24}(\mathbf{x}) = f_{17}(\mathbf{x}) + f_{20}(\mathbf{x}) + f_3(\mathbf{x})$	$[-100, 100]^n$	$f(\mathbf{x}^*) = 0$; $\mathbf{x}^* = (0, \dots, 0)$
f_{25}- Composite 2	$f_{25}(\mathbf{x}) = f_8(\mathbf{x}) + f_{17}(\mathbf{x}) + f_{18}(\mathbf{x})$	$[-100, 100]^n$	$f(\mathbf{x}^*) = n-1$; $\mathbf{x}^* = (0, \dots, 0)$
f_{26}- Composite 3	$f_{26}(\mathbf{x}) = f_6(\mathbf{x}) + f_{pnlty}(\mathbf{x}) + f_{18}(\mathbf{x}) + f_{20}(\mathbf{x})$	$[-100, 100]^n$	$f(\mathbf{x}^*) = (1.1n) - 1$; $\mathbf{x}^* = (0, \dots, 0)$
f_{27}- Composite 4	$f_{27}(\mathbf{x}) = f_6(\mathbf{x}) + f_8(\mathbf{x}) + f_{17}(\mathbf{x}) + f_{18}(\mathbf{x}) + f_{20}(\mathbf{x})$	$[-100, 100]^n$	$f(\mathbf{x}^*) = n-1$; $\mathbf{x}^* = (0, \dots, 0)$
f_{pnlty}—*Penalty2*	$f_{pnly}(\boldsymbol{x}) = 0.1\left\{ \begin{array}{c} \sin^2(3\pi x_1) \\ +\sum_{i=1}^{n-1}(x_i-1)^2\left[1+\sin^2(3\pi x_{i+1})\right] \\ +(x_n-1)^2\left[1+\sin^2(2\pi x_n)\right] \end{array} \right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$; $u(x_i, a, k, m) = \begin{cases} k(x_i-a)^m, & x_i > a \\ 0, & -a \le x_i \le a \\ k(-x_i-a)^m, & x_i < -a \end{cases}$	$[-50, 50]^n$	$f(\boldsymbol{x}^*) = 0$; $\boldsymbol{x}^* = (1, \dots, 1)$

simulated annealing (SA), crow search algorithm (CSA), particle swarm optimization (PSO) and grew wolf optimizer (GWO). The non-parametric Wilcoxon test and the Bonferroni correction tested these results, to prove that the stochastic nature of the algorithms not caused them. We tested and compared SBLA against the algorithms aforementioned, in solving real problems of engineering like; the welded beam, FM synthesizer, and rolling element bearing. The results validate that the proposed approach is precise, robust, and has a unique problem-solving strategy.

Future lines of work might include a multi-objective version of the algorithm. As discussed previously, the fluctuation produced by the changes on the subpopulations promote diversification and intensification at different stages. The effects of this mechanism could benefit the diversity of solutions on the objective-space of a multi-objective problem. Hyperheuristcs could also be benefited from the inclusion of the fluctuation as a criterion to select from different operators. Finally, as the proposed algorithm performed well, it can be applied to real-life problems to determine its performance on specific scenarios.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.asoc.2019.106039.

CRediT authorship contribution statement

Oscar Maciel C.: Investigation, Writing - original draft. **Erik Cuevas:** Formal analysis. **Mario A. Navarro:** Writing - original draft. **Daniel Zaldívar:** Validation. **Salvador Hinojosa:** Investigation.

Appendix

The sets of benchmark test functions that we implemented in the experiments are described in Tables A.1–A.3, representing the unimodal, multimodal and composite functions.

In the column of minimum we have $f(\mathbf{x}^*)$ which is the optimum value of the function and $\mathbf{x}^*$ are the optimum positions, the column **S** is the continuous search space, every function has n dimensions, in this case we performed the experiments using 30, 50 and 100 dimensions.

References

[1] F. Glover, G.A. Kochenberger, Handbook of Metaheuristics, Kluwer Academic Publishers, 2003.

[2] S. Nesmachnow, An overview of metaheuristics: accurate and efficient methods for optimisation, Int. J. Metaheuristics 3 (2014) 320, http://dx.doi.org/10.1504/IJMHEUR.2014.068914.

[3] M. Dorigo, C. Blum, Ant colony optimization theory: A survey, Theoret. Comput. Sci. 344 (2005) 243–278, http://dx.doi.org/10.1016/j.tcs.2005.05.020.

[4] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67, http://dx.doi.org/10.1016/j.advengsoft.2016.01.008.

[5] F. Fausto, A. Reyna-Orta, E. Cuevas, Á.G. Andrade, M. Perez-Cisneros, From ants to whales: metaheuristics for all tastes, Artif. Intell. Rev. (2019) 1–58, http://dx.doi.org/10.1007/s10462-018-09676-2.

[6] T. Bäck, F. Hoffmeister, H.-P. Schwefel, A survey of evolution strategies, in: Proc. Fourth Int. Conf. Genet. Algorithms, 1991, p. 8, https://bi.snu.ac.kr/Info/EC/A%20Survey%20of%20Evolution%20Strategies.pdf.

[7] M. Mitchell, Genetic algorithms: An overview, Complexity 1 (1995) 31–39, http://dx.doi.org/10.1002/cplx.6130010108.

[8] R. Storn, K. Price, Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (1997) 341–359, http://dx.doi.org/10.1023/A:1008202821328.

[9] K.S. Tang, K.F. Man, S. Kwong, Q. He, Genetic algorithms and their applications, IEEE Signal Process. Mag. 13 (1996) 22–37, http://dx.doi.org/10.1109/79.543973.

[10] H.-G. Beyer, H.-G. Beyer, H.-P. Schwefel, H.-P. Schwefel, Evolution strategies – A comprehensive introduction, Nat. Comput. 1 (2002) 3–52, http://dx.doi.org/10.1023/A:1015059928466.

[11] N. Hansen, He CMA Evolution Strategy: A Tutorial, 2016, http://arxiv.org/abs/160400772, (Accessed 18 September 2019).

[12] J. Zhang, A.C. Sanderson, JADE: Self-adaptive differential evolution with fast and reliable convergence performance, in: 2007 IEEE Congr. Evol. Comput. CEC 2007, 2007, pp. 2251–2258, http://dx.doi.org/10.1109/CEC.2007.4424751.

[13] R.A. Rutenbar, Simulated annealing algorithms: An overview, IEEE Circuits Devices Mag. 5 (1989) 19–26, http://dx.doi.org/10.1109/101.17235.

[14] Ş.I. Birbil, S.C. Fang, An electromagnetism-like mechanism for global optimization, J. Global Optim. 25 (2003) 263–282, http://dx.doi.org/10.1023/A:1022452626305.

[15] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: A gravitational search algorithm, Inf. Sci. (Ny) 179 (2009) 2232–2248, http://dx.doi.org/10.1016/j.ins.2009.03.004.

[16] E. Cuevas, A. Echavarría, M.A. Ramírez-Ortegón, An optimization algorithm inspired by the states of matter that improves the balance between exploration and exploitation, Appl. Intell. 40 (2014) 256–272, http://dx.doi.org/10.1007/s10489-013-0458-0.

[17] S. Mirjalili, SCA: A Sine Cosine Algorithm for solving optimization problems, Knowl. Based Syst. 96 (2016) 120–133, http://dx.doi.org/10.1016/j.knosys.2015.12.022.

[18] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (80) (1983) 671–680, http://dx.doi.org/10.1126/science.220.4598.671.

[19] R.A. Rutenbar, Simulated annealing algorithms: An overview, IEEE Circuits Devices Mag. 5 (1989) 19–26.

[20] N. Siddique, H. Adeli, Simulated annealing its variants and engineering applications, Int. J. Artif. Intell. Tools 25 (2016) http://dx.doi.org/10.1142/S0218213016300015.

[21] O.K. Erol, I. Eksin, A new optimization method: Big bang-big crunch, Adv. Eng. Softw. 37 (2006) 106–111, http://dx.doi.org/10.1016/j.advengsoft.2005.04.005.

[22] A. Valdivia-Gonzalez, D. Zaldívar, F. Fausto, O. Camarena, E. Cuevas, M. Perez-Cisneros, A states of matter search-based approach for solving the problem of intelligent power allocation in plug-in hybrid electric vehicles, Energies 10 (2017) http://dx.doi.org/10.3390/en10010092.

[23] H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi, Water cycle algorithm - A novel metaheuristic optimization method for solving constrained engineering optimization problems, Comput. Struct. 110–111 (2012) 151–166, http://dx.doi.org/10.1016/j.compstruc.2012.07.010.

[24] A. Kaveh, M. Khayatazad, A new meta-heuristic method: Ray optimization, Comput. Struct. 112–113 (2012) 283–294, http://dx.doi.org/10.1016/j.compstruc.2012.09.003.

[25] G.V.V. Loganathan, Z.W. Geem, J.H. Kim, G.V.V. Loganathan, Z.W. Geem, J.H. Kim, G.V.V. Loganathan, Z.W. Zong Woo Geem, J.H. Joong Hoon Kim, G.V.V. Loganathan, A new heuristic optimization algorithm: Harmony search, Simulation 76 (2001) 60–68, http://dx.doi.org/10.1177/003754970107600201.

[26] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition, in: 2007 IEEE Congr. Evol. Comput. CEC 2007, 2007, pp. 4661–4667, http://dx.doi.org/10.1109/CEC.2007.4425083.

[27] A. Husseinzadeh Kashan, League Championship Algorithm (LCA): An algorithm for global optimization inspired by sport championships, Appl. Soft Comput. J. 16 (2014) 171–200, http://dx.doi.org/10.1016/j.asoc.2013.12.005.

[28] S. Balochian, H. Baloochian, Social mimic optimization algorithm and engineering applications, Expert Syst. Appl. 134 (2019) 178–191, http://dx.doi.org/10.1016/j.eswa.2019.05.035.

[29] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems, CAD Comput. Aided Des. 43 (2011) 303–315, http://dx.doi.org/10.1016/j.cad.2010.12.015.

[30] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Neural Networks, 1995 Proceedings. IEEE Int. Conf., vol. 4, 1995, pp. 1942–1948, http://dx.doi.org/10.1109/ICNN.1995.488968.

[31] X.S. Yang, S. Deb, Cuckoo search via Lévy flights, in: 2009 World Congr. Nat. Biol. Inspired Comput. NABIC 2009 - Proc., 2009, pp. 210–214, http://dx.doi.org/10.1109/NABIC.2009.5393690.

[32] X.-S. Yang, Firefly algorithms for multimodal optimization, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 2009, pp. 169–178, http://dx.doi.org/10.1007/978-3-642-04944-6_14.

[33] X.-S. Yang, Firefly algorithm, Lévy flights and global optimization, in: Res. Dev. Intell. Syst. XXVI, Springer London, London, 2010, pp. 209–218, http://dx.doi.org/10.1007/978-1-84882-983-1_15.

[34] X.S. Yang, Firefly algorithms for multimodal optimization, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 2009, pp. 169–178, http://dx.doi.org/10.1007/978-3-642-04944-6_14.

[35] A. Banharnsakun, T. Achalakul, B. Sirinaovakul, The best-so-far selection in artificial bee colony algorithm, Appl. Soft Comput. J. (2011) 2888–2901, http://dx.doi.org/10.1016/j.asoc.2010.11.025.

[36] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61, http://dx.doi.org/10.1016/J.ADVENGSOFT.2013.12.007.

[37] A. Askarzadeh, A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm, Comput. Struct. 169 (2016) 1–12, http://dx.doi.org/10.1016/j.compstruc.2016.03.001.

[38] P. Díaz, M. Pérez-Cisneros, E. Cuevas, O. Avalos, J. Gálvez, S. Hinojosa, D. Zaldivar, An improved crow search algorithm applied to energy problems, Energies 11 (2018) 571, http://dx.doi.org/10.3390/en11030571.

[39] O. Camarena, E. Cuevas, M. Pérez-Cisneros, F. Fausto, A. González, A. Valdivia, Ls-II: An improved locust search algorithm for solving optimization problems, Math. Probl. Eng. 2018 (2018) http://dx.doi.org/10.1155/2018/4148975.

[40] E. Cuevas, J. Gálvez, O. Avalos, Parameter estimation for chaotic fractional systems by using the locust search algorithm, Comput. Sist. 21 (2017) 369–380, http://dx.doi.org/10.13053/CyS-21-2-2741.

[41] X.-S. Yang, A New Metaheuristic Bat-Inspired Algorithm, Springer, Berlin, Heidelberg, 2010, pp. 65–74, http://dx.doi.org/10.1007/978-3-642-12538-6_6.

[42] M. Dorigo, V. Maniezzo, A. Colorni, et al., Ant system: optimization by a colony of cooperating agents, IEEE Trans. Syst. Man Cybern. B 26 (1996) 29–41.

[43] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowl. Based Syst. 89 (2015) 228–249, http://dx.doi.org/10.1016/j.knosys.2015.07.006.

[44] D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) algorithm, Appl. Soft Comput. J. 8 (2008) 687–697, http://dx.doi.org/10.1016/j.asoc.2007.05.007.

[45] E. Cuevas, M. Cienfuegos, D. Zaldívar, M. Pérez-Cisneros, A swarm optimization algorithm inspired in the behavior of the social-spider, Expert Syst. Appl. 40 (2013) 6374–6384, http://dx.doi.org/10.1016/j.eswa.2013.05.041.

[46] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1997) 67–82, http://dx.doi.org/10.1109/4235.585893.

[47] F.B. Ozsoydan, A. Baykasoglu, Analysing the effects of various switching probability characteristics in flower pollination algorithm for solving unconstrained function minimization problems, Neural Comput. Appl. 31 (2019) 7805–7819, http://dx.doi.org/10.1007/s00521-018-3602-2.

[48] F.B. Ozsoydan, Effects of dominant wolves in grey wolf optimization algorithm, Appl. Soft Comput. J. 83 (2019) http://dx.doi.org/10.1016/j.asoc.2019.105658.

[49] F.B. Ozsoydan, Artificial search agents with cognitive intelligence for binary optimization problems, Comput. Ind. Eng. 136 (2019) 18–30, http://dx.doi.org/10.1016/j.cie.2019.07.007.

[50] F.B. Ozsoydan, A. Baykasoglu, A swarm intelligence-based algorithm for the set-union knapsack problem, Futur. Gener. Comput. Syst. 93 (2019) 560–569, http://dx.doi.org/10.1016/j.future.2018.08.002.

[51] F.B. Ozsoydan, A. Baykasoğlu, Quantum firefly swarms for multimodal dynamic optimization problems, Expert Syst. Appl. 115 (2019) 189–199, http://dx.doi.org/10.1016/j.eswa.2018.08.007.

[52] K. Sörensen, Metaheuristics-the metaphor exposed, Int. Trans. Oper. Res. 22 (2015) 3–18, http://dx.doi.org/10.1111/itor.12001.

[53] A. Hassan, N. Pillay, Hybrid metaheuristics: An automated approach, Expert Syst. Appl. 130 (2019) 132–144, http://dx.doi.org/10.1016/j.eswa.2019.04.027.

[54] K.C. Tan, S.C. Chiam, A.A. Mamun, C.K. Goh, Balancing exploration and exploitation with adaptive variation for evolutionary multi-objective optimization, European J. Oper. Res. 197 (2009) 701–713, https://www.sciencedirect.com/science/article/pii/S0377221708005894, (Accessed 22 February 2017).

[55] W. Ye, W. Feng, S. Fan, A novel multi-swarm particle swarm optimization with dynamic learning strategy, Appl. Soft Comput. 61 (2017) 832–843, http://dx.doi.org/10.1016/J.ASOC.2017.08.051.

[56] W.-D. Chang, A modified particle swarm optimization with multiple subpopulations for multimodal function optimization problems, Appl. Soft Comput. 33 (2015) 170–182, http://dx.doi.org/10.1016/J.ASOC.2015.04.002.

[57] Z. Liang, X. Wang, Q. Lin, F. Chen, J. Chen, Z. Ming, A novel multi-objective co-evolutionary algorithm based on decomposition approach, Appl. Soft Comput. 73 (2018) 50–66, http://dx.doi.org/10.1016/J.ASOC.2018.08.020.

[58] H. Zhang, M. Yuan, Y. Liang, Q. Liao, A novel particle swarm optimization based on prey–predator relationship, Appl. Soft Comput. 68 (2018) 202–218, http://dx.doi.org/10.1016/J.ASOC.2018.04.008.

[59] C.-F. Wang, W.-X. Song, A novel firefly algorithm based on gender difference and its convergence, Appl. Soft Comput. 80 (2019) 107–124, http://dx.doi.org/10.1016/J.ASOC.2019.03.010.

[60] S.H. Alonzo, B. Sinervo, Mate choice games context-dependent good genes and genetic cycles in the side-blotched lizard, Uta stansburiana, Behav. Ecol. Sociobiol. 49 (2001) 176–186, http://dx.doi.org/10.1007/s002650000265.

[61] P.J.M. van Laarhoven, E.H.L. Aarts, Simulated annealing, in: Simulated Annealing Theory Appl, Springer Netherlands, Dordrecht, 1987, pp. 7–15, http://dx.doi.org/10.1007/978-94-015-7744-1_2.

[62] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (2007) 459–471, http://dx.doi.org/10.1007/s10898-007-9149-x.

[63] X.S. Yang, Firefly. algorithm, Firefly algorithm stochastic test functions and design optimisation, Int. J. Bio-Inspired Comput. 2 (2010) 78, http://dx.doi.org/10.1504/IJBIC.2010.032124.

[64] A. Askarzadeh, A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm, Comput. Struct. 169 (2016) 1–12, http://dx.doi.org/10.1016/j.compstruc.2016.03.001.

[65] B. Sinervo, C.M. Lively, The rock–paper–scissors game and the evolution of alternative male strategies, Nature 380 (1996) 240–243, http://dx.doi.org/10.1038/380240a0.

[66] L.D. LaDage, T.C. Roth, B. Sinervo, V.V. Pravosudov, Environmental experiences influence cortical volume in territorial and nonterritorial side-blotched lizards, Uta stansburiana, Anim. Behav. 115 (2016) 11–18, http://dx.doi.org/10.1016/j.anbehav.2016.01.029.

[67] T. Mai, M. Mihail, I. Panageas, W. Ratcliff, V. Vazirani, P. Yunker, Cycles in zero-sum differential games and biological diversity, in: Proc. 2018 ACM Conf. Econ. Comput. - EC '18, ACM Press, New York, New York, USA, 2018, pp. 339–350, http://dx.doi.org/10.1145/3219166.3219227.

[68] S. Abbas, Golden ratio, Resonance 22 (2017) 51–60, http://dx.doi.org/10.1007/s12045-017-0432-y.

[69] J. Hančl, Sharpening of theorems of Vahlen and Hurwitz and approximation properties of the golden ratio, Arch. Der Math. 105 (2015) 129–137, http://dx.doi.org/10.1007/s00013-015-0788-8.

[70] M. Hassaballah, K. Murakami, S. Ido, Face detection evaluation: a new approach based on the golden ratio Φ, Signal Image Video Process. 7 (2013) 307–316, http://dx.doi.org/10.1007/s11760-011-0239-3.

[71] E. Unal, F. Dincer, E. Tetik, M. Karaaslan, M. Bakir, C. Sabah, Tunable perfect metamaterial absorber design using the golden ratio and energy harvesting and sensor applications, J. Mater. Sci. Mater. Electron. 26 (2015) 9735–9740, http://dx.doi.org/10.1007/s10854-015-3642-7.

[72] Y. Liang, C.-T. Li, Y. Guan, Y. Hu, Gait recognition based on the golden ratio, EURASIP J. Image Video Process. 2016 (2016) 22, http://dx.doi.org/10.1186/s13640-016-0126-5.

[73] E. Cuevas, D. Oliva, M. Diaz, J. Osuna, Optimización- Algoritmos Programados Con Matlab Optimization-Algorithms Programmed with Matlab, Alfaomega, Mexico, 2016.

[74] M. Hollander, D.A. Wolfe, E. Chicken, Wilcoxon, in: John Wiley & Sons (Ed.), Nonparametric Stat. Methods, second ed., Wiley, Florida, 2013, pp. 39–59.

[75] F. Marini, B. Walczak, Particle swarm optimization (PSO). A tutorial, Chemom, Intell. Lab. Syst. 149 (2015) 153–165, http://dx.doi.org/10.1016/j.chemolab.2015.08.020.

[76] N.A. Ab Aziz, M. Mubin, M.S. Mohamad, K. Ab Aziz, A synchronous-asynchronous particle swarm optimisation algorithm, Sci. World J. 2014 (2014) 17, http://dx.doi.org/10.1155/2014/123019.

[77] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Comput. Eng. Dep. Eng. Fac. Erciyes Univ. (2005).

[78] F. Luo, J. Zhao, Z.Y. Dong, A new metaheuristic algorithm for real-parameter optimization: Natural aggregation algorithm, in: 2016 IEEE Congr. Evol. Comput. CEC 2016, 2016, pp. 94–103, http://dx.doi.org/10.1109/CEC.2016.7743783.

[79] R.A. Armstrong, When to use the Bonferroni correction, Ophthalmic Physiol. Opt. 34 (2014) http://dx.doi.org/10.1111/opo.12131.

[80] B. Ramakrishnan, S.S. Rao, A general loss function based optimization procedure for robust design, Eng. Optim. 25 (1996) 255–276, http://dx.doi.org/10.1080/03052159608941266.

[81] S. Das, P.N. Suganthan, Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems, Jadavpur Univ. Nanyang Technol. Univ. Kolkata (2010) 341–359.

[82] B. Rajeswara Rao, R. Tiwari, Optimum design of rolling element bearings using genetic algorithms, Mech. Mach. Theory 42 (2007) 233–250, http://dx.doi.org/10.1016/j.mechmachtheory.2006.02.004.