Revista MATUA Páginas: 79–94

Facultad de Ciencias Básicas
©Programa de Matemáticas
Vol. V , № 2, (2018)

Benchmark Functions Optimization Using Binary Biogeography-Based Optimization with Aleatory-Mixed Migration (BBBO-AMM) and Binary Ant-Lion Optimizer (BALO)

José L. Gutiérrez., Sergio R. Rivera.
{jlgutierreza, srriverar}@unal.edu.co
Universidad Nacional de Colombia.
Department of Electrical and Electronics Engineering

***Abstract*— Problem statement:** Metaheuristic optimization algorithms have been taking more impulse in order to improve processes and solve complex problems that require a high computing capacity. These complex problems can have binary terms as variable decisions. There is steel a need for transforming the traditional heuristic algorithms in tools able to handle binary variables.
Current tools: In 2008, the biogeography-based optimization (BBO) algorithm was presented for the first time. This algorithm produced good results by using a model of species migration within ecosystems in order to find the optimal points of benchmark functions. Similarly, ALO, a new optimizer based on the hunting of ant-lion, was released in 2015. These algorithms can handle very well the benchmark functions when the variables are continuous.
Proposal: In this paper, we present a modification to both types of algorithms (BBO and ALO) that will improve the manner how the optimal points are found within the search space. The main modification to both algorithms allows solving problems, in their target functions, with binary decision variables.
Main contributions for each algorithm: An important modification to the first algorithm is how species migrate between ecosystems; this model is based on a modification to the proposal made in 2010. By adding two important features, migration processes are randomly chosen, and a new method for species migration is developed. The manner how species migrate thus becomes random between two migration models. The new proposal for the ALO (second algorithm) solves optimization problems through two different binary random models within the search space.
Validation: To evaluate the behavior of algorithms, fifteen benchmarking functions are used. In addition, a comparison with other optimization algorithms, such as the Binary Particle Swarm Optimization and Gravitational Search Algorithm (BPSOGSA), Genetic Algorithms (GA), and the Binary Bat Algorithm (BBA), is made. We also demonstrate the proposed algorithms for a real-world binary optimization problem.

Index Terms—Biogeography-Based Optimization, Ant-Lion Optimizer, Binary Algorithm, Binary Optimization.

Universidad del Atlántico
MATUA
Revista MATUA Páginas: 79–94
Facultad de Ciencias Básicas
©Programa de Matemáticas
Vol. V , Nº 2, (2018)

I. INTRODUCTION

Each year, metaheuristic optimization algorithms have been taking more impulse in order to improve processes and solve complex problems that require a high computing capacity. Many of these algorithms have been inspired by biological processes such as the Ant Colony Optimization (ACO) [1], Dolphin Echolocation [2], Grey Wolf Optimizer (GWO) [3], among others. There are also methods inspired by natural processes, such as the Gravitational Search Algorithm (GSA) [4], Ray Optimization [5], among others.

The advantage of these metaheuristic algorithms is their ease to solve different problems at high speed and with great accuracy; this makes them major computing models compared to conventional optimization techniques [6]. Despite their great computational capacity, it is evident that no heuristic algorithm is able to solve all current optimization problems, as proved by analyzing search algorithms developed so far [7] that is to say, solving certain problem does not allow a different one to be solved with the same technique.

The algorithms modified on this article, such as BBO [8] [9] and ALO [10], were developed based on the movement of groups of species between habitats and the movement of colonies of ant-lions in quest of food, respectively. Both types of algorithms usually operate within continuous search spaces, thus allowing for very short computation times; however, they sometimes can generate errors in search of a global minimum when finding a local minimum.

Despite optimization problems with discrete binary search, algorithms that allow moving between these spaces in a binary fashion must be developed. Nowadays, some algorithms have their binary version, such as Binary Harmony Search Algorithm (BHSA) [11] or Binary Magnetic Optimization Algorithm (BMOA) [12]

This article presents the binary version of BBO algorithm with modifications to the type of migration performed in order to find a more optimal search model. It also presents the binary version of ALO algorithm to work on search spaces ranging between "0" and "1" and vice versa. The rest of the article is organized as follows:

Section II presents an introduction to BBO algorithm with an explanation of its operation. Section III presents an introduction to ALO algorithm in its explanation. Section IV describes the operation of a binary algorithm. Subsections IV.A and IV.B explain the modifications to BBO algorithm and ALO algorithm, respectively. Section V presents the results of the proposed algorithms compared to those of other optimization algorithms, such as the Binary Particle Swarm Optimization (BPSO) algorithm and a Genetic Algorithm (GA), Binary Particle Swarm Optimization and Gravitational Search Algorithm (BPSOGSA), and the Binary Bat Algorithm (BBA). Finally, section VI presents the results of a real-world binary optimization problem.

II. BIOGEOGRAPHY-BASED OPTIMIZATION

Biogeography-based optimization is a model of heuristic optimization used for optimizing multidimensional functions, that is, a multivariable model that can be used in discontinuous functions.

The principle of this model is based on the biological distribution of species within a habitat. Assuming that certain habitats have a defined population, and considering that each habitat is isolated (Islands), species within the habitat will increase or decrease depending on habitat suitability. A habitat is considered isolated when conditions are not interchanged [8], [9].

Each habitat will have specific conditions that will allow certain geographical area to be home for each species; these conditions may comprise, for instance, rain, food, vegetation, among others. Each condition is known as Suitability Index Variable (SIV). All these features of each habitat, that is, the global of all variables within each geographical area, will indicate how convenient for each species is to reside within this area. Habitat Suitability Index (HSI) indicates how adequate each habitat is for its resident species [8], [9].

The population living within this habitat can be in two conditions: it may increase or decrease. In either case, the population will tend to migrate to other places (immigration) and the habitat will tend to receive species from other islands (emigration), depending on the number of species living within the habitat [8],

[9]. This interchange of species is beneficial to the habitat since it allows biological diversity to increase within the ecosystem. Furthermore, it allows the number of species to constantly vary until finding a balance, where the number of species emigrating corresponds to the number of species immigrating [8].

For the optimization algorithm, interchange of species is carried out considering a λ_i and μ_i parameter that determines species immigration and emigration, respectively, within the habitat H_i. Whenever interchange of species within certain area occurs, the area will be diversified and increase its HSI; with this, the emigration rate increases according to the number of residents within the area, whereas the immigration rate decreases due to species saturation. This indicates that the higher HSI is, the more static species distribution will be within the habitat [8], [9].

For the algorithm, a good solution can be considered in case of a high-HSI habitat, that is, the level of HSI indicates how good the solution is (Fitness). Similarly, a low level of HSI indicates an unfavorable solution. Likewise, it can be affirmed that the number of species in the solution is represented by HSI. A high HSI indicates that solutions are more likely to share characteristics with others, that is, there is exchange of information between habitats. By contrast, low-HSI habitats are more likely to receive information from other systems [9].

The BBO algorithm has 2 very important features:

A. *Migration*

Migration is the operator that indicates the likelihood of improving the conditions of the habitat H_i. This is used, mainly, as the probability that habitats share information between them. This migration depends on the immigration rate λ_i since it determines if the species will leave the habitat H_i. In case the species decides to immigrate, the target area will be selected based on its emigration rate μ_j [9].

$$\boldsymbol{H_j(SIV) \rightarrow H_i(SIV)} \qquad \textbf{(1)}$$

To determine whether the information will be shared between habitats, a probabilistic approach is performed. Given a probabilityP_{share}, a solution is selected to be modified. To do this, the immigration rate λ is used. λ decides if any parameter of the SIV will be modified, that is, it selects a habitat and determines if any condition of the system will be modified. If the parameter is to be modified, the emigration parameter μ of the other habitats determines the solution that will modify, in this case, traveling to the habitat where the research is conducted.

Although the migratory strategy of this model is similar to those used in other genetic algorithms, migration in BBO is used with current solutions; in addition, it is an adaptive process and not a reproductive one as in other type of algorithms.

B. *Mutation*

It is a probabilistic operator which randomly modifies the SIV of the habitat based on the number of species living there. This mutation is performed in order to increase population diversity within the system. If we consider a low-HSI habitat, then mutation allows solutions within the habitat to be improved; likewise, it operates with high-HSI systems, determining that solutions can be further improved.

Since HSI may vary with large-scale events directly affecting the number of species within the habitat and, therefore, affecting the balance within the system, it was decided that the algorithm could model these changes by mutating the SIV value [8], [9].

Each member of the population has an associated probability; this indicates how its existence is expected within the solutions to the problem. Being directly related to the HSI value, very high or very low values are considered improbable solutions; by contrast, the average HSI values are more likely to be selected. In case of high (or low) HSI values, the solution will tend to mutate into another solution value given by the following equation [8]:

$$\boldsymbol{m(S) = m_{max}\left(1 - {P_s}/{P_{max}}\right)} \quad \textbf{(2)}$$

where the value of m_{max} is a parameter determined by the user; this is done with the aim of diversifying the population and reducing the effect of selecting solutions with higher probability, since they will tend to be dominant by ignoring other solution values.

C. Overview of BBO

BBO algorithm works with different migration rates depending on several factors. Distance from the closest habitat, size habitat, climate, amount of food, and even human activity modify variation patterns of species migrations [9]. Therefore, there must be restrictions on emigration and immigration rates determined by:

$$\lambda_0 \geq \cdots \geq \lambda_k \geq \cdots \geq \lambda_n = 0$$

$$\boldsymbol{\mu_0 \leq \cdots \leq \mu_k \leq \cdots \leq \mu_n = 0} \qquad \textbf{(3)}$$

for each $k = 0,1,\cdots,n$ respectively, in order to model the behavior of species emigration and immigration between habitats. It is known that the higher the number of species within a habitat is, the emigration rate will tend to zero and the immigration rate will be higher. This does not occur when the habitat has few species, hence the restrictions shown in (3).

Migration curves are models which allow analyzing possible behaviors of the species within the habitat [13]. Various types of linear and non-linear models are employed, as shown in Figure 1.

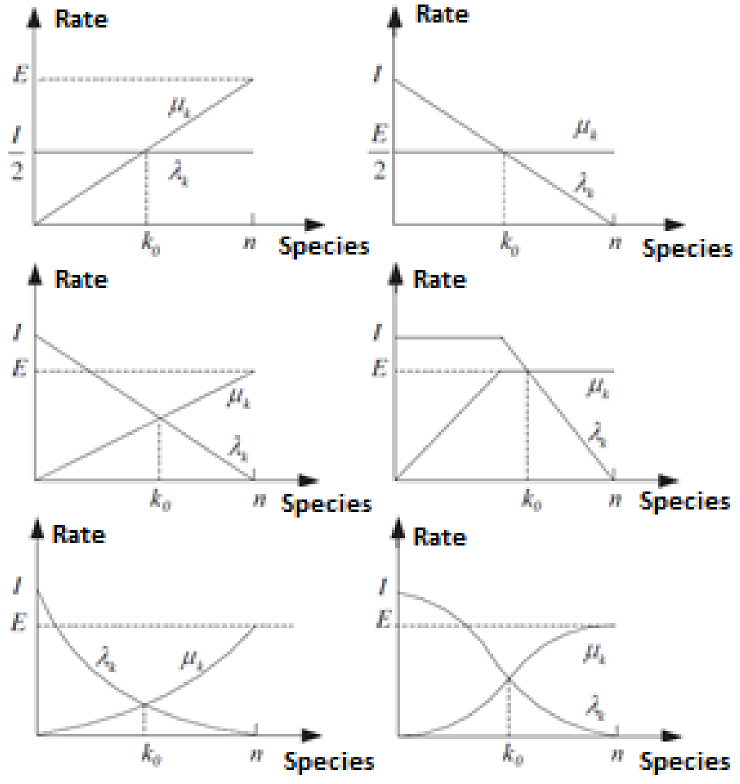

Figure 1. Examples of migration models (modification based on [9])

III. ANT-LION OPTIMIZER

This algorithm was called Ant-Lion Optimizer for an insect belonging to the family Myrmeleontidae, similar to a dragonfly [14]. The ant-lion hunts its prey by digging a cone-shaped pit where insects fall. If the prey tries to escape from the trap, the ant-lion will throw sand to slide it into its jaw and then consume it [14].

The behavior of these insects, regarding nourishment, is related to their degree of hunger and the moon. The hungrier they are, the deeper the holes will be and/or when the moon is full; this latter apparently occurs thanks to their internal lunar clock [15].

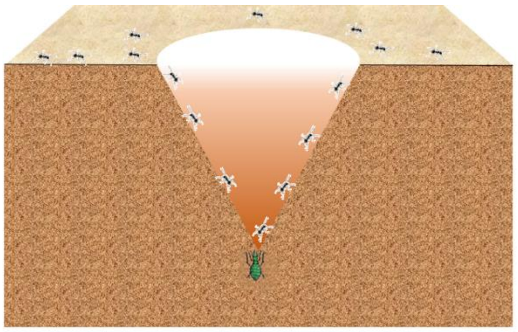
Figure 2. Cone-shaped pit dug by the ant-lion (modification based on [10])

According to this, the ALO algorithm performs a movement through the search space simulating the behavior of food and simulates the ant-lion's methods of hunting using traps. In this manner, the movement of food is modeled with stochastic patterns by adding a random pattern, as shown below [10]:

$$\boldsymbol{X(t) = [0, cumsum(2r(t_1) - 1), cumsum(2r(t_2) - 1), \cdots, cumsum\big(2r(t_3)\big)} \qquad \textbf{(4)}$$

where $cumsum$ calculates the cumulative sum, n is the maximum number of iterations, t indicates the iteration used, and $r(t)$ is the stochastic function defined as:

$$\boldsymbol{r(t) = \begin{cases} 1 & if\ \text{rand} > 0.5 \\ 0 & if\ \text{rand} \leq 0.5 \end{cases}} \qquad \textbf{(5)}$$

where t shows the iteration being calculated and $rand$ is a generator random number with uniform distribution in an [0,1] interval.

The behavior shown by the ants is very similar to that of the particles in the PSO algorithm. To solve the problems, each position of the ants is stored to then be evaluated in the function. If A_{ij} is the value of the $i-th$ dimension of the $j-th$ ant, the objective function will be evaluated as follows:

$$\boldsymbol{M_{OA} = \begin{bmatrix} f\big([A_{1,1}, A_{1,2}, \cdots, A_{1,d}]\big) \\ f\big([A_{2,1}, A_{2,2}, \cdots, A_{2,d}]\big) \\ \vdots \\ f\big([A_{n,1}, A_{n,2}, \cdots, A_{n,d}]\big) \end{bmatrix}} \quad \textbf{(6)}$$

where M_{OA} is the matrix of values, of each ant, obtained after being calculated from the function.

Additional to this modeling, the simulation of the ant-lions' location is performed; they are hidden anywhere in the search space. Positions are also stored similarly to those of ants; the same calculation of equation (6) is performed.

$$M_{OAL} = \begin{bmatrix} f([AL_{1,1}, AL_{1,2}, \cdots, AL_{1,d}]) \\ f([AL_{2,1}, AL_{2,2}, \cdots, AL_{2,d}]) \\ \vdots \\ f([AL_{n,1}, AL_{n,2}, \cdots, AL_{n,d}]) \end{bmatrix} \qquad (7)$$

As shown above for the case of ants, the matrix in (7) also shows values for the ant-lion in this case.

This algorithm has several characteristics explained below [14], [15]:

A. *Ants' Random Path:*

As explained at the beginning, an update on ants' position around the search space is made by limiting their movement to prevent them from crossing the boundaries of the search space.

B. *Trapped in the ant-lion's pit:*

In the random ant's movements, the ant will come to fall into the trap at some point due to its random movement when walking.

C. *Building the trap:*

To model the hunting capabilities of the ant-lion, the *"Roulette Wheel"* model is used. This model allows selecting the pit where the ant was trapped. This selection is made based on fitness results, thus choosing the trap with improved capabilities.

D. *Sliding ants towards the ant-lion:*

The ant-lion digs the pit according to its optimal value, in this case, simulating hunger and lunar phase. However, the ant-lion throws sand to its prey to attract it into the center of the hole when it is trapped. The simulation of this condition is performed by reducing the radius where the prey moves randomly.

E. *Trapping the prey and rebuilding the pit:*

When the prey is being drawn towards the ant-lion, the pit will be modified due to the prey's escape attempts; once the food reaches the bottom of the pit, the ant-lion devours it. To simulate this behavior, the ant-lion's position is updated with respect to the prey's position.

F. *Elitism:*

Apart from the above-mentioned characteristics, an elitist system is implemented to select and maintain the best solutions; information loss in successive iterations of the algorithm is thus not allowed. In this manner, each one of the best solutions is stored and considered an elite result; this allows affecting the random movement of food within the search space.

G. *ALO algorithm:*

After defining the characteristics of the code, the algorithm works under 3 functions of ordered lists which approximate the global optimum as a function:

$$\boldsymbol{ALO(A, B, C)} \qquad (8)$$

Where A is a function generating the initial values of the solutions, B manipulates the initial population predicted by the A function, and C, which is activated once the termination criterion is met. The A, B, C functions are defined as follows

$$\Phi \xrightarrow{A} \{M_{ant}, M_{OA}, M_{Antlion}, M_{OAL}\} \quad (9)$$

$$\{\boldsymbol{M_{ant}}, \boldsymbol{M_{Antlion}}\} \overset{b}{\rightarrow} \{\boldsymbol{M_{ant}}, \boldsymbol{M_{Antlion}}\}\textbf{(10)}$$
$$\{\boldsymbol{M_{ant}}, \boldsymbol{M_{Antlion}}\} \overset{c}{\rightarrow} \{\boldsymbol{true}, \boldsymbol{false}\}\textbf{(11)}$$

where M_{ant} is the matrix of ants' position, $M_{Antlion}$ includes the ant-lions' position, M_{OA} contains the optimal values of ants, and M_{OAL} has the optimal values of ant-lions.

IV. BINARY ALGORITHMS

When optimization algorithms, especially bio-inspired ones, seek the minimum of functions, they may seek around local minima, thus not ensuring high accuracy in results. The algorithm trying to find values outside its search range is also a problem, since it remains between two values and cannot converge to the desired value.

When the algorithm is used in a binary manner, these problems are avoided by working on a model that supports only values of 1 or 0. Since any other number can be represented in a binary fashion, variables are represented by using 15 bits. The overall dimensions of each function are given by [16]:

$$\boldsymbol{Dim_{habitat}} = \boldsymbol{Dim_{Function}} \times \mathbf{15} \quad \textbf{(12)}$$

where $Dim_{habitat}$ indicates the dimensions of each habitat within the system and $Dim_{function}$ are the dimensions of each function. According to (12), dimensions of the particles is 75 for functions of dimension 5, and 450 for functions of dimension 30.

To describe the binary development, it should be noted that, for each dimension of the functions, a 15-bit binary conversion is performed for each of them.

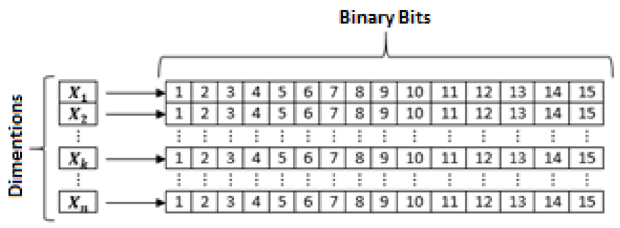

Figure 2. Binary conversion

After that, the first bit of each transformation is taken; these transformations will define their own sign: if "0", it is said to be negative, and if "1", it is said to be positive.

Table 1. Single-modal test functions

FUNCTION	DIM	RANGE	f_{min}
$f_1(x) = \sum_{i=1}^{n} x_i^2$	5 or 30	[0,1]	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	5 or 30	[0,1]	0
$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j\right)^2$	5 or 30	[0,1]	0
$f_4(x) = max_i\{\lvert x_i\rvert, 1 \le i \le n\}$	5 or 30	[0,1]	0
$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	5 or 30	[0,1]	0
$f_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	5 or 30	[0,1]	0
$f_7(x) = \sum_{i=1}^{n} ix_i^4 + random[0,1]$	5 or 30	[0,1]	0

Table 2. Multimodal test functions

FUNCTION	DIM	RANGE	f_{min}
$f_8(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	5 or 30	[0,1]	-418.9829 x Dim
$f_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	5 or 30	[0,1]	0
$f_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)\right) + 20 + e$	5 or 30	[0,1]	0
$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	5 or 30	[0,1]	0
$f_{12}(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1)\sum_{i=1}^{n}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1}) + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10{,}100{,}4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i > -a \end{cases}$	5 or 30	[0,1]	0
$f_{13}(x) = 0.1\left\{\sin^2(3\pi x_i) + \sum_{i=1}^{n}(x_i - 1)^2\{1 + \sin^2(3\pi x_i)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\right\} + \sum_{i=1}^{n} u(x_i, 5{,}100{,}4)$	5 or 30	[0,1]	0
$f_{14}(x) = -\sum_{i=1}^{n} \sin(x_i) \cdot \left(\sin\left(\frac{i x_i}{n}\right)\right)^{2m}, \qquad m = 10$	5 or 30	[0,1]	-4.687
$f_{15}(x) = \left\{\left[\sum_{i=1}^{n} \sin^2(x_i)\right] - \exp\left(-\sum_{i=1}^{n} x_i^2\right)\right\} \cdot \exp\left[-\sum_{i=1}^{n} sin^2\left(\sqrt{\lvert x_i \rvert}\right)\right]$	5 or 30	[0,1]	-1

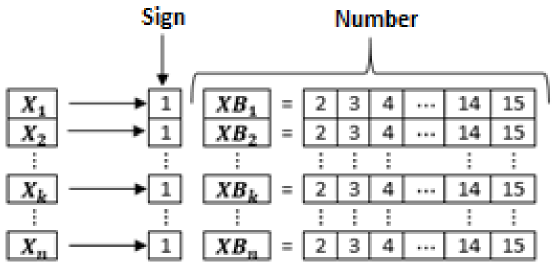

Figure 4. Sign selection

Each $\boldsymbol{XB_k}$ is taken and converted from binary to decimal with the respective sign; $\boldsymbol{n}$ has values that will be computed within the algorithm. As shown in Figures 3 and 4, working with a binary system provides more accuracy regarding the development of the algorithm, by sacrificing response times due to issues of large calculations to be performed aimed at finding the optimal solution.

A. *Binary Biogeography-based optimizer, proposed BBBO-AMM*

When working on BBO algorithm, the above-mentioned modifications were made from dimension to functions. Internally, the algorithm also had to be modified in order to obtain sufficient habitats according to the new binary dimensions.

Another modification to the code was a variant of migration models presented in Figure 1; a random selection between sinusoidal migration model and the out-of-phase sinusoidal model is performed, as shown in Figure 5.

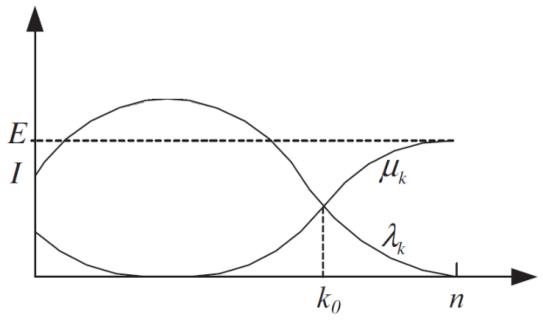

Figure 5. Simulated new migration model

B. *Ant-lion binary optimizer, proposed BALO*

For the ALO algorithm, the same procedure as that one performed for BBO is followed (see IV.A). The change to functions in the search range is made and the work values are expanded according to the dimensions of each function, as shown in Figure 3.

The difference between ALO and BBO is that ALO has no migration function; therefore, adding elements to the algorithm is not necessary.

V. TEST AND PERFORMANCE OF PROPOSED ALGORITHMS

To verify the two algorithms mentioned in IV.A and IV.B, 15 benchmark functions are used. These functions are separated into two different sets: single-modal functions and multimodal functions, as shown in Table 1 and Table 2, respectively. It is worth mentioning that the search range of functions, being binary, will only comprise 0 and 1. More accurate information about these functions can be found in [17], [18], [19], and more deeply studied in [20], [21], [22].

The results obtained by the algorithms can be compared with the initial versions of BBO and ALO algorithms [23]. A comparison using Binary Particle Swarm Optimization and Gravitational Search Algorithm, Genetic Algorithms, and Binary Bat Algorithm is also made, as shown in Table 3. Table 3 shows the results using a dimension 5 in the benchmarking functions. Additionally, it is presented the average (Ave) and standard deviation (Std) of 15 runs. As shown in Table 3, the new algorithms have obtained better results compared to the other algorithms. The results using a dimension of 30 in the benchmarking functions for the proposed algorithms are presented in Table 4. A comparison using Binary Particle Swarm Optimization and Gravitational Search Algorithm, and Binary Bat Algorithm is also made, as shown in Table 4. In this case, the new algorithms also have obtained better results compared to the other algorithms.

Table 3. Results of test functions (5 dimensions)

f	BPSOGSA	GA	BBA	BBBO-AMM	BALO
f1					
Ave	0.753882	10.070500	0.000000	**0.000000**	**0.000000**
Std	0.744054	24.944500	0.000000	**0.000000**	**0.000000**
f2					
Ave	0.158447	0.269483	0.000000	**0.000000**	**0.000000**
Std	0.121911	0.237880	0.000000	**0.000000**	**0.000000**
f3					
Ave	45.286676	555.903900	0.001676	**0.000000**	**0.000000**
Std	94.452227	250.693000	0.002835	**0.000000**	**0.000000**
f4					
Ave	2.464063	1.593750	0.006714	**0.000000**	**0.000000**
Std	2.429516	1.213480	0.004504	**0.000000**	**0.000000**
f5					
Ave	281.414962	369.754500	42.306340	**2.939208**	3.874140
Std	667.874313	342.889300	90.840110	**3.023899**	0.179688
f6					
Ave	8.093701	6.984222	0.001380	**0.000299**	0.370900
Std	17.670570	7.010388	0.002244	**0.000306**	0.112182
f7					
Ave	0.006397	0.047174	0.002724	0.002427	**0.000069**
Std	0.008876	0.043587	0.002066	0.000877	**0.000078**
f8					
Ave	-979.813200	-929.324000	-2094.678000	**-2094.678880**	-1747.220000
Std	25.037740	27.952310	0.097177	**0.097186**	271.203101
f9					
Ave	1.875194	2.189600	2.488134	0.878160	**0.000000**
Std	1.271683	0.833027	1.204948	1.023358	**0.000000**
f10					
Ave	0.541234	1.399853	0.000005	0.000005	**0.000000**
Std	0.800463	1.338105	0.000005	0.000005	**0.000000**
f11					

Ave	0.179551	0.706700	0.082874	0.040653	**0.000000**
Std	0.092974	0.322300	0.059624	0.018476	**0.000000**
f12					
Ave	0.370201	0.191197	-5.723056	**-6.115740**	0.063160
Std	0.485135	0.244347	0.838137	**0.027324**	0.056554
f13					
Ave	0.255321	0.193006	0.013475	**0.007289**	0.455900
Std	0.305777	0.254864	0.020621	**0.009035**	0.050934
f14					
Ave	-3.902076	-3.884920	-3.862056	**-4.002700**	-3.582546
Std	0.446362	0.717682	0.172965	**0.009319**	0.337560
f15					
Ave	0.000317	0.001575	0.000030	0.000006	**-1.000000**
Std	0.000718	0.001603	0.000022	0.000001	**0**

Table 4. Results of test functions (30 dimensions)

f	BPSOGSA	BBA	BBBO-AMM	BALO
f1				
Ave	3866.257983	730	**0**	**0**
Std	1437.537485	197	**0**	**0**
f2				
Ave	37.99127171	11	**0**	**0**
Std	9.396961534	2	**0**	**0**
f3				
Ave	27968.84947	14644	13100	**0**
Std	6803.046329	3993	4671	**0**
f4				
Ave	55.07736442	22	14	**0**
Std	8.466992652	3	11	**0**
f5				
Ave	6309639.467	60641.40230416030	377.14270326018	**29.00000000000**
Std	2564573.409	27486.05737483480	352.67830887436	**0.00000000000**
f6				
Ave	3401.603578	791.72800930458	**0.50764390821**	5.99930760648
Std	1543.648334	141.55525559901	**0.17091207924**	0.44254266122
f7				
Ave	2.640800742	0.30122779080	0.04152316862	**0.00007322754**
Std	2.42120684	0.10151718048	0.01473846813	**0.00006872469**
f8				
Ave	-7132.160017	-9011.78279072680	**-10801.74426217690**	-5309.21725024123
Std	864.9940034	741.94936489747	**346.29435209321**	481.35151993387
f9				
Ave	82478.30087	20888.63312964420	23.03969033795	**0.00000000000**
Std	17365.80727	5094.93849527133	7.25751926788	**0.00000000000**

f10				
Ave	11.0114596	7.28959839779	1.70637841518	**0.00000000000**
Std	1.096116914	0.90594927712	0.65279172258	**0.00000000000**
f11				
Ave	37.16827297	7.81628152526	0.17281571537	**0.00000000000**
Std	7.83326925	1.40165056162	0.15065236324	**0.00000000000**
f12				
Ave	1157.174018	120.26032397724	**-4.30002313485**	0.77827253380
Std	1436.354911	72.20438311236	**1.24099875980**	0.21143507960
f13				
Ave	1236.329722	83.36264260857	**0.69154785074**	2.99832753160
Std	762.8826081	21.86075906021	**0.15817651695**	0.00528880947
f14				
Ave	-12.41688963	-15.24559508689	-20.22678864141	**-8.05029392173**
Std	1.573980008	0.97544276027	1.23249843567	**1.21661039650**
f15				
Ave	7.00663E-12	0	0	**-1.000000**
Std	6.06052E-12	0	0	**0**

VI. STUDY IN A REAL CASE PROBLEM

To test the algorithms presented in this article, the study is based on the "Wind Farm Layout Optimization Competition", organized by *L'Institut de recherche en informatique de Toulouse* [24]. This competition is focused on reducing the cost of energy generated by a wind farm. This will help to verify the algorithms presented. This contest mainly aims to choose the best location of the turbines under land restrictions and avoid the turbulence effect.

To solve the problem, we must use the cost function equation:

$$\boldsymbol{fitness = \frac{(CC \cdot EOS) + YOP}{I} \cdot YPO + FSC} \quad (13)$$

where CC is the building cost given by (14), EOS is the factor that determines the economy of scale given by (15), YOP is the annual operation cost given by (16), I are the interests given by (17), YPO is the power supplied annually given by (18), and finally FSC is the coefficient of farm size given by (19).

$$\boldsymbol{CC = (c_t \cdot n) + \left(c_s \cdot floor\left(\frac{n}{m}\right)\right)} \quad (14)$$

$$\boldsymbol{EOS = \frac{2}{3} + \left(\frac{1}{3} \cdot e^{-0.00174 \cdot n^2}\right)} \quad (15)$$

$$\boldsymbol{YOP = C_{OM} \cdot n} \quad (16)$$

$$\boldsymbol{I = \frac{(1-(1+r)^{-y})}{r}} \quad (17)$$

$$YPO = \frac{1}{8760 \cdot P} \quad (18)$$

$$FSC = \frac{0.1}{n} \quad (19)$$

where

- $c_t = USD$ 750,000 is the cost of the turbine.
- $c_s = USD$ 8,000,000 is the cost of the substation.
- $m = 30$ are turbines per substation.
- $r = 0.03$ is the interest rate.
- $y = 20$ is the lifetime of the farm in years.
- $C_{OM} = 20{,}000$ is the annual operation of the turbine.
- n is the number of turbines.
- P is the power supplied by the wind farm.

The aforementioned parameters are those used in the 2015 contest [24]. This contest aimed to reduce generation costs. The optimization decision variables refer to the position and the number of wind turbines to be installed. Additionally, it is considered that there may be obstacles on the terrain, making the work area in a discontinuous area.

The results obtained by the algorithms worked in this research were compared with the Genetic Algorithm used in the contest. Table 5 shows the results:

Table 5. Results of Competition

f	ALO	BALO	GA	BBBO-MM
Ave	16.1542	**12.3989**	12.43737	**12.39472**
Std	0.0115	**0.0051**	0.0066	**0.03428**

These results show how new algorithms have performed better than others heuristic optimizations worked.

VII. CONCLUSIONS

In this article, two significant modifications to BBO and ALO algorithms have been made. On the one hand, a binary modification to both algorithms was made, thus allowing for a better approach to the optimal points of functions. On the other hand, a random change to the migration model of BBO algorithm was made. These two algorithms have been put to the test with 15 functions and compared with other 3 algorithms of great importance such as BPSOGSA, GA, and BBA. After being modified, they have proved that better results can be obtained and therefore perform a more realistic approach to natural behaviors.

In the case of the first algorithm, when modeling new migration models and adding randomness on them, we can expand the way how the algorithm finds optimal points of functions within the search space without sacrificing computation times.

In the case of binary modification to both algorithms, this allows finding, accurately, the optimal points of functions. This change is made considering that, by continuously varying the search within the space, this can "skip" optimal points; by contrast, by conducting the search within a discretized space in a binary fashion, the search is carried out accurately because it does not leave "holes" along its journey through the search space.

For future studies, it is recommended to apply both BBBO-AMM and BALO algorithms to real problems in order to determine the efficiency of these algorithms when solving real-world problems. It may be also interesting to study new migration models in BBBO-AMM and add different search models to BALO.

VIII. References

[1] M. Dorigo, V. Maniezzo and A. Colorni, «The Ant System: Optimization by a Colony of Cooperating Agents,» *IEEE Transactions Systems, Man, and Cybernetics B*, vol. 26, nº 1, pp. 26:29-41, 1996.

[2] A. Kaveh and N. Farhoudi, «A new optimization method: Dolphin echolocation,» *Advances in Engineering Software*, vol. 59, pp. 53-70, May 2016.

[3] S. Mirjalili, and A. Lewis, «Grey Wolf Optimizer,» *Advances in Engineering Software*, vol. 69, pp. 46-61, March 2014.

[4] E. Rashedi, H. Nezamabadi-pour and S. Saryazdi, «GSA: A Gravitational Search Algorithm,» *Information Sciences*, vol. 179, nº 13, pp. 2232-2248, June 2009.

[5] A. Kaveh and M. Khayatazad, «A new meta-heuristic method: Ray Optimization,» *Computer & Structures*, vol. 1, pp. 283-294, Dec 2012.

[6] A. Hernández Sauta, E. Torres Iglesias, M. A. Rodríguez Vidal and P. Eguía Lopez, «Survey and Crossed Comparison of Types, Optimal Location Techniques, and Power System Applications of FACTS,» *PowerTech (POWERTECH)*, Grenoble, Grenoble 2013, 2018 IEEE.

[7] D. Wolpert and W. Macready, «No Free Lunch Theorems for Optimization,» *IEEE Transactions on Evolutionary Computation*, nº 1, p. 67, 1997.

[8] D. Simon, «Biogeography-Based Optimization,» *IEEE Transactions on Evolutionary Computation 12*, pp. 702-713, 2008.

[9] H. Ma, «An analysis of the equilibrium of migration models for biogeography-based optimization,» *Information Sciences 180*, pp. 3444-3465, 2010.

[10] S. Mirjalili, «The Ant Lion Optimizer,» *Advances in Engineering Software*, vol. 83, pp. 80-98, March 2015.

[11] Z. W. Geem, J. H. Kim and G. V. Loganathan, «A New Heuristic Optimization Algorithm: Harmony Search,» *Simulation: Transactions of The Society for Modeling and Simulation International*, vol. 76, nº 2, pp. 60-68, Feb 2001.

[12] S. Mirjalili and S. Z. Mohd Hashim, «BMOA: Binary Magnetic Optimization Algorithm,» de *3rd International Conference on Machine Learning and Computing (ICMLC 2011)*, Singapore., 2011.

[13] R. H. MacArthur and E. O. Wilson, The Theory of Island Biogeography., Princeton, New Jersey: Princeton University Press, 1967.

[14] I. Scharf and O. Ovadia, «Factors Influencing Site Abandonment and Site Selection in a Sit-and-Wait Predator: A Review of Pit-Building Antlion Larvae,» *Journal of Insect Behavior*, vol. 19, nº 2, pp. 197-218, March 2016.

[15] J. Goodenough, B. McGuire y E. Jakob, Perspectives On Animal Behavior, John Wiley & Sons, 2009.

[16] S. Mirjalili, S. M. Mirjalili and X.-S. Yang, «Binary Bat Algorithm,» *Journal: Neural Computing and Applications*, vol. 25, nº 3, pp. 663-681, Sept. 2014.

[17] T. Back, Evolutionary Algorithms in Theory and Practice, Oxford, U.K.: Oxford Univ. Press, 2016.

[18] M. Iqbal, B. Xue, H. Al-Sahaf and M. Zhang, "Cross-Domain Reuse of Extracted Knowledge in Genetic Programming for Image Classification," in *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 4, pp. 569-587, Aug. 2017.

[19] Z. Cai and Y. Wang, «A Multiobjective Optimization-Based Evolutionary Algorithm for Constrained Optimization,» *IEEE Transactions. Evolutionary Computation.*, vol. 10, nº 6, pp. 658-675, Dec. 2016.

[20] X.-S. Yang, Engineering Optimization: An Introduction with Metaheuristic Applications, London: Wiley, Jul 2010.

[21] M. Molga and C. Smutnicki, «Test Functions for optimization needs.,» 2005. [On line]. Available: http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf.

[22] F. Wei, S. Li and J. Xue, "A New Local Searching Strategy for Global Optimization with a Large Number of Local Optimum," *2017 13th International Conference on Computational Intelligence and Security (CIS)*, Hong Kong, 2017, pp. 229-232.

Universidad del Atlántico

MATUA

Revista MATUA Páginas: 79–94

Facultad de Ciencias Básicas
©Programa de Matemáticas
Vol. V , N<u>o</u> 2, (2018)

[23] S. Mirjalili, G.-G. Wang y L. d. S. Coelho, «Binary Optimization Using Hybrid Particle Swarm Optimization and Gravitational Search Algorithm,» *Neural Computing and Applications*, vol. 25, nº 6, pp. 1423-1435, Nov. 2015.

[24] «Wind Farm Layout Optimization Competition.,» 2015. [on line]. Available: https://www.irit.fr/wind-competition/2015/#home.