KYBERNETIKA — VOLUME *54* (2018), NUMBER 3, PAGES 610–628

META-OPTIMIZATION OF BIO-INSPIRED ALGORITHMS FOR ANTENNA ARRAY DESIGN

Virgilio Zúñiga-Grajeda, Alberto Coronado-Mendoza and Kelly Joel Gurubel-Tun

In this article, a technique called Meta-Optimization is used to enhance the effectiveness of bio-inspired algorithms that solve antenna array synthesis problems. This technique consists on a second optimization layer that finds the best behavioral parameters for a given algorithm, which allows to achieve better results. Bio-inspired computational methods are useful to solve complex multidimensional problems such as the design of antenna arrays. However, their performance depends heavily on the initial parameters. In this paper, the distances between antenna array elements are calculated in order to reduce electromagnetic interference from undesired sources. The results are compared to previous works, showing an improvement on the performance of bio-inspired optimization algorithms such as Particle Swarm Optimization and Differential Evolution. These results are found to be statistically significant based on the Wilcoxon's rank sum test as compared to these methods using the standard parameters proposed in the literature. Furthermore, graphical representations of the Meta-Optimization process called meta-landscapes are presented, showing the behavior of these algorithms for a range of different parameters, providing the best parameter combinations for each antenna problem.

Keywords: bio-inspired algorithms, particle swarm optimization, differential evolution, meta-optimization, computer-aided design, antenna arrays

Classification: 65K10, 68U07, 68T20, 49M37, 78A50

1. INTRODUCTION

Nowadays, the use of wireless communication devices like cellular phones and global positioning systems has increased in such a way that the network bandwidth is affected. One way to tackle this problem is to design antenna architectures that meet the requirements of communication systems. In recent years, antenna designers have benefited from the use of computer systems and the application of numerical optimization techniques to explore a wide diversity of configurations before fabrication [1]. In order to improve the performance of an antenna, a set of individual antenna elements can be arranged in a geometrical configuration to create an antenna array; the combination of the radiated fields of every individual antenna element produces the overall radiation pattern

DOI: 10.14736/kyb-2018-3-0610

of the array. The interference amongst all radiation patterns depends on the particular geometry of the array (number of elements, distance between elements, etc.), where the radiated pattern of the elements should interfere constructively in the required direction and destructively in the remaining space. One goal in antenna array geometry synthesis is to define the arrangement of the array elements that creates a radiation pattern with the desired characteristics. A common requirement is the suppression of Side Lobe Levels (SLL), while preserving the main lobe and also reducing the effects of interference by placing nulls on the direction of undesired signals [12].

Bio-inspired computational methods like Particle Swarm Optimization (PSO) [18] have been used in the past to solve complex multidimensional problems in engineering [33] and electromagnetics [16, 34]. This algorithm has been found to be successful for antenna design, as shown in [26, 41] and even has outperform, in certain cases, other optimization methods [19]. Another widely used approach is Differential Evolution (DE), proposed by Price and Storn [39]. This technique is based on combining the position of individuals within a population and updating them if there is an improvement on their fitness. Differential Evolution has been used to design antenna arrays [4, 14], and several variations have been presented such as Ensemble Differential Evolution [35], and a multi-objective version in [23]. Not only in the radio frequency domain bio-inspired methods have been found useful: In [7], a Genetic Algorithm (GA) was applied to improve the geometry of optical antennas, since traditional RF design rules are not suitable for higher frequencies. The GA produced nonconventional geometries that outperform the classical dipole designs. Other bio-inspired techniques have been recently used, such as Firefly and Invasive Weed Optimization, as well as Wind Driven Optimization [22, 24, 25].

The performance of all these optimization algorithms to solve a given problem depends heavily on its initial parameters [17, 28]. To enhance the effectiveness of the algorithm, these parameters should be carefully selected according to the problem to be solved. In this paper, the parameters of algorithms Particle Swarm Optimization and Differential Evolution are selected using a technique called Meta-Optimization. This process consists of using another optimization algorithm to find good behavioral parameters. Meta-Optimization allows for an objective way to find the most suitable set of parameters for a given optimization method and problem to be solved. Different antenna synthesis problems proposed in the literature [20, 21], namely the optimization of distances between antenna elements, are tackled using Meta-Optimization techniques in order to improve the effectiveness of bio-inspired optimization algorithms. The results show an improvement on their capability to find better distance vectors that produce more desirable antenna array patterns.

2. META-OPTIMIZATION

Traditionally, the behavioral parameters have been chosen according to numerous experiments done by researchers. One example, presented in [36], shows the influence of the parameters maximum velocity and inertia weight on the performance of the PSO algorithm. A number of experiments were performed with different values for these parameters and it was concluded that when the maximum velocity is small, an inertia weight of approximately 1 is a good choice. Another example of parameter analysis is given in [8], where a constriction factor is proposed to limit the maximum velocity while using the inertia weight according to a given equation. In relation to the DE algorithm, Storn et al. [38] describe several variants of the algorithm and provide some general hints on their usage. These parameters can also be found by means of mathematical analysis as shown in [40] in which a selection of graphical guidelines is provided. This research shows how the speed of convergence impacts on the robustness of the solutions. Clerc and Kennedy [5] also show an analytical view of the particle's trajectory which leads to a generalized model of the algorithm and its convergence tendencies. These studies suggest that an automatic approach could be useful to find the best parameters for a given problem.

The selection of parameters can be divided into two cases: parameter control and parameter tuning [10]. In parameter control, the parameter values change during the optimization run. On the other hand, in parameter tuning the values do not change during the run but there is still a large number of combinations depending on the number of parameters (variables). Meta-Optimization, thus, is a kind of parameter tuning. Meta-Optimization consists of using one optimization technique to tune the parameters of another optimization technique. Meta-Optimization is also known as meta-evolution or automated parameter calibration. This concept was considered as early as 1978 by Mercer and Sampson [27], but their research was very limited due to the large computational costs. This suggests that, given the recent advances in computational power, the Meta-Optimization approach can now be an alternative.

As mentioned before, optimization techniques have a set of parameters that control their behavior. These parameters must be chosen carefully since they have a great effect on the output of the optimization method. It is worth mentioning that a given set of parameters could work well when optimizing a specific problem but perform differently when optimizing another. The way Meta-Optimization works is by using an optimization algorithm that has the parameters as output. During the Meta-Optimization process, every new set of parameters is used by the optimization algorithm and its output evaluated. Thus, the outer layer is in charge of finding a better set of parameters until a stop condition is met. Figure 1 illustrates the Meta-Optimization concept.

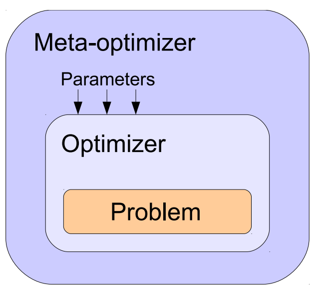


Fig. 1. Meta-Optimization. The parameters of the optimization algorithm are obtained by a second optimization layer.

For example, the PSO inertia weight determines how the previous velocity of a particle influences its velocity in the next iteration. If the inertia weight is low, the particle will change its velocity instantly and this effect would favor exploitation. On the other hand, if the inertia weight is high, the particle will hardly change its velocity and this will favor exploration. Therefore, by adjusting these parameters, a good balance between exploration and exploitation can be achieved. In the following section, the two bio-inspired optimization algorithms used in this work, namely Particle Swarm Optimization and Differential Evolution are explained. Likewise, the two techniques used to tune their parameters, Pattern Search and Local Unimodal Sampling are presented.

3. OVERVIEW OF THE ALGORITHMS

This section provides the reader with a general overview of the optimization algorithms used in this paper. Particle Swarm Optimization and Differential Evolution are used to optimize the radiation pattern of linear antenna arrays. The remaining algorithms, Pattern Search and Local Unimodal Sampling are used for the Meta-Optimization of those algorithms.

3.1. Particle swarm optimization

Particle Swarm Optimization (PSO) is an optimization technique developed by Eberhart and Kennedy [18]. It is inspired by the social behavior of various species. In these biological systems, simple individuals interact on their environment showing a collective behavior in order to fulfill their needs, for example, a swarm of bees searching for pollen. The bees try to find as many flowers as possible by randomly flying over the field. Each individual remembers the location where it found the most flowers, and it communicates this information to other individuals. Occasionally, a bee finds an area with more flowers than any other place found previously by the population in the swarm. Over time, more bees end up flying closer and closer to the best patch in the field, and soon, all the bees swarm around this point. The PSO algorithm begins by initializing a population of random solutions (or *particles*) and tries to find the best solution by updating generations of these particles. Each particle keeps track of their personal experience $\vec{p}$ and the overall experience $\vec{g}$, which are the best solution achieved by the particle and by the whole population respectively. The particles also have a position and a velocity associated to them, where the position represents a possible solution to the given problem and the velocity controls their movement to a new and better position. This velocity changes over the generations following the next two equations:

$$\vec{v}_{n+1} = \omega \cdot \vec{v}_n + c_1 r_1(\vec{p}_n - \vec{x}_n) + c_2 r_2(\vec{g}_n - \vec{x}_n) \tag{1}$$

$$\vec{x}_{n+1} = \vec{x}_n + \vec{v}_{n+1} \tag{2}$$

where $\vec{v}_n$ is the velocity of the particle and $\vec{x}_n$ its position, both at the nth generation. ω is called the inertia weight and controls the trade-off between the personal experience and overall experience of the group of particles and usually takes values in the range of [0,1]. c_1 and c_2 are acceleration constants and are usually taken as $c_1 = c_2 = 2.0$. r_1 and r_2 are random values uniformly distributed between 0 and 1. Each particle moves to its updated location when a new velocity has been obtained. The new position is determined according to Equation 2. The PSO algorithm will finish when a stopping criteria is met, usually when it reaches a maximum number of iterations.

3.2. Differential evolution

Differential Evolution (DE) is a population-based optimization technique introduced by Storn and Price in [39]. Being an Evolutionary Algorithm, DE uses crossover, mutation and selection operations to create a population of potential individuals that represent a possible solution to the given problem. On each iteration, new candidates are created by combining individuals from the existing population. These candidates are the result of the addition of a weighted difference between two individuals to a third member, this weight is a mutation factor called F. Then, a crossover operation is performed to make trial individuals using a crossover probability of CR. Candidates are replaced when a new individual has a better fitness value. There are many variants of the Differential Evolution algorithm, for example [13], where a novel hybrid version is shown to be effective for nonlinear optimization problems in high dimensional spaces.

3.3. Pattern search

Pattern Search (PS) is a method originally published by Hooke and Jeeves in [15], although an early version is attributed to Fermi and Metropolis [6]. Pattern Search is a kind of simple direct-search method since it can be used on functions that are not continuous. It samples the search-space from the current position and decreases its sampling-range when it fails to improve the fitness [29]. In this work, the Pattern Search algorithm was chosen for the outer layer of the Meta-Optimization process due to its simplicity. Since the Meta-Optimization approach requires a large number of iterations, PS helps to reduce the computational effort.

3.4. Local Unimodal sampling

The heuristical optimization technique Local Unimodal Sampling (LUS) can be consider an extension of the Pattern Search algorithm as it samples all dimensions simultaneously [29]. LUS adapts its sampling range during the optimization process and reduces its search-range through an exponential decrease. Since the number of possible solutions to a problem increases exponentially (a phenomenon known as the Curse of Dimensionality [3] random search algorithms can be improved by localizing the sampling around the best-known position [31], thus, the Local Unimodal Sampling is also a good choice for the Meta-Optimization process.

4. NUMERICAL RESULTS

In order to test the meta-optimized algorithms, a popular array synthesis technique proposed by Khodier in [21] is used. Khodier formulated a fitness function used by the PSO to obtain minimum Side Lobe Levels and null control by calculating the area under the curve of the desired array pattern. The distance between antenna elements of a linear antenna array was optimized and the results were compared with the QPM (Quadratic Programming Method) technique. This concept was later used in [32] where the synthesis was carried out for planar arrays and in [37] for circular arrays. In this paper, two experiments were performed and the resulting antenna patterns were compared with the ones obtained by Khodier in [20, 21]. In the next subsections, the Meta-Optimization

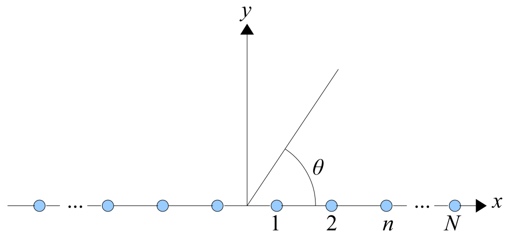


Fig. 2. Linear array of $2N$ elements positioned along the x-axis.

process, the output parameters and the resulting enhanced antenna pattern are presented.

4.1. First experiment

The problem to be solved is the second example in [21] which consists on optimizing the space between the elements of a 28-element linear array designed for SLL suppression in the region [0°, 180°] and prescribed nulls at 55°, 57.5°, 50°, 120°, 122.5° and 125°. To achieve this, the following expression is used as the fitness function [21]:

$$Fitness = \sum_i \frac{1}{\Delta\phi_i} \int_{\phi_{li}}^{\phi_{ui}} |AF(\phi)|^2 \,\mathrm{d}\phi + \sum_k |AF(\phi_k)|^2 \tag{3}$$

where $[\phi_{li}, \phi_{ui}]$ are the spatial regions in which the SLL are suppressed, in this case from 0° to 180°. $\Delta\phi_i = \phi_{ui} - \phi_{li}$, and ϕ_k are the directions of the nulls. The first term of the right-hand side of Equation 3 optimizes for suppressed Side Lobe Levels, while the second term is used to optimize for null control. AF is the antenna array factor and it represents an array of $2N$ infinitesimal dipoles positioned along the x-axis as shown in Figure 2. The array factor is independent of the antenna type assuming all of the elements are identical. In this work, isotropic radiators, which are ideal antennas that radiate an equal amount of power in all directions, are considered. Since the total field of the array is equal to the field of a single element positioned at the origin multiplied by the array factor [2], and the element pattern of identical elements is known, the main effort in the antenna array design is the synthesis of the array factor. Mathematically, AF is given by:

$$AF(\theta) = 2 \sum_{n=1}^{N} I_n \cos[kx_n \cos(\theta) + \beta_n] \tag{4}$$

where $2N$ is the number of antenna elements, θ is the angle of interfering or desired signal, x_n is the location of the nth element, k is the wavenumber, I_n is the amplitude weight at element n and β_n is the phase shift weight at element n. Since the aim is to obtain the optimized locations of the antenna elements, the amplitude and phase shift weights can be constant ($I_n = 1$ and $\beta_n = 0$) and the array factor is simplified as

$$AF(\theta) = 2 \sum_{n=1}^{N} \cos[kx_n \cos(\theta)]. \quad (5)$$

The Meta-Optimization algorithms were programmed in C♯ language using a modified version of *SwarmOps* which is a source-code library for numerical optimization problems written by Pedersen [30].

Table 1 shows the results of the Meta-Optimization process which was run to obtain the best parameters for the DE algorithm. The first 2 columns show the algorithms used as meta-optimizers as well as their parameters. PS needs no initial parameters and LUS uses $\gamma = 3$. Another factor in the Meta-Optimization process is the number of meta-runs which in this case is 5. The number of meta-iterations is defined according to the number of parameters to be optimized multiplied by 20. This will allow a fair number of meta-iterations depending on the number of variables. The fifth column in Table 1 is the number of optimization iterations performed in the Meta-Optimization phase. In this case, for each process three different experiments were run: using 200, 500 and 1000 iterations. This is to observe the variation in the quality of the results according to the number of iterations, since optimization algorithms tend to get closer and closer to the solution as the number of iterations increases. The next column is the resulting meta-fitness from every experiment and shows the value obtained at the end of the meta-iterations. The last three columns present the output which is the set of parameters suitable for the DE algorithm to best solve the antenna problem. The first is the number of particles NP, the second is the crossover probability CR and the third the differential weight F. The parameters for the PSO algorithm are obtained in the same fashion. The only difference is the number of meta-iterations, given that each algorithm has a different number of parameters. The results are shown in Table 2, where the last four columns are the best parameters found for the PSO algorithm: The number of particles S, the inertia weight ω, and constants c_1 and c_2.

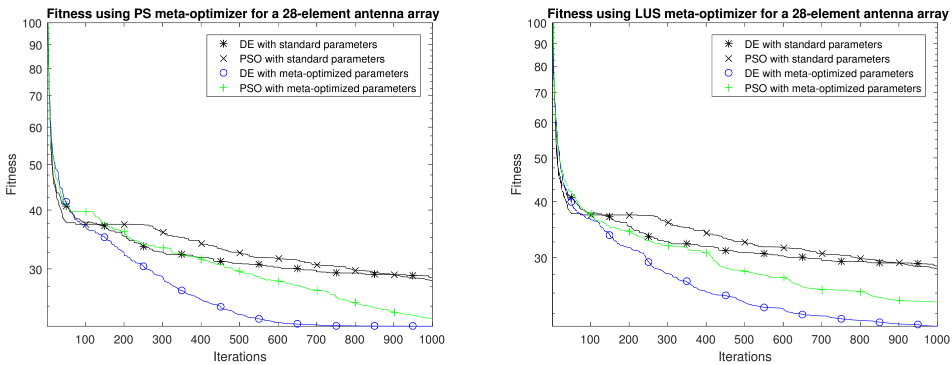
Meta-method	Meta-parameters	Meta-runs	Meta-iterations	Optimization iterations	Meta-fitness	Best found parameters		
						NP	CR	F
PS	N/A	5	60	200	96.88	3.0000	0.0124	0.2342
				500	82.14	84.1775	0.9688	0.0429
				1000	76.76	32.0036	0.9375	0.2968
LUS	$\gamma = 3$	5	60	200	97.25	23.1180	0.9927	0.0935
				500	82.74	34.8685	0.8697	0.1827
				1000	80.13	119.4043	0.6731	0.0172

Tab. 1. Meta-optimization of the Differential Evolution parameters using meta-methods PS and LUS with number of iterations 200, 500 and 1000. The problem being solved is a 28-element array with SLL suppression region of [0°, 180°] and prescribed nulls at 55°, 57.5°, 60°, 120°, 122.5° and 125°.

These set of parameters are used in the second phase of the optimization problem which is to run the algorithms to optimize the linear array described above. Another important factor in the Meta-Optimization process is the number of runs performed by the algorithm being meta-optimized. This is to obtain statistical significance and is set

Meta-method	Meta-parameters	Meta-runs	Meta-iterations	Optimization iterations	Meta-fitness	Best found parameters			
						S	ω	c_1	c_2
PS	N/A	5	80	200	114.81	300.0000	2.0000	-2.3792	6.0000
				500	111.79	300.0000	-1.0514	4.0000	1.4987
				1000	84.86	60.1695	-0.4515	-0.5656	1.4386
LUS	$\gamma = 3$	5	80	200	113.93	202.6578	-1.9376	2.4022	4.8989
				500	100.54	17.9624	-0.2811	-1.9813	1.9578
				1000	84.95	202.9297	0.0102	0.1499	1.6154

Tab. 2. Meta-optimization of the Particle Swarm Optimization parameters using meta-methods PS and LUS with number of iterations 200, 500 and 1000. The problem being solved is a 28-element array with SLL suppression region of [0°, 180°] and prescribed nulls at 55°, 57.5°, 60°, 120°, 122.5° and 125°.



(a) Pattern Search meta-optimized. (b) Local Unimodal Sampling meta-optimized.

Fig. 3. Fitness average of 50 runs using PS and LUS meta-optimizers for a 28-element linear antenna array.

to 50 runs. In other words, several runs are performed to minimize the chance that a better or worse result is obtained due only to the random nature of the algorithms. The average fitness scores are plotted in Figure 3, where it is shown that the DE and PSO meta-optimized parameters achieve a better convergence compared to the standard parameters.

At the end of the optimization process, the PSO and DE algorithms found the distances between elements needed to comply with the requirements of the 28-element linear antenna array: SLL suppression in the region [0°, 180°] and prescribed nulls at 55°, 57.5°, 50°, 120°, 122.5° and 125°. Each of the found values represent the distance of each antenna element to the center of the array with respect to λ and they are shown in Table 3. Once the distances for every antenna element are found, it is possible to build a full model of the resulting antenna array using Equation 5. The array factor AF is calculated by the sum of all the radiation patterns of the 28 antenna elements. Each value of x_n is taken from Table 3 and the magnitude of the array factor for every transmitting direction θ is obtained. Figure 4 and Figure 5 show a rectangular 2-D pattern in dB of the normalized AF, which is a graphical representation of the antenna array behavior. Both figures show the results for the PS and LUS algorithms respectively. In Figure 4, the "PSO" legend corresponds to the results obtained by Khodier in [21], and the legends "meta-PSO" and "meta-DE" are the results found in this work. It can be observed that the meta-optimized algorithms (particularly PSO) show lower Side Lobe Levels, especially at the outer angles, where the PSO technique achieves around -35dB. They also keep a narrow main lobe, as required by the suppression region [0°, 180°] and the prescribed nulls are also obtained. Similar results can be seen in Figure 5, where the LUS algorithm was used.

PS		LUS	
DE	PSO	DE	PSO
−14.0000	−13.9999	−14.0000	−13.9958
−12.7162	−12.8550	−12.6238	−12.7723
−11.7458	−12.2137	−11.5534	−11.9684
−10.8999	−11.3429	−10.5314	−11.0294
−9.9774	−10.7613	−9.4410	−10.2245
−8.9429	−9.7964	−8.6096	−8.8950
−8.0736	−8.7276	−7.7340	−7.7529
−6.8577	−7.6411	−6.6975	−6.5539
−5.4952	−6.4483	−5.5469	−5.3588
−4.5287	−5.2548	−4.5823	−4.6704
−3.4568	−4.2565	−3.5783	−3.7413
−2.4958	−3.0349	−2.6328	−2.8647
−1.5006	−1.7781	−1.6174	−1.8965
−0.5102	−0.6313	−0.5643	−0.7349
0.5102	0.6313	0.5643	0.7349
1.5006	1.7781	1.6174	1.8965
2.4958	3.0349	2.6328	2.8647
3.4568	4.2565	3.5783	3.7413
4.5287	5.2548	4.5823	4.6704
5.4952	6.4483	5.5469	5.3588
6.8577	7.6411	6.6975	6.5539
8.0736	8.7276	7.7340	7.7529
8.9429	9.7964	8.6096	8.8950
9.9774	10.7613	9.4410	10.2245
10.8999	11.3429	10.5314	11.0294
11.7458	12.2137	11.5534	11.9684
12.7162	12.8550	12.6238	12.7723
14.0000	13.9999	14.0000	13.9958

Tab. 3. 28-element linear antenna array locations obtained with the DE and PSO algorithms using the meta-optimized parameters calculated by the PS and LUS meta-optimizers. Each value corresponds to the distance of each element to the center of the array with respect to λ.

4.2. Second experiment

A second experiment was performed to compare the results obtained by the Cuckoo Search (CS) and Comprehensive Learning PSO (CLPSO) algorithms. These techniques were recently introduced to the electromagnetics and antenna community by Khodier in [20]. This experiment was conducted following the same procedure as the previous one. The problem consists of a 32-element array with SLL suppression region of [0°, 87°] and [93°, 180°] and prescribed nulls at 81° and 99°. The same fitness function from the previous experiment was used. The results of the Meta-Optimization process using DE and PSO are shown in Tables 5 and 6 respectively. Table 4.1 shows the resulting distances between antenna elements and the array center. The average fitness scores are plotted in Figure 6, showing the improved performance due to the meta-optimized DE

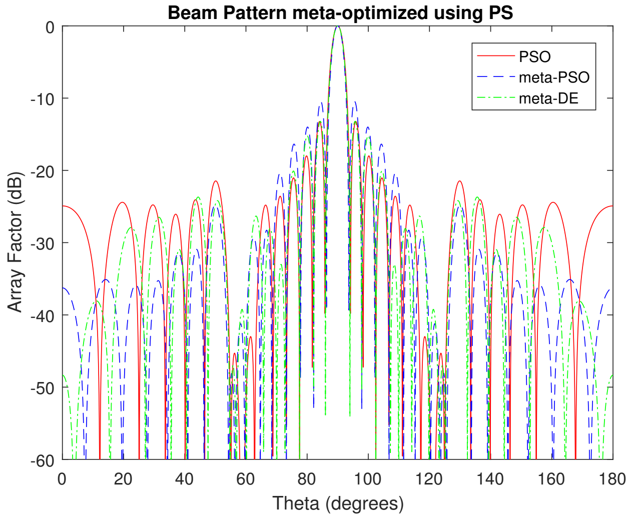


Fig. 4. Lower Side Lobe Levels obtained with PS meta-optimized parameters compared with Khodier's results [21]. A 28-element array with SLL suppression region of [0°, 180°] and prescribed nulls at 55°, 57.5°, 60°, 120°, 122.5° and 125°.

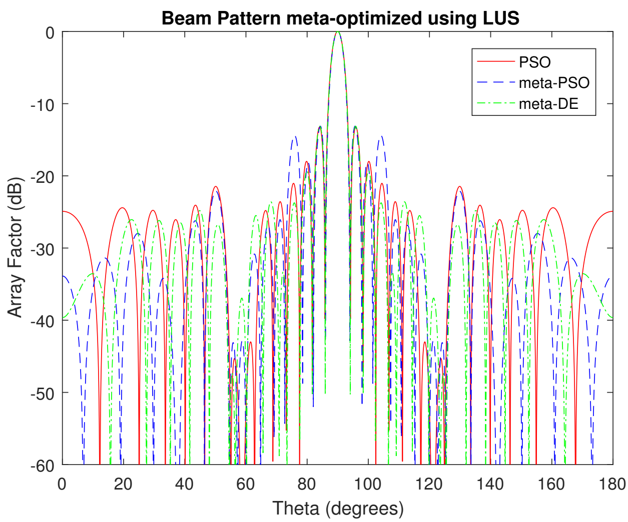
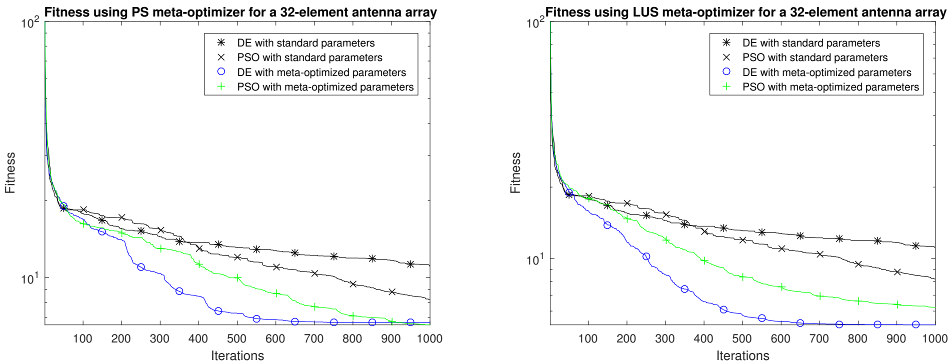


Fig. 5. Lower Side Lobe Levels obtained with LUS meta-optimized parameters compared with Khodier's results [21]. A 28-element array with SLL suppression region of [0°, 180°] and prescribed nulls at 55°, 57.5°, 60°, 120°, 122.5° and 125°.



(a) Pattern Search meta-optimized. (b) Local Unimodal Sampling meta-optimized.

Fig. 6. Fitness average of 50 runs using PS and LUS meta-optimizers for a 32-element linear antenna array.

PS	LUS
−18.67	−19.80
−16.44	−18.50
−14.14	−14.79
−11.62	−12.26
−8.96	−10.95
−7.50	−10.42
−6.46	−9.07
−4.92	−8.49
−4.25	−6.98
−3.00	−6.97
−2.70	−5.47
−0.78	−5.10
0.04	−4.04
1.14	−3.83
1.62	−3.49
2.01	−2.57
3.36	−2.36
4.36	−1.44
5.66	−0.55
6.01	0.38
7.33	1.60
8.33	5.70
9.34	7.58
10.22	8.96
10.49	10.07
11.48	11.58
12.78	12.91
12.98	13.56
14.62	14.72
15.61	16.15
17.47	16.41
20.00	17.65

Tab. 4. 32-element linear antenna array locations obtained with the PSO algorithm using the meta-optimized parameters calculated by the PS and LUS meta-optimizers. Each value corresponds to the distance of each element to the center of the array with respect to λ.

Meta-method	Meta-parameters	Meta-runs	Meta-iterations	Optimization iterations	Meta-fitness	Best found parameters		
						NP	CR	F
PS	N/A	5	60	200	30.80	22.9343	1.0000	0.3670
				500	11.84	194.0127	0.8570	0.0098
				1000	8.06	103.3261	1.0000	0.1710
LUS	$\gamma = 3$	5	60	200	18.70	11.9133	0.5375	0.0342
				500	11.45	84.9178	0.8226	0.0156
				1000	8.14	61.6216	0.9196	0.1955

Tab. 5. Meta-optimization of the Differential Evolution parameters using meta-methods PS and LUS with number of iterations 200, 500 and 1000. The problem being solved is a 32-element array with SLL suppression region of [0°, 87°] and [93°, 180°] and prescribed nulls at 81° and 99°.

Meta-method	Meta-parameters	Meta-runs	Meta-iterations	Optimization iterations	Meta-fitness	Best found parameters			
						S	ω	c_1	c_2
PS	N/A	5	80	200	36.95	52.6579	0.0000	4.0000	2.4090
				500	35.49	282.1608	-0.0352	-1.4511	5.6875
				1000	12.65	126.2368	-0.0938	0.4913	1.5450
LUS	$\gamma = 3$	5	80	200	42.21	265.5426	-0.3580	3.3055	1.9896
				500	24.93	144.7197	0.0052	2.4909	1.5117
				1000	12.23	74.8707	0.1816	0.2119	1.5254

Tab. 6. Meta-optimization of the Particle Swarm Optimization parameters using meta-methods PS and LUS with number of iterations 200, 500 and 1000. The problem being solved is a 32-element array with SLL suppression region of [0°, 87°] and [93°, 180°] and prescribed nulls at 81° and 99°.

and PSO parameters. The radiation pattern results obtained by the PS and LUS Meta-Optimization parameters are shown in Figure 7 and Figure 8 respectively. In Figure 7, the "CS" and "CLSPSO" legends correspond to the results by [20], the Cuckoo Search and the Comprehensive Learning PSO algorithms. The legend "meta-PSO" is the result from this work. It can be seen that the meta-optimized PSO achieves lower Side Lobe Levels. It also keeps a desired main lobe and the prescribed nulls are placed at the angles 81° and 99°, reaching as low as -70dB, surpassing CS and CLSPSO. The results presented in Figure 8 show an even better pattern from the meta-optimized PSO.

4.3. Statistical tests

In order to statistically analyze the results in previous experiments, a non-parametric significance procedure known as the Wilcoxon's rank test for independent samples has been conducted [11]. Such proof aims to detect significant differences between the behavior of two algorithms. As per null hypothesis, it is assumed that there is no significant difference between accuracy values in both optimized and meta-optimized approaches. However, the alternative hypothesis considers a significant difference between accuracy values in both methods. In this case, the significance level is 5%, which means that the confidence level is 95%, a strong evidence that the second hypothesis is true, thus the optimized and meta-optimized techniques are significantly different and therefore their results did not occurred by coincidence, i. e., due to their random initialization nature. These statistical results for both 28-array and 32-array experiments are shown in Ta-

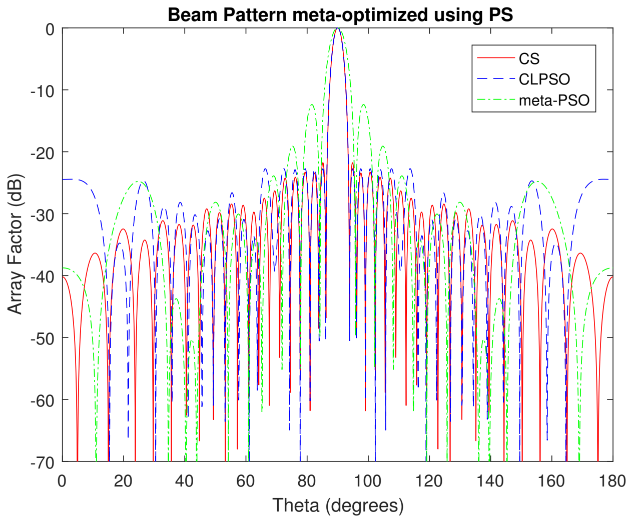


Fig. 7. Lower Side Lobe Levels obtained with PS meta-optimized parameters compared with Khodier's results [20]. A 32-element array with SLL suppression region of [0°, 87°] and [93°, 180°] and prescribed nulls at 81° and 99°.

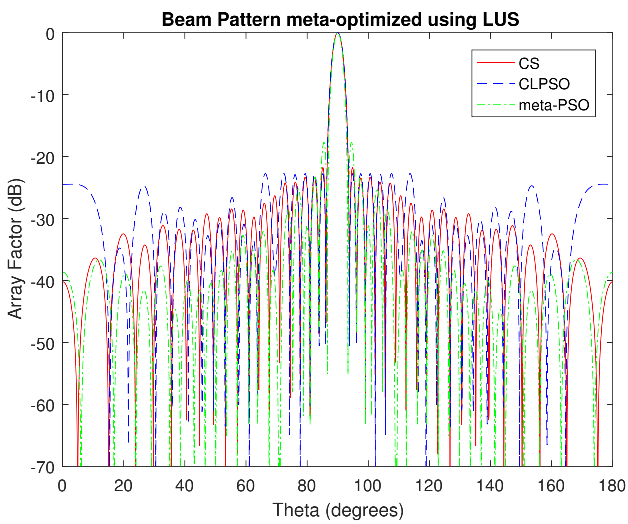


Fig. 8. Lower Side Lobe Levels obtained with LUS meta-optimized parameters compared with Khodier's results [20]. A 32-element array with SLL suppression region of [0°, 87°] and [93°, 180°] and prescribed nulls at 81° and 99°.

	Standard		Meta-optimized			
			PS		LUS	
	DE	PSO	DE	PSO	DE	PSO
28-array						
Best	20.85	22.03	16.07	18.61	17.25	19.13
Mean	28.73	28.33	22.66	23.39	21.00	23.83
Worst	38.40	37.39	32.13	32.07	24.87	29.90
SD	3.22	3.38	3.44	3.10	1.80	2.64
***p*-value**	N/A	N/A	**8.35E-11**	**1.73E-15**	**3.17E-12**	**2.21E-10**
Runtime	12.05	12.04	11.27	13.53	13.69	10.98
32-array						
Best	6.63	3.43	2.12	2.73	2.81	1.83
Mean	11.08	8.18	6.68	6.54	5.25	6.20
Worst	15.60	12.67	10.83	14.79	9.00	10.44
SD	1.97	2.34	1.83	2.22	1.50	1.97
***p*-value**	N/A	N/A	**1.59E-15**	**2.61E-17**	**1.72E-09**	**8.69E-04**
Runtime	13.60	13.29	15.27	13.90	17.83	13.96

Tab. 7. Statistical results using standard and meta-optimized parameters, including Wilcoxon's p-values.

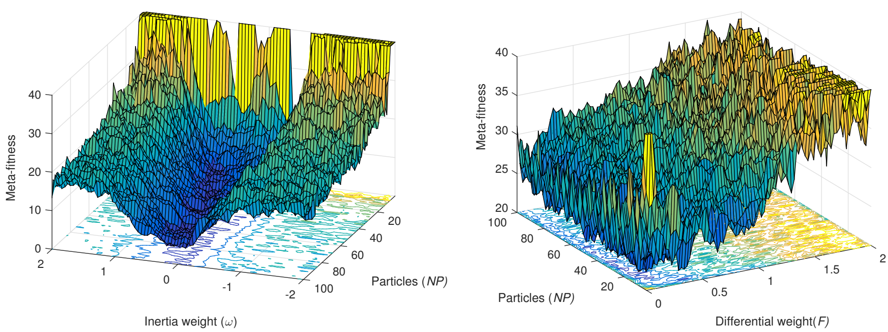
ble 7, together with the best, mean, worst and standard deviation fitness values from 50 experiments. It can be observed that the p-values obtained by the Wilcoxon test for each meta-optimized method are much smaller than 5% (or 0.05), which is evidence of a significant difference between methods.

4.4. Meta-landscapes

A third experiment was conducted, this time to study the Meta-Optimization process itself. It consists on the graphical representation of a set of resulting parameters that allows to observe the behavior of the procedure. This representation is called meta-landscape.

As mentioned before, in optimization problems, researchers often have to deal with the complexity of having an exponential increase of dimensions in a problem space. In Meta-Optimization, the number of parameters to be optimized is multiplied by the meta-optimizer's own parameters. For each new parameter, the number of candidate solutions grows exponentially. For this reason, it is desirable that the meta-optimizer algorithm is as simple as possible. This means that the problem to be solved by the meta-optimizer should not be so complex as to have the algorithm fail to converge. One way to observe the kind of problem the meta-optimizer has to deal with, is to generate a 3-dimensional plot of the meta-fitness landscape. Most of the methods have more than two parameters so it is necessary to fix some of them to be able to produce a viewable plot.

Consider the PSO algorithm with c_1 and c_2 fixed to 1.49445, which is considered the standard value [9]. The variable parameters being the number of particles NP and the inertia weight ω. The boundaries for these parameters are [1, 100] for NP and [-2, 2] for ω. Figure 9(a) shows the meta-fitness landscape of the PSO when optimizing the 28-element linear antenna array of the first experiment. The algorithm is executed 5 times and runs for 1000 iterations. These results suggest that the problem of meta-optimizing the PSO algorithm to solve the antenna array is simple. The meta-landscape surface is fairly regular and without obvious local minima. The graph shows that the meta-fitness values are worst when only a few number of particles are used. Another finding is that the best results are achieved whenever the inertia weight values are close to 0, regardless of their sign. It can be also observed that once a certain minimum number of particles NP is used (around 20), the meta-optimizer is capable of finding good results without having higher number of particles. This finding could be an advantage in the Meta-Optimization process, since the processing time of the algorithm can be reduced by using less particles. Something similar can be observed in Figure 9(b) where the meta-landscape for the DE algorithm was obtained with the same problem. In this case, the variables are the number of particles NP and the differential weight F. The third parameter CR is fixed to 0.5313. It can also be noticed that the meta-landscape is fairly simple with a single minimum. Moreover, the number of particles NP seems to be unimportant, as long as the value of the differential weight F is kept below 0.5. This is a useful discovery, as it allows to use less of particles, reducing the processing effort. Both meta-landscapes show a valley of good performing parameter combinations. It is worth mentioning that in these experiments some parameters were fixed, so it is possible that using all of them the meta-landscape will look different.



(a) Varying NP and ω. c_1 and $c_2 = 1.49445$.

(b) Varying NP and F. $CR = 0.5313$.

Fig. 9. Meta-landscapes for PSO (a) and DE (b) obtained by varying two dimensions. For 5 runs and 1000 iterations. For a 28-element linear antenna array.

5. CONCLUSION

In this paper, a technique called Meta-Optimization was used to improve the performance of bio-inspired optimization algorithms that solve antenna array synthesis problems. Meta-Optimization consists of employing a second optimization algorithm to find good behavioral parameters for a given technique. The parameters for Particle Swarm Optimization and Differential Evolution were calculated using two different meta-optimizers: Pattern Search and Local Unimodal Sampling. The results show an improvement on the antenna array radiation pattern as compared to previous works. The Meta-Optimization approach reduced the Side Lobe Levels while keeping nulls at certain directions. It has also been found that the improvement obtained by this technique is statistically significant based on Wilcoxon's rank sum test as compared to previous methods. Furthermore, the meta-landscapes for both algorithms were examined and it was observed that for this particular problem the number of particles did not have a considerable impact on the results. It was also found that the algorithms would perform best when certain parameter values ranges were met.

ACKNOWLEDGEMENT

This work was supported by the National Council of Science and Technology (CONACyT).

(Received June 1, 2017)

REFERENCES

[1] J. Anguera, A. Andújar, M. C. Huynh, C. Orlenius, C. Picher, and C. Puente: Advances in antenna technology for wireless handheld devices. Int. J. Antennas Propag. *2013* (2013), 1–25. DOI:10.1155/2013/838364

[2] C. A. Balanis: Antenna Theory: Analysis and Design. Fourth edition. John Wiley and Sons, New Jersey 2016.

[3] R. E. Bellman: Dynamic Programming. Princeton University Press 1957.

[4] A. Chowdhury, R. Giri, A. Ghosh, S. Das, A. Abraham, and V. Snasel: Linear antenna array synthesis using fitness-adaptive differential evolution algorithm. In: IEEE Congress on Evolutionary Comutation 2010, pp. 1–8. DOI:10.1109/cec.2010.5586518

[5] M. Clerc and J. Kennedy: The particle swarm – explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. *6* (2002), 58–73. DOI:10.1109/4235.985692

[6] W. C. Davidon: Variable metric method for minimization. J. Optim. *1*, (1991), 1–17. DOI:10.1137/0801001

[7] R. Díaz de León-Zapata, G. González, E. Flores-García, A. G. Rodríguez and F. J. González: Evolutionary algorithm geometry optimization of optical antennas. Int. J. Antennas Propag. *2016* (2016), 1–7. DOI:10.1155/2016/3156702

[8] R. C. Eberhart and Y. Shi: Comparing inertia weights and constriction factors in particle swarm optimization. In: Proc. 2000 Congress on Evolutionary Computation *1* (2000), pp. 84–88. DOI:10.1109/cec.2000.870279

[9] R. C. Eberhart and Y. Shi: Particle swarm optimization: developments, applications and resources. In: Proc. 2001 Congr. Evol. Comput. *1* (2001), pp. 81–86. DOI:10.1109/cec.2001.934374

[10] A. E. Eiben, R. Hinterding, and Z. Michalewicz: Parameter control in evolutionary algorithms. In: IEEE Trans. Evol. Comput. *3* (1999), 124–141. DOI:10.1109/4235.771166

[11] S. García, D. Molina, M. Lozano and F. Herrera: A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 Special Session on Real Parameter Optimization. J. Heurist. *15* (2009), 6, 617–644. DOI:10.1007/s10732-008-9080-4

[12] L. C. Godara: Handbook of Antennas in Wireless Communications. CRC Press, Inc., Boca Raton 2001. DOI:10.1201/9781420042146

[13] Y. He, J. Zhou, N. Lu, H. Qin, and Y. Lu: Differential evolution algorithm combined with chaotic pattern search. Kybernetika *46* (2010), 4, 684–696.

[14] M.-H. Ho, C.-C. Chiu, and S.-H. Liao: Bit error rate reduction for circular ultrawideband antenna by dynamic differential evolution. Int. J. RF Microw. Comput. Eng. *22* (2012), 260–271. DOI:10.1002/mmce.20604

[15] R. Hooke and T. A. Jeeves: "Direct Search" Solution of numerical and statistical problems. J. ACM *8* (1961), 212–229.

[16] N. Jin and Y. Rahmat-Samii: Hybrid Real-Binary Particle Swarm Optimization (HPSO) in engineering electromagnetics. IEEE Trans. Antennas Propag. *58* (2010), 3786–3794. DOI:10.1109/tap.2010.2078477

[17] A. R. Jordehi and J. Jasni: Parameter selection in particle swarm optimisation: a survey. J. Exp. Theor. Artif. Intell. *25* (2013), 527–542. DOI:10.1080/0952813x.2013.782348

[18] J. Kennedy and R. Eberhart: Particle swarm optimization. In: IEE Internat. Conf on Neural Networks *4* (1995), pp. 1942–1948. DOI:10.1109/icnn.1995.488968

[19] J. Kennedy and W. M. Spears: Matching algorithms to problems: an experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator. In: IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence 1998, pp. 78–83. DOI:10.1109/icec.1998.699326

[20] M. M. Khodier: Optimisation of antenna arrays using the cuckoo search algorithm. Microwaves, Antennas Propagation, IET *7* (2013), 458–464. DOI:10.1049/iet-map.2012.0692

[21] M. M. Khodier and C. G. Christodoulou: Linear array geometry synthesis with minimum sidelobe level and null control using particle swarm optimization. IEEE Trans. Antennas Propag. *53* (2005), 2674–2679. DOI:10.1109/tap.2005.851762

[22] G. Lalithamanohar, A. T. Praveen Kumar, and K. R. Subhashini: Constrained performance comparison of antenna synthesis using swarm and ecology techniques. In: 2013 1st Int. Conf. Commun. Signal Process. Their Appl. ICCSPA 2013, pp. 0-5. DOI:10.1109/iccspa.2013.6487256

[23] X. Li, W.-T. Li, X.-W. Shi, and J. Yang: Synthesis of antenna arrays with an efficient multiobjective differential evolution algorithm. Int. J. RF Microw. Comput. Eng. *24* (2014), 161–169. DOI:10.1002/mmce.20744

[24] Y. Liu, Y.-C. Jiao, and Y.-M. Zhang: A novel hybrid invasive weed optimization algorithm for pattern synthesis of array antennas. Int. J. RF Microw. Comput. Eng. *25* (2015), 154–163. DOI:10.1002/mmce.20844

[25] S. K. Mahto, A. Choubey, and S. Suman: Linear array synthesis with minimum side lobe level and null control using wind driven optimization. In: 2015 International Conference on Signal Processing and Communication Engineering Systems, pp. 191–195. DOI:10.1109/spaces.2015.7058246

[26] D. Mandal, S. P. Ghoshal, and A. K. Bhattacharjee: Wide null control of symmetric linear antenna array using novel particle swarm optimization. Int. J. RF Microw. Comput. Eng. *21* (2011), 376–382. DOI:10.1002/mmce.20526

[27] R. E. Mercer and J. R. Sampson: Adaptive search using a reproductive metaplan. Int. J. Systems Cybernet. *7* (1978), 3, 215–228. DOI:10.1108/eb005486

[28] M. A. Panduro and C. A. Brizuela: A comparative analysis of the performance of GA, PSO and DE for circular antenna arrays. In: 2009 IEEE Antennas and Propagation Society International Symposium 2009, pp. 1–4. DOI:10.1109/aps.2009.5171514

[29] M. E. H. Pedersen: Tuning & Simplifying Heuristical Optimization. University of Southampton 2010.

[30] M. E. H. Pedersen: SwarmOps. Numerical & Heuristic optimization. Online, 2011. Available: http://www.hvass-labs.org/projects/swarmops/.

[31] M. E. H. Pedersen and A. J. Chipperfield: Local unimodal sampling. Hvass Laboratories Technical Report HL0801 (2008), 1–10.

[32] N. Petrella, M. M. Khodier, M. Antonini, M. Ruggieri, S E. Barbin, and C. G. Christodoulou: Planar array synthesis with minimum sidelobe level and null control using particle swarm optimization. In: 2006 International Conference on Microwaves, Radar & Wireless Communications 2006, pp. 1087-1090. DOI:10.1109/mikon.2006.4345374

[33] E. A. Portilla-Flores, M.B. Calva-Yáñez, M. G. Villarreal-Cervantes, P. A. Niño Suárez, and G. Sepúlveda-Cervantes: Dynamic approach to optimum synthesis of a four-bar mechanism using a swarm intelligence algorithm. Kybernetika *50* (2014), 5, 786–803. DOI:10.14736/kyb-2014-5-0786

[34] J. Robinson and Y. Rahmat-Samii: Particle swarm optimization in Electromagnetics. IEEE Trans. Antennas Propag. *52* (2004), 397–407. DOI:10.1109/tap.2004.823969

[35] M. Secmen, M. F. Tasgetiren, and K. Karabulut: Null control in linear antenna arrays with ensemble differential evolution. In: Proc. 2013 IEEE Symposium on Differential Evolution (SDE) 2013, pp. 92–98. DOI:10.1109/sde.2013.6601447

[36] Y. Shi and R. C. Eberhart: Parameter selection in particle swarm optimization. In: Proc. 7th International Conference on Evolutionary Programming VII 1998, pp. 591–600. DOI:10.1007/bfb0040810

[37] M. Shihab, Y. Najjar, N. Dib, and M. Khodier: Design of non-uniform circular antenna arrays using particle swarm optimization. J. Electr. Eng. *59* (2008), 216–220.

[38] R. Storn: On the usage of differential evolution for function optimization. In: Bienn. Conf. North Am. Fuzzy Inf. Process. Soc. 1996, pp. 519–523. DOI:10.1109/nafips.1996.534789

[39] R. Storn and K. Price: Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. *11* (1997), 341–359. DOI:10.1023/a:1008202821328

[40] I. C. Trelea: The particle swarm optimization algorithm: convergence analysis and parameter selection. Inf. Process. Lett. *85* (2003), 317–325. DOI:10.1016/s0020-0190(02)00447-7

[41] S. H. Yang and J. F. Kiang: Adjustment of beamwidth and side-lobe level of large phased-arrays using particle swarm optimization technique. IEEE Trans. Antennas Propag. *62* (2014), 138–144. DOI:10.1109/tap.2013.2287280

Virgilio Zúñiga-Grajeda, Universidad de Guadalajara, CUTonalá, Jalisco. México.
e-mail: Virgilio.Zuniga@cutonala.udg.mx

Alberto Coronado-Mendoza, Universidad de Guadalajara, CUTonalá, Jalisco. México.
e-mail: Alberto.Coronado@cutonala.udg.mx

Kelly Joel Gurubel-Tun, Universidad de Guadalajara, CUTonalá, Jalisco. México.
e-mail: Joel.Gurubel@cutonala.udg.mx