ISSN 2007-9737

Offshore Wind Farm Layout Optimization via Differential Evolution

Valentín Osuna-Enciso[1], J. Israel Espinoza-Haro[1], Diego Oliva[2], Irán F. Hernández-Ahuactzi[1]

[1] Universidad de Guadalajara, Sciences Division, Centro Universitario de Tonalá, Mexico

[2] Universidad de Guadalajara, Electronic and Computer Division, Centro Universitario de Ciencias Exactas e Ingenierías, Mexico

{valentin.osuna, i.fernando.hernandez}@cutonala.udg.mx, israel.espinosa@alumno.udg.mx, diego.oliva@cucei.udg.mx,

Abstract. The Wind Farm Layout Problem (WFLP) consists in the placement of eolic generators (either in a grid, or at any position) into a delimited terrain. Several factors are taken into account to solve the WFLP, which include produced energy, costs - environmental, installation, maintenance, etc-, average useful life of turbines, among other. Likewise, optimization techniques involve the use of one or more objective functions, considering traditional as well as evolutionary approaches. Differential Evolution (DE) is an algorithm proposed for global optimization, whose operators are both simple to program and to utilize, still providing good convergence properties. The original authors of DE suggested its first five variants, which are: $best/1/bin$, $best/2/bin$, $current-to-best/1/bin$, $rand/1/bin$, and $rand/2/bin$. In this article it is proposed the comparison of five DE variants when they are used to solve 25 different instances of the WFLP; experimental results show that $DE/best/1/bin$ outperforms the remaining algorithms in terms of convergence velocity as well as in the quality of the obtained wind-farm.

Keywords. Evolutionary algorithm, differential evolution, optimization, wind farm layout problem.

1 Introduction

Several human activities depend directly on fossil fuels [37]; the continuity of such activities is endangered by two main problems: first, the fact that oil and its derivatives are finite resources (with a maximum reserve for 150 years [37]), and second, that they have a high environmental impact [24]. Accordingly, since recent years [13] research efforts are focused on making a better exploitation of renewable resources, such as: sun's radiation [46, 44, 6, 57], geothermic energy [10, 36, 50], wind energy [29, 1], tidal energy [2], among others.

From the aforementioned, wind is the most abundant renewable resource, found at every corner on the world [26]; therefore, improving the capture of wind energy is an actual and important issue of energy conversion [2].

One way to harness wind energy is by using conversion systems which include eolic turbines, that are capable of transforming mechanic energy into electricity. Usually, those turbines are placed over two kind of physical places: terrain or sea [5].

Such locations are commonly restricted by physical space, environmental and/or economical issues, interference among turbines, etc [34, 9]. Finding the optimal layout of wind farms, or Wind Farm Layout Problem (WFLP), is an open problem in scientific research [11, 35, 22, 4], because it belongs to the NonPolinomial (NP) - class [55].

There are several factors considered in the objective functions utilized to optimize the Wind Farm Layout Problem (WFLP) [9]; the most common are the costs associated with installation, operation, and maintenance [34], used to find the minimum cost of energy. Besides the use of an objective function, it is also common to consider restrictions in an optimization problem. In that

sense, a restriction to optimize the WFLP is the turbines location, due to the wake effect [48, 9]. Several models for the wake effect have been proposed, with the Jensen model being the most common [27], due to its mathematical simplicity, it is easily programmed, and also because it represents relatively well the wake decay effect, when it is compared against more complicated models [30].

In order to maximize the energy production it is necessary to increase the utilization of the eolic resource in the wind farm, while the restrictions are kept. One way to tackle this kind of problems is by using metaheuristic algorithms; this class of algorithms is nature-inspired, population-based, and is capable to find, relatively fast, good solutions for optimization problems in reasonable times [59]. The utilization of such techniques to solve the WFLP is a promising research area, as it was stated by [4]. In accordance with such study, several metaheuristic algorithms have been used to solve such problem.

In that sense, it is considered that Genetic Algorithm (GA) is the most used [9]; for example, in [23], a simple GA is utilized to find the best allocation of wind turbines into an equally spaced grid. Authors in [18] proposed a modification to individual codification in a GA in order to improve the disposition of wind turbines into a similar grid, while a weighted objective function is minimized.

Several researchers utilize similar grids, while are introduced GA's with improvements [21]. Other metaheuristics were utilized in similar studies, such as Ant Colony Optimization [20], Simulated Annealing [51], Firefly algorithm [40], Coral Reef Optimization [54], Imperialist Competitive algoritm [32], Binary Differential Evolution (DE) [28], as well as multiobjective Evolutionary Algorithms (EA) [34, 31]. In all the reviewed literature, every algorithm was independently proposed to solve the WFLP; however, authors in [31] state that there is a lack of comparative studies for algorithms applied to solve that problem; in such sense, in this work it is proposed an advance in that direction, because to the best of our knowledge, this has not been addressed in the literature, at least for the variants of DE utilized in this paper. In general terms, DE in its several variants (e.g. canonical, hybridizations, single, and multi-objective), as has been applied to solve various problems not only in scientific but also in industrial areas [15]. Considering the above mentioned, in this work are used five variants of the canonical DE for the offshore WFLP: best/1, rand/1, current-to-best/1, best/2, and rand/2, with all of them using the binomial crossover as well as synchronous population update; it is important to clarify that those versions were proposed in the seminal work of Storn and Price [56, 16].

The remainder of this paper is as follows: in section 2 is explained the wake effect model proposed by Jensen [27]. Section 3 clarifies the differences among the used algorithms, whereas section 4 gives experimental setup and results, which include some statistical results. Finally, in section 5 are drawn some conclusions as well as future work.

2 Wake Effect Modeling

Even though there are several wake effect models, one of the simplest is the one proposed by [27]; in that model some assumptions are considered, such as same height hubs, entrainment constant, turbine radii, among others. Also, that model is easy to program [48], which makes it one of the most preferred. A simple representation of the wake effect model is given in Figure 1, where the first turbine's wake is not affecting the second turbine.

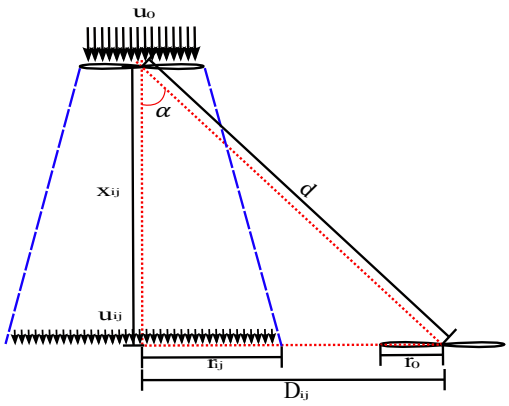


Fig. 1. The Jensen model

The reduced wind velocity, or the wake effect, is given by next equation:

$$u_{ij} = u_0 \cdot \left(1 - \left(\frac{2 \cdot a}{(1 + \gamma \cdot (x_{ij}/r_{ij}))^2}\right)\right), \quad (1)$$

where u_0 is the free wind speed, and it is considered that r_{ij} is the radius of the wake at the distance x_{ij} alongside the wake center:

$$r_{ij} = \gamma \cdot x_{ij} + r_0, \quad (2)$$

with the axial induction factor represented by:

$$a = \frac{1 - \sqrt{1 - C_T}}{2}, \quad (3)$$

where C_T is the thrust coefficient and γ is the entrainment constant, or the wake decay constant, calculated with:

$$\gamma = \frac{0.5}{ln\,(z/z_0)}, \quad (4)$$

where z is the hub height of the turbine, and z_0 is the surface roughness corresponding to the wind farm terrain. The simplest case is when the second turbine is completely immersed into the wake of only one turbine, where the mean velocity is thus given by Equations 1-4. However, this is not always the case (Figure 2):

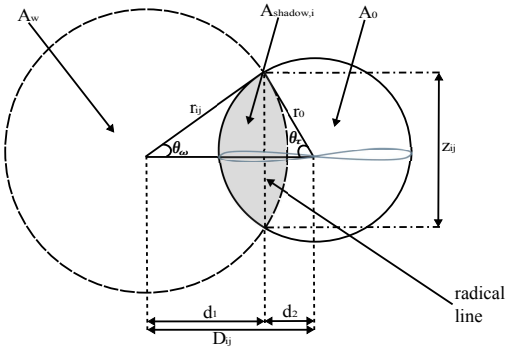


Fig. 2. Partial wake area

As it can be seen in this image, three cases could be considered regarding the wake effect:

1. $r_{ij} > D_{ij} + r_0$,
2. $r_{ij} \leq D_{ij} + r_0 \;\&\&\; r_{ij} \geq D_{ij} - r_0$,
3. $r_{ij} < D_{ij} - r_0$,

where D_{ij}, is the distance between the wake center and the center of the affected turbine and r_0 the turbine radius. For the first case, it happens that $A_{shadow,i} = A_0$, and therefore the velocity received by the turbine j its proportional to the the velocity leaving the turbine i. The third case is the easiest, because the wake from turbine i is not affecting the turbine j, and accordingly, $A_{shadow,i} = 0$. In the second case, in order to calculate the area $A_{shadow,i}$, it is necessary to make some trigonometric computations, considering the wake area as well as the second turbine area:

$$A_{shadow,i} = r_{ij}^2 \cdot \left(\theta_\omega - \frac{\sin\,(2 \cdot \theta_\omega)}{2}\right) + r_0^2 \cdot \left(\theta_\tau - \frac{\sin\,(2 \cdot \theta_\tau)}{2}\right), \quad (5)$$

or alternatively with:

$$A_{shadow,i} = r_{ij}^2 \cdot \cos^{-1}\left[\frac{d_1}{r_{ij}}\right] + r_0^2 \cdot \cos^{-1}\left[\frac{D_{ij} - d_1}{r_{ij}}\right] - D_{ij} \cdot z_{ij}, \quad (6)$$

considering that:

$$\theta_\omega = \cos^{-1}\left[\frac{D_{ij}^2 + r_{ij}^2 - r_0^2}{2 \cdot D_{ij} \cdot r_{ij}}\right], \quad (7)$$

$$\theta_\tau = \cos^{-1}\left[\frac{D_{ij}^2 + r_0^2 - r_{ij}^2}{2 \cdot D_{ij} \cdot r_0}\right], \quad (8)$$

$$A_0 = \pi \cdot r_0^2, \quad (9)$$

$$d_1 = r_{ij} \cdot \cos\,(\theta_\omega)\,, \quad (10)$$

$$z_{ij} = 2 \cdot r_{ij} \cdot \sin\,(\theta_\omega)\,. \quad (11)$$

A highlight is the fact that D_{ij} is the distance between the wake center and the center of the affected turbine; considering that, as the wind direction changes, such distance changes as well (Figure 3).

Once that $A_{shadow,i}$ and u_{ij} have been calculated with the Equations 1 and 7, the next step is to determine the velocity received by the

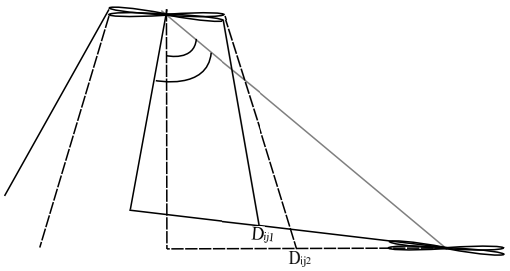


Fig. 3. Distances from wake center to second turbine center

turbine j:

$$\bar{u_j} = u_0 \cdot \left(1 - \sqrt{\sum_{i=1, i\neq j}^{N_j} \left(\frac{A_{shadow,i}}{A_0}\right) \cdot \left(1 - \frac{u_{ij}}{u_0}\right)^2}\right) \tag{12}$$

where N_j represents the number of turbines affecting turbine j and A_0 the circular area of each turbine.

Finally, to calculate the power generated by a turbine with a specific wind direction:

$$p_k(\bar{u_j}) = \begin{cases} 0 & \bar{u_j} \leq 2.3m/s, \\ 0.3\bar{u_j}^3 & 2.3 < \bar{u_j} \leq 12.8m/s, \\ 630 & 12.8 < \bar{u_j} \leq 18m/s, \\ 0 & \bar{u_j} > 18m/s. \end{cases} \tag{13}$$

In the consulted literature (e.g. [48], [9], [19]), in order to estimate the energy produced by the wind farm, three cases were considered:

Case 1: Uniform wind speed and direction: $k = 1$, $u_0 = 8m/s$.

Case 2: Uniform wind speed and variable direction: $0 < k \leq 360$, $u_0 = 8m/s$, $f_k = 0.02778$.

Case 3: Variable wind speed and direction: See Table 1.

In the interest of giving a more realistic scenario, the experiments developed in this work only consider the third case. Also, as the final power obtained by the wind farm depends mainly on the entrance velocity for each turbine, and such velocities are at the same time dependent on the shadow area of the wake, we propose to minimize the next objective function:

$$g(\mathbf{y}) = min\,(1 + \lambda_2) \cdot \left[\sum_{j=1}^{N_t} \sum_{i=1, i\neq j}^{N_j} N_j \cdot \lambda_1 \cdot \left(\frac{A_{shadow,i}}{A_0}\right) \cdot \left(\frac{r_0}{r_{ij}}\right)^2\right], \tag{14}$$

where:

$$\lambda_2 = \begin{cases} 0 & if\ \nexists\ d \leq 400, \\ 10 * \lambda_1 & otherwise. \end{cases} \tag{15}$$

With respect to λ_1, this is a factor empirically selected to penalize the value of the objective function g every time that any other turbine is closer than a security distance to the other turbines in the wind farm (usually, a standard value for such distance is ten times the turbine radius [48]).

Table 1. Functions of probability density wind distribution

	$u_0 = 8m/s$	$u_0 = 12m/s$	$u_0 = 17m/s$
k	f_k	f_k	f_k
0 - 270	0.0042	0.0084	0.0112
280	0.0042	0.0107	0.0135
290	0.0042	0.0126	0.0163
300	0.0042	0.0149	0.0191
310	0.0042	0.0149	0.0302
320	0.0042	0.0195	0.0358
330	0.0042	0.0149	0.0307
340	0.0042	0.0149	0.0191
350	0.0042	0.0126	0.0163
360	0.0042	0.0102	0.0135

3 Differential Evolution

Since the appearing of Genetic Algorithms (GA) [25], several Evolutionary Algorithms (EA) have been proposed including several versions of global optimization (mono-objective), and also for the optimization of problems that include several objective functions; some of those algorithms are the Non-dominated Sortign Genetic Algorithm II (NSGA-II) [38], the NSGA-III [39], the Multiple Objective Particle Swarm Optimization (MOPSO) [42], the Cellular Genetic Algorithm

for Multiobjective Optimization (MOCell) [45], and others. Every one of those algorithms are called evolutionary in the sense that their operators are inspired in the evolution theory, which was first proposed by Darwin and later enhanced by Mendel studies [52].

In this article, five variants of an EA, called Differential Evolution (DE), are used to solve the Wind Farm Layout Problem (WFLP). Some comparisons are also presented in order to find which one gives the best result for a given family of problems [58], that are related to the performance and quality of previously found solutions. The DE algorithm was proposed by [56] as a global optimization problem solver, and it has some interesting characteristics: it holds few parameters to tune, it is population-based, it does not use gradient information, and its operators are inspired by evolution, just to mention some of them [16].

3.1 Initialization

The first step in this algorithm consists in the initialization of a population of N_p individuals:

$$\mathbf{y}_i = \mathbf{y}^l + \textbf{rand()} \cdot \left(\mathbf{y}^u - \mathbf{y}^l\right); \quad i = 1, ..., N_p, \quad (16)$$

where $\mathbf{y}_i \in \mathbb{R}^D$ and $\{\mathbf{y}^u, \mathbf{y}^l\} \in \mathbb{R}^D$. $\mathbf{y}_i$ is a vector that represents a candidate solution and $\{\mathbf{y}^u, \mathbf{y}^l\}$ the upper and lower limits of search space. **rand()** is a vector of uniformly generated random numbers.

After the initialization, the operators that are proposed for each version make different path searches over the solution space [16]. In the next paragraphs some specific operators that belong to each version are explained.

3.2 Mutation Variations

After the initialization, for each individual in the population a mutant vector is generated. This is achieved in several ways:

$$\begin{gathered}
\mathbf{y}_i^m = \mathbf{y}_{best} + F \cdot \left(\mathbf{y}_{r_1} - \mathbf{y}_{r_2}\right); \\
DE/best/1/bin \\
\mathbf{y}_i^m = \mathbf{y}_{r_1} + F \cdot \left(\mathbf{y}_{r_2} - \mathbf{y}_{r_3}\right); \\
DE/rand/1/bin \\
\mathbf{y}_i^m = \mathbf{y}_i + F \cdot \left(\mathbf{y}_{best} - \mathbf{y}_i + \mathbf{y}_{r_1} - \mathbf{y}_{r_2}\right); \\
DE/current-to-best/1/bin \\
\mathbf{y}_i^m = \mathbf{y}_{best} + F \cdot \left(\mathbf{y}_{r_1} - \mathbf{y}_{r_2} + \mathbf{y}_{r_3} - \mathbf{y}_{r_4}\right); \\
DE/best/2/bin \\
\mathbf{y}_i^m = \mathbf{y}_{r_1} + F \cdot \left(\mathbf{y}_{r_2} - \mathbf{y}_{r_3} + \mathbf{y}_{r_4} - \mathbf{y}_{r_5}\right); \\
DE/rand/2/bin \quad (17)
\end{gathered}$$

considering that $r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq i$ are integer numbers, that are randomly obtained from a uniform distribution. $\mathbf{y}_{best}$ is the best candidate solution found so far. The value of F is used as a scaling factor to weight the difference between the selected parents.

These equations represent the five canonical variations proposed by [56]. Even tough there are several other variants of Differential Evolution (DE) [15], a more extensive comparison that considers all of them is out of the scope of this work.

3.3 Crossover

Once the mutant vector is acquired, the next step consists in completing a trial vector:

$$y_j^c = \begin{cases} y_j^m & \text{if } rand() \leq C_r \parallel j == j_{rand}, \\ y_{ij} & \text{otherwise,} \end{cases} \quad (18)$$

where $0 < C_r \leq 1.0$, and j_{rand} is an integer random number that ensures that at least one gene from the candidate solution has changed. This scheme is known as binomial crossover; other two crossover versions are the exponential, and the arithmetic recombination. Only the binomial crossover scheme, given by Equation 18, is actually employed in this article.

3.4 Population Update

The final operator of this technique is called selection:

$$\mathbf{y}_i = \begin{cases} \mathbf{y}^c & \text{if } g(\mathbf{y}^c) < g(\mathbf{y}_i), \\ \mathbf{y}_i & \text{otherwise}, \end{cases} \quad (19)$$

in which the trial vector is evaluated and compared against the original individual. This step is applied over the whole population (synchronous update); in other words, at each iteration both a mutant population, and a trial population are generated. Finally, the operator called population update is applied over the trial population.

The mentioned operators given by Equations Equation 17 - 19 are applied to each individual of the population until a certain criterion is reached, which it is usually a maximum iteration number.

4 Numerical Experiments

4.1 Problem Instances of the WFLP

The methodology adopted in this article, that was used to compare the Differential Evolution (DE) variants, is similar to the one utilized in [33]. In that sense, 25 instances of the WFLP were considered: $N_t \in [10, 20, 30, 40, 50]$, with physical terrain limits given by $y^u \in [1000, 1500, 2000, 2500, 3000]$. As it can be seen from N_t, the size of the candidate solution changes as the number of turbines changes; this is because each candidate solution is composed by the (p_{ij},q_{ij}) coordinates. For example, consider a problem instance with $N_t = 10$ and $y^u = 1000$, and therefore $D = 20$; in this case, a possible candidate solution is composed as:

$$\mathbf{y}_i = [p_{i1}, p_{i2}, p_{i3}, ..., p_{i10}, q_{i1}, q_{i2}, q_{i3}, ..., q_{i10}],$$

or alternatively as:

$$\mathbf{y}_i = [p_{ij}, q_{ij}], \quad i = 1, ..., N_p; \quad j = 1, ..., N_t,$$

whereas the box restrictions are:

$$\mathbf{y}^u = [1000, 1000, 1000, ..., 1000]; \quad \mathbf{y}^l = [0, 0, 0, ..., 0];$$

and in a more general representation:

$$\mathbf{y}^u = [y_j^u]; \quad \mathbf{y}^l = [y_j^l]; \quad j = 1, ..., 2 * N_t.$$

Again, it is necessary to remark that the problem's dimension increases in a twofold proportion with respect to the number of turbines placed into each instance of the wind farm; consequently, the WFLP instances considered in this paper are of 20, 40, 60, 80, and 100 dimensions.

4.2 Parameters Optimization

The term 'meta-optimization' (also called self-adaptive methodology [47]) is considered a relatively new concept, that consists in using a high level optimization technique to obtain the best tuning parameters of lower level optimization(s) algorithm(s) [14], [60]. By following such idea, in this article the Differential Evolution (DE) version best/1/bin is utilized as the meta-algorithm, and which was manually tuned by following the suggestions given in [41], whereas the lower level algorithms are the five DE variants examined in this work. The fitness function utilized for the meta-optimization is simply the sum of best individuals [60] found after each iteration of the meta-algorithm:

$$f(F_1, Cr_1, F_2, Cr_2, F_3, Cr_3, F_4, Cr_4, F_5, Cr_5) = min \sum_{i=1}^{5} g(\mathbf{y}_{ibest}), \quad (20)$$

since the limits of the search space are:

$$\mathbf{y}^u = [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0];$$

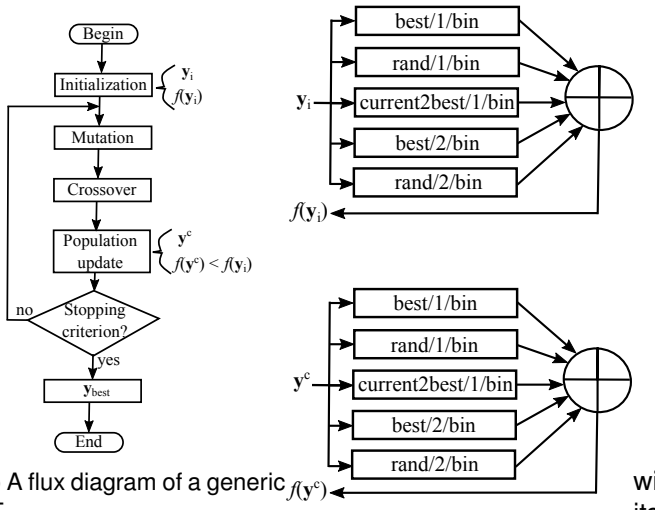
and

$$\mathbf{y}^l = [0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1];$$

$F_1, ..., F_5$ and $C_{r_1}, ..., C_{r_5}$, represents the scaling factor of difference and the crossover factor respectively, of each DE variant.

In the meta-optimization process, each candidate solution is a possible combination of the tuning parameters for each DE variant; therefore, the search space has 10 dimensions ($D = 10$). A simple graphic explanation of the meta-optimization process, utilized in this work, is given in Figure 4. As can be seen, in order to evaluate the fitness function f, every of the five

DE algorithms is executed once (200 iterations, $N_t = 46$ and $y^u = 2000$ [60], with the fitness function g) with the actual candidate solution (e.g. the tuning parameters), and the best individual evaluations are added every run per algorithm (Equation 20). Such process is repeated until the best configuration of parameters is found (Table 2).



(a) A flux diagram of a generic DE

(b) The fitness function f

Fig. 4. A simple meta-optimizer

Table 2. Tuning Parameters

DE variant	F	Cr
$best/1/bin$	0.38	0.5
$rand/1/bin$	0.86	0.15
$current-to-best/1/bin$	0.84	0.15
$best/2/bin$	0.3	0.8
$rand/2/bin$	0.58	0.1

4.3 Wind Farm Considerations

In this article it is taken into account the fact that the collocation of each turbine could be anywhere in the offshore wind farm (e.g. [53, 47, 9]) as opposed to other cases where it is divided in an equally spaced grid of cells (e.g. [7, 8, 10]).

As mentioned before, with the idea of making simulations as real as possible, in this paper it is only considered the case when the wind speed and direction are variable [12], according to a certain probability density function. The wind farm remaining parameters are standard values, usually given by other authors in similar studies (e.g. [48]); those values are: $z_0 = 0.3$, $z = 60$, and $C_T = 0.88$.

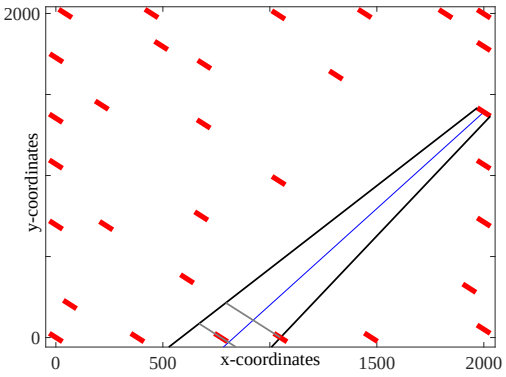


Fig. 5. A wake affecting other two turbines

Moreover, the objective function is mainly related with the minimization of the shadow areas and its accumulated penalties; in other words, the main idea behind this objective function is to minimize the number of turbines affecting to each turbine in the wind farm. Lets consider a possible configuration, as shown in Figure 5. In this case, two turbines are both partially, and completely immersed into the wake of a third turbine; therefore, the shadow area of the first turbines are respectively accumulated, together with a penalization factor.

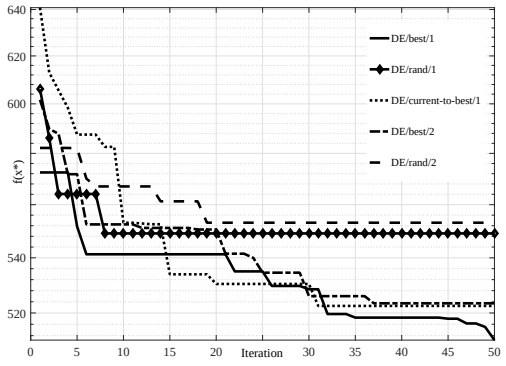


Fig. 6. Convergence of the algorithms after one run

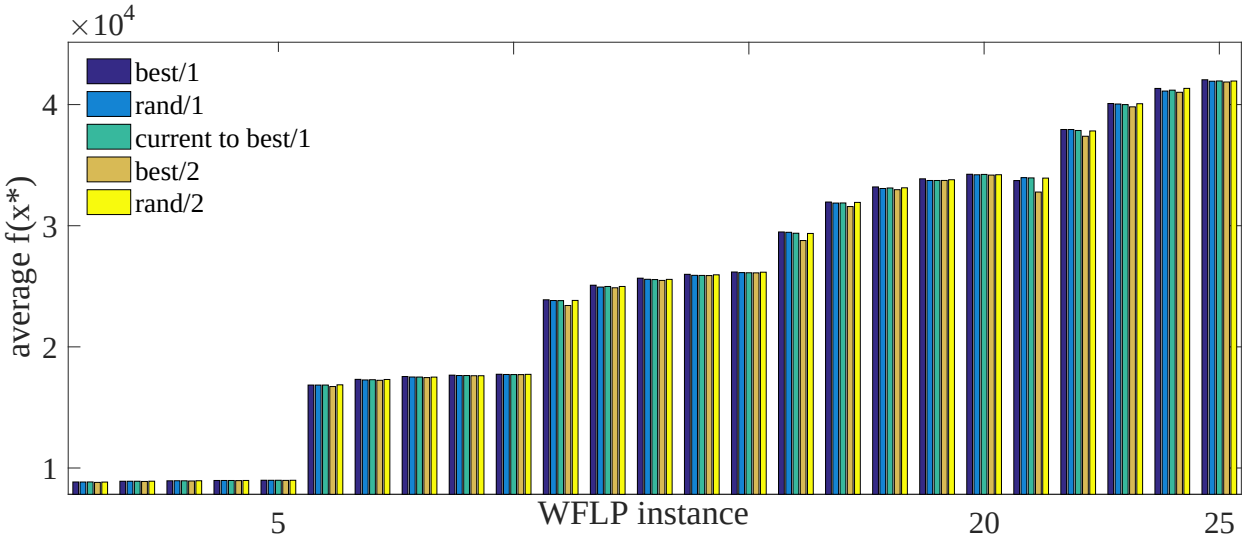


Fig. 7. Experimental outcomes

Table 3. Average best individuals for the 25 problem instances of the WFLP (in kWh)

N_t	WF size	best/1	rand/1	curtobest/1	best/2	rand/2
10	1000X1000	8846.81	8842.90	**8848.45**	8812.75	8839.03
10	1500X1500	8907.04	8908.09	8907.26	8891.01	**8910.96**
10	2000X2000	8944.64	**8948.59**	8943.72	8927.31	8948.54
10	2500X2500	8970.86	8970.45	8969.51	8957.80	**8971.97**
10	3000X3000	**8989.90**	8986.20	8988.10	8978.11	8987.19
20	1000X1000	16847.52	16845.48	16848.12	16724.52	**16867.48**
20	1500X1500	**17318.15**	17269.29	17289.16	17240.88	17308.15
20	2000X2000	**17552.24**	17511.51	17508.70	17463.40	17504.18
20	2500X2500	**17665.67**	17634.01	17633.32	17617.76	17617.71
20	3000X3000	**17739.49**	17721.21	17713.19	17708.64	17731.32
30	1000X1000	**23887.42**	23824.83	23815.34	23415.82	23834.04
30	1500X1500	**25089.43**	24938.36	24985.92	24878.41	24985.90
30	2000X2000	**25669.91**	25578.55	25555.24	25486.11	25571.06
30	2500X2500	**25988.81**	25906.26	25897.38	25885.10	25947.96
30	3000X3000	**26180.01**	26128.43	26117.32	26109.38	26164.00
40	1000X1000	**29485.65**	29457.23	29380.95	28777.05	29361.93
40	1500X1500	**31956.53**	31871.57	31879.86	31583.46	31925.59
40	2000X2000	**33203.87**	33074.22	33116.09	32976.35	33130.64
40	2500X2500	**33873.82**	33735.19	33734.12	33738.25	33796.71
40	3000X3000	**34249.54**	34206.96	34235.98	34180.89	34210.11
50	1000X1000	33726.53	**33967.76**	33948.51	32780.35	33931.54
50	1500X1500	**37944.01**	37942.27	37863.35	37387.27	37823.94
50	2000X2000	**40088.87**	40046.81	39997.65	39814.76	40071.16
50	2500X2500	41333.62	41117.88	41191.57	41016.10	**41338.35**
50	3000X3000	**42051.30**	41929.66	41950.96	41864.38	41945.04

4.4 Evaluation and Comparison of Algorithms

In this section are given the considerations with respect to the evaluations of the performance for the five Differential Evolution (DE) variants prior to the experiments made in this work; for example, the utilized algorithms were programmed in Matlab 9.0.0, run on a PC with an Intel i7 multicore

Table 4. List of symbols for the WFLP

Name	*Description*
a	Axial induction factor
A_0	Circular area of each turbine
$A_{shadow,i}$	Area of the intersection between A_w and A_o
A_w	Wake area at distance x_{ij}
C_T	Thrust coefficient
D_{ij}	Distance between the wake center and the affected turbine's center
d	Distance between turbine i center and turbine j center
D_{ij}	Distance between the wake center and the affected turbine's center
d_1	Distance between the wake center and the radical line center
d_2	Distance between the affected turbine's center and the radical line center
f_k	Probability density of wind distribution for wind direction k
k	Wind direction
N_j	Number of turbines affecting turbine j
N_t	Number of turbines in the wind farm
$p_k(\overline{u}_j)$	Power generated by turbine j at wind direction k
r_0	Turbine radius
r_{ij}	Wake radius at distance x_{ij}
u_0	Free wind speed
u_{ij}	Reduced wind speed between turbine i and turbine j
$\overline{u}_j$	Wind speed received by turbine j
x_{ij}	Distance along the wake between turbine i and turbine j
z	Turbine hub height
z_0	Surface roughness
z_{ij}	Height of $A_{shadow,i}$ over the radical line
γ	Entrainment constant, also called wake decay constant
θ_ω	Angle between r_{ij} and D_{ij}
θ_τ	Angle between r_0 and D_{ij}
λ_1	Penalization factor related with a security distance among turbines

processor to 3.4 GHz, and with 16GB of RAM memory.

In terms of the methodology, every algorithm was executed 30 times for each problem instance (200 iterations per run, Figure 6), after which the best results were averaged [33], obtaining 25 results per DE version (Table 3); in order to determine the best algorithm for the WFLP, those results were later statistically compared by means of the Wilcoxon non-parametric test [3, 43].

This test is used when the distribution of data is unknown [17], and also to statistically probe two hypotheses: on one hand, the null hypothesis, which establishes that two samples came from the same experiment, and on the other hand, the alternative hypothesis, that considers that those specimens are significantly different and, therefore, they came from different experiments.

4.5 Results and Discussion

In this section are presented the results obtained by each algorithm. A simple graphic representation of those outcomes is depicted in Figure 7.

For the first 10 instances, it is clear that each algorithm behaves in a similar fashion; in this case, it is relatively easy for every algorithm to place 10 and 20 turbines into each wind farm, particularly when the terrain's size increases, which is the case for every Wind Farm Layout Problem (WFLP) instance. However, as the turbines' number extends to 30, 40, and 50,

Table 5. List of symbols for DE

Name	*Description*
C_r	Crossover factor
D	Problem dimension
f	Meta-optimization objective function
F	Scaling factor of difference
g	Shadow objective function
N_p	Population of candidate solutions
(p, q)	Coordinates where a single turbine is placed
rand()	Vector of uniformly generated random numbers
$\mathbf{y}_i$	Vector that represents a candidate solution
$\mathbf{y}^u, \mathbf{y}^l$	Upper and lower limits of the search space
$\mathbf{y}_{best}$	Best candidate solution found so far
$\mathbf{y}_{r_1,r_2,...,r_5}$	Candidate solutions randomly taken from population
$\mathbf{y}_i^m$	Mutated vector
$\mathbf{y}_i^c$	Trial vector

Table 6. Wilcoxon p-values for DE/rand/2/bin against the other DE variants

best/1 vs.	**p-value**	**Observation**
rand/1	0.00054	Different
currenttobest/1	0.00049	Different
best/2	1.22903×10^{-5}	Different
rand/2	0.0022	Different

Table 7. Wilcoxon p-values for DE/best/1/bin against the other DE variants

rand/2 vs.	**p-value**	**Observation**
rand/1	0.07800	Failure to reject the null hypothesis
currenttobest/1	0.08265	Failure to reject the null hypothesis
best/2	1.38980×10^{-5}	Different

the problems then become more complicated, and therefore the differences of the algorithms are more evident. From the same graphic, it can be seen that in many instances, the DE/best/2/bin has the poorest performance. Yet, in order to give a detailed statistical explanation to the depicted results (shown in Table 3), the outcomes of the Wilcoxon non-parametric test are displayed in Table 6. As it can be seen after reviewing Table 3, the best technique seems to be the DE version best/1/bin, and therefore, it is statistically compared against the remaining algorithms; according to those results, that algorithm is the best to solve the WFLP, at least for the considered problem instances. In other words, for every pair of algorithms and based on the Wilcoxon non-parametric test, it can be concluded with a confidence of 5% that the algorithm **DE/best/1/bin** outperforms every other algorithm in the given experimental set, and therefore is the most adequate to tackle the offshore WFLP, at least under the conditions mentioned in this article. Finally, the second best algorithm was selected, which according to Table 3 is the so-called rand/2. That algorithm was compared against the remaining three (Table 6); the results suggest that the performance of the algorithms rand/2, rand/1, and current-to-best/1, is quite similar. In that sense, it can be considered, with a confidence of the 5%, that every algorithm behaves in a similar way for the problem at hand. For the WFLP, it can be considered that DE/best/2 has the poorest efficiency (Table 3-6).

5 Conclusions and Future Work

In this paper is proposed a comparison among five different DE algorithms that are utilized to solve the Wind Farm Layout Problem (WFLP). After running the algorithms several times, it can be statistically

stated that the version DE/best/1/bin finds the best results regarding to the quality of the obtained solution. As it has been stated, a possible future venue in the area of Offshore Wind Farm Layout Optimization could be the contrast between other variants of DE, such as hybridizations in order to find the most appropriate algorithm that delivers a better solution for the aforementioned problem.

Even though it has been only mentioned in this article, another established research area in renewable energy is the use of parallel as well as multiobjective optimization approaches [4], [53]; in that sense, a possible extension for this work is the application of several multi-objective algorithms such as the NSGA-II [38], the NSGA-III [39], the MOPSO [42], the MOCell [45], etc., for the WFLP optimization, assuming the objective function that has been proposed in this work. Likewise, the Net Present Value [53], the Energy Yield [49], the Cost of Energy [9], and others can also be considered. Moreover, in order to produce further improvements, the use of more than two objective functions could be potentially achieved by means of using the first two as the functions being optimized, while the remaining are kept as restrictions for the WFLP.

Acknowledgements

Authors express their gratitude to the Mexican National Council for Science and Technology (CONACYT) and PRODEP, for financing this work under various grants.

References

1. **Ackermann, T. (2005).** *Wind Power in Power Systems*.
2. **Akhtar, H., Syed, M. A., & Muhammad, A. (2017).** Emerging renewable and sustainable energy technologies: State of the art. *Renewable and Sustainable Energy Reviews*, Vol. 71, pp. 12–28.
3. **Ali, E., Abd Elazim, S., & Abdelaziz, A. (2016).** Ant lion optimization algorithm for renewable distributed generations. *Energy*, Vol. 116, pp. 445–458.
4. **Banos, R., Manzano-Agugliaro, F., Montoya, F., Gil, C., Alcayde, A., & Gómez, J. (2011).** Optimization methods applied to renewable and sustainable energy: A review. *Renewable and Sustainable Energy Reviews*, Vol. 15, No. 4, pp. 1753–1766.
5. **Bilgili, M., Yasar, A., & Simsek, E. (2011).** Offshore wind power development in europe and its comparison with onshore counterpart. *Renewable and Sustainable Energy Reviews*, Vol. 15, No. 2, pp. 905–915.
6. **Brabec, C., Sariciftci, N., & Hummelen, J. (2001).** Plastic solar cells. *Advanced Funtional Materials*, Vol. 11, No. 1, pp. 15–26.
7. **Castro Mora, J., Calero Barón, J., Riquelme Santos, J., & Burgos Payán, M. (2007).** An evolutive algorithm for wind farm optimal design. *Neurocomputing*, Vol. 70, No. 16-18, pp. 2651–2658.
8. **Changshui, Z., Guangdong, H., & Jun, W. (2011).** A fast algorithm based on the submodular property for optimization of wind turbine positioning. *Renewable Energy*, Vol. 36, No. 11, pp. 2951–2958.
9. **Chehouri, A., Younes, R., Ilinca, A., & Perron, J. (2015).** Review of performance optimization techniques applied to wind turbines. *Applied Energy*, Vol. 142, pp. 361–388.
10. **Chen, Y., Li, H., Jin, K., & Song, Q. (2013).** Wind farm layout optimization using genetic algorithm with different hub height wind turbines. *Energy Conversion and Management*, Vol. 70, pp. 56–65.
11. **Chowdhury, S., Zhang, J., Messac, A., & Castillo, L. (2012).** Unrestricted wind farm layout optimization (uwflo): Investigating key factors influencing the maximum power generation. *Renewable Energy*, Vol. 38, No. 1, pp. 16–30.
12. **Chowdhury, S., Zhang, J., Messac, A., & Castillo, L. (2013).** Optimizing the arrangement and the selection of turbines for wind farms subject to varying wind conditions. *Renewable Energy*, Vol. 52, pp. 273–282.
13. **Chu, S. & Majumdar, A. (2012).** Opportunities and challenges for a sustainable energy future. *Nature*, Vol. 488, No. 7411, pp. 294–303.
14. **Cioaca, A. (2015).** Adaptive numerical control for dynamic data-driven applications. pp. WR5–WR10.
15. **Das, S., Mullick, S., & Suganthan, P. (2016).** Recent advances in differential evolution-an updated survey. *Swarm and Evolutionary Computation*, Vol. 27, pp. 1–30.

16. **Das, S. & Suganthan, P. (2011).** Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, Vol. 15, No. 1, pp. 4–31.
17. **Denux, T., Masson, M.-H., & Hébert, P.-A. (2005).** Nonparametric rank-based statistics and significance tests for fuzzy data. *Fuzzy Sets and Systems*, Vol. 153, No. 1, pp. 1–28.
18. **Emami, A. & Noghreh, P. (2010).** New approach on optimization in placement of wind turbines within wind farm by genetic algorithms. *Renewable Energy*, Vol. 35, No. 7, pp. 1559–1564.
19. **Emami, A. & Noghreh, P. (2010).** New approach on optimization in placement of wind turbines within wind farm by genetic algorithms. *Renewable Energy*, Vol. 35, No. 7, pp. 1559–1564.
20. **Eroglu, Y. & Seckiner, S. U. (2012).** Design of wind farm layout using ant colony algorithm. *Renewable Energy*, Vol. 44, No. C, pp. 53–62.
21. **Feng, J. & Shen, W. (2015).** Solving the wind farm layout optimization problem using random search algorithm. *Renewable Energy*, Vol. 78, pp. 182–192.
22. **González, J., Rodríguez, T., Mora, J., Burgos Payán, M., & Santos, J. (2011).** Overall design optimization of wind farms. *Renewable Energy*, Vol. 36, No. 7, pp. 1973–1982.
23. **Grady, S., Hussaini, M., & Abdullah, M. (2005).** Placement of wind turbines using genetic algorithms. *Renewable Energy*, Vol. 30, No. 2, pp. 259–270.
24. **Hill, J., Nelson, E., Tilman, D., Polasky, S., & Tiffany, D. (2006).** Environmental, economic, and energetic costs and benefits of biodiesel and ethanol biofuels. *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 103, No. 30, pp. 11206–11210.
25. **Holland, J. H. (1992).** *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA.
26. **Jacobson, M. (2009).** Review of solutions to global warming, air pollution, and energy security. *Energy and Environmental Science*, Vol. 2, No. 2, pp. 148–173.
27. **Jensen, N. (1983).** *A note on wind generator interaction*, volume 1.
28. **Jiang, D., Peng, C., Fan, Z., Chen, Y., & Cai, X. (2013).** Modified binary differential evolution for solving wind farm layout optimization problems. pp. 23–28.
29. **Johnson, C. C. & Smith, R. T. (1976).** Dynamics of wind generators on electric utility networks. *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-12, No. 4, pp. 483–493.
30. **Katic, I., Hjstrup, J., & Jensen, N. (1987).** *A Simple Model for Cluster Efficiency*, volume 1.
31. **Khan, S. & Rehman, S. (2013).** Iterative non-deterministic algorithms in on-shore wind farm design: A brief survey. *Renewable and Sustainable Energy Reviews*, Vol. 19, pp. 370–384.
32. **Kiamehr, K. & Hannani, S. (2014).** Wind farm layout optimization using imperialist competitive algorithm. *Journal of Renewable and Sustainable Energy*, Vol. 6, No. 4.
33. **Komaki, G. & Kayvanfar, V. (2015).** Grey wolf optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time. *Journal of Computational Science*, Vol. 8, pp. 109–120.
34. **Kusiak, A. & Song, Z. (2010).** Design of wind farm layout for maximum wind energy capture. *Renewable Energy*, Vol. 35, No. 3, pp. 685–694.
35. **Lackner, M. & Elkinton, C. (2007).** An analytical framework for offshore wind farm layout optimization. *Wind Engineering*, Vol. 31, No. 1, pp. 17–31.
36. **Lentz, A. & Almanza, R. (2003).** Geothermal-solar hybrid system in order to increase the steam flow for geothermic cycle in cerro prieto, mexico. volume 27, pp. 543–546.
37. **Lewis, N. & Nocera, D. (2006).** Powering the planet: Chemical challenges in solar energy utilization. *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 103, No. 43, pp. 15729–15735.
38. **Li, H. & Zhang, Q. (2009).** Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. *IEEE Transactions on Evolutionary Computation*, Vol. 13, No. 2, pp. 284–302.
39. **Li, X., Zeng, S., Qin, S., & Liu, K. (2015).** Constrained optimization problem solved by dynamic constrained nsga-iii multiobjective optimizational techniques. pp. 2923–2928.
40. **Massan, S.-U.-R., Wagan, A., Shaikh, M., & Abro, R. (2015).** Wind turbine micrositing by using the firefly algorithm. *Applied Soft Computing Journal*, Vol. 27, pp. 450–456.

41. **Mezura-Montes, E., Miranda-Varela, M., & Del Carmen Gómez-Ramón, R. (2010).** Differential evolution in constrained numerical optimization: An empirical study. *Information Sciences*, Vol. 180, No. 22, pp. 4223–4262.
42. **Mostaghim, S., Branke, J., & Schmeck, H. (2007).** Multi-objective particle swarm optimization on computer grids. pp. 869–875.
43. **Nabil, E. (2016).** A modified flower pollination algorithm for global optimization. *Expert Systems with Applications*, Vol. 57, pp. 192–203.
44. **Nazeeruddin, M. K., Kay, A., Rodicio, I., Humphry-Baker, R., Mueller, E., Liska, P., Vlachopoulos, N., & Gratzel, M. (1993).** Conversion of light to electricity by charge-transfer sensitizers on nanocrystalline tio2 electrodes. *J. Am. Chem. Soc*, Vol. 115, pp. 6382–6390.
45. **Nebro, A., Durillo, J., Luna, F., Dorronsoro, B., & Alba, E. (2009).** Mocell: A cellular genetic algorithm for multiobjective optimization. *International Journal of Intelligent Systems*, Vol. 24, No. 7, pp. 726–746.
46. **O'Regan, B. & Gratzel, M. (1991).** A low-cost, high-efficiency solar cell based on dye-sensitized colloidal tio2 films. *Nature*, Vol. 353, No. 6346, pp. 737–740.
47. **Pérez, B., Mínguez, R., & Guanche, R. (2013).** Offshore wind farm layout optimization using mathematical programming techniques. *Renewable Energy*, Vol. 53, pp. 389–399.
48. **Peuuri, F., Cab, C., Carvente, O., Zambrano-Arjona, M., & Tapia, J. (2016).** A study of the classical differential evolution control parameters. *Swarm and Evolutionary Computation*, Vol. 26, pp. 86–96.
49. **Pookpunt, S. & Ongsakul, W. (2013).** Optimal placement of wind turbines within wind farm using binary particle swarm optimization with time-varying acceleration coefficients. *Renewable Energy*, Vol. 55, No. C, pp. 266–276.
50. **Quintero-Núñez, M., Sweedler, A., & Tanaka, S. (2006).** Renewable resources of energy in northern baja california, mexico. *WIT Transactions on Ecology and the Environment*, Vol. 99.
51. **Rivas, R. A., Clausen, J., Hansen, K., & Jensen, L. (2009).** Solving the turbine positioning problem for large offshore wind farms by simulated annealing. *Wind Engineering*, Vol. 33, No. 3, pp. 287–297.
52. **Rocca, P., Benedetti, M., Donelli, M., Franceschini, D., & Massa, A. (2009).** Evolutionary optimization as applied to inverse scattering problems. *Inverse Problems*, Vol. 25, No. 12.
53. **Rodrigues, S., Bauer, P., & Bosman, P. (2016).** Multi-objective optimization of wind farm layouts complexity, constraint handling and scalability. *Renewable and Sustainable Energy Reviews*, Vol. 65, pp. 587–609.
54. **Salcedo-Sanz, S., Gallo-Marazuela, D., Pastor-Sánchez, A., Carro-Calvo, L., Portilla-Figueras, A., & Prieto, L. (2014).** Offshore wind farm design with the coral reefs optimization algorithm. *Renewable Energy*, Vol. 63, pp. 109–115.
55. **Serrano González, J., Burgos Payán, M., Santos, J., & González-Longatt, F. (2014).** A review and recent developments in the optimal wind-turbine micro-siting problem. *Renewable and Sustainable Energy Reviews*, Vol. 30, pp. 133–144.
56. **Storn, R. & Price, K. (1997).** Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, Vol. 11, No. 4, pp. 341–359.
57. **Tang, C. (1986).** Two-layer organic photovoltaic cell. *Applied Physics Letters*, Vol. 48, No. 2, pp. 183–185.
58. **Wolpert, D. & Macready, W. (1997).** No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, pp. 67–82.
59. **Yang, X.-S. (2009).** Harmony search as a metaheuristic algorithm. *Studies in Computational Intelligence*, Vol. 191, pp. 1–14.
60. **Zhang, H. & Ishikawa, M. (2008).** Evolutionary particle swarm optimization: A metaoptimization method with ga for estimating optimal pso models. *Lecture Notes in Electrical Engineering*, Vol. 6, pp. 75–90.

Article received on 03/04/2017; accepted on 14/11/2017.
Correspoding author is Valentín Osuna-Enciso.