GA and PSO Applied to Wind Energy Optimization

Martín Bilbao[1], Enrique Alba[2]

[1] Universidad Nacional de la Patagonia Austral, Caleta Olivia, Argentina
mbilbao@uaco.unpa.edu.ar
[2] Universidad de Málaga, España
eat@lcc.uma.es

Abstract. In this article we analyze two kinds of metaheuristic algorithms applied to wind farm optimization. The basic idea is to utilize CHC (a sort of GA) and GPSO (a sort of PSO) algorithms to obtain an acceptable configuration of wind turbines in the wind farm that maximizes the total output energy and minimize the number of wind turbines used. The energy produced depends of the farm geometry, wind conditions and the terrain where it is settled. In this work we will analyze three study farm scenarios with different wind speeds and we will apply both algorithms to analyze the performance of the algorithms and the behavior of the computed wind farm designs.

Keywords: CHC, Geometric Particle Swarm Optimization, Optimization, Wind Energy, Metaheuristics.

1 Introduction

Nowadays, using renewable energies is an increasing area of research and development in all the world, because they are important alternatives to generate free and clean power. The raise of this energy is clear in Europe and America, being a strategic part of development for many countries like Argentina and Spain. A capital interest resides in combining a maximum of energy generation at the same time as reducing the total cost of the wind farm. A farm is a set of wind turbines, every one being costly, whose position is a strategic decision in orden to maximize the produced energy. One of the most important aspects of wind farm design is to obtain an optimal location of the wind turbines, because they recieve lower wind speed and less energy captures if e.g. they are located behind each other. This effect is called *the wake effect* [1]. The wake effect can be reduced by optimizing the geometry of the wind farm. Then, obtaining a maximum annual profit means taking into account the number of wind turbines and their proper positioning simultaneously. Therefore, an effective algorithm is necessary to get an optimal solution by using a mathematical model of the wind farm as close as possible to a real world complex problem.

Simulated Annealing and Distributed Genetic Algorithms have been used in the past to solved this kind of problem [2][3]. In this work we use other techniques that have provided in the past good solutions in problems like RND (Radio Network Design) that share some points in common to our work [4] and Geometric Particle Swarm Optimization (GPSO) [5] will be applied and analyzed here, showing also that they can provide new state of the art solutions to optimal wind farm design applications.

The rest of the article is structured as follows: Section 2 we will explain the wake model, the power model and the cost model used. Section 3 will detail CHC and GPSO the proposed algorithms. In section 4 we will show the experimental studies and discuss on the results obtained and in Section 5 the conclusions and future work.

2 Wind Farm Modelling

In this section we describe the mentioned inter-turbine wake effect model, the power model, and the cost model for our further mathematical manipulations. This are the basic components to deal with a realistic farm design, and they are combine together for the needed guidance offered to the design algorithms in their quest for an optimal farm configuration.

2.1 Wake Effect Model

The used model in this work is similar to the wake decay model developed by Katic [6]. Depending of the farm geometry, the wind turbine that is upwind of other wind turbine results in lower wind speeds than the one downwind, as shown in Fig. 1. The *velocity deficit* measures this effect [6]:

$$dV = U_0 - U_t = U_0 \frac{1 - \sqrt{1 - C_t}}{(\frac{1+2kX}{D})^2}, \tag{1}$$

where U_0 is the initial free stream velocity, U_t is the velocity in the wake at a distance X downstream of the upwind turbine, C_t is the thrust coefficient of the turbine, D is the diameter of the upwind turbine, and k is the wake decay constant. This model assumes that the kinetic energy deficit of interacting wakes is equal to the sum of the energy deficits of the individual wakes. Thus, the velocity deficit at the intersection of several wakes is:

$$U_t = U_0 \left[1 - \sqrt{\sum_{i=1}^{N} (1 - \frac{U_i}{U_0})^2} \right], \tag{2}$$

where U_i is the free strem velocity of the individual wake, and N is the number of wind turbines in the wind farm.

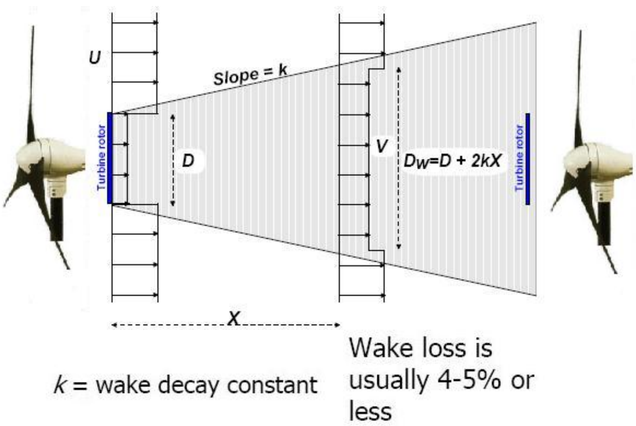


Fig. 1. Wake model for interaction between two wind turbines

2.2 Power Model

The wake model directly defines the power model, that is to be maximized. The power curve for the wind turbine under consideration in our work follows here:

$$P_i(kW) = \begin{cases} 0 & \text{for } U_x < 3m/s, \\ 0.3U_x^3 & \text{for } 3m/s \leq U_x \leq 13m/s, \\ 750 & \text{for } 13m/s \leq U_x \leq 25m/s, \\ 0 & \text{for } 25m/s < U_x \end{cases} \tag{3}$$

where Ux is the wind speed on the wind turbine.

the total power generation for all the wind turbines in the wind farm is:

$$P_{tot} = \sum_{i=1}^{N} P_i, \tag{4}$$

where N is the total number of wind turbines.

2.3 Cost Model

In our case, only the number of wind turbines influences the total cost to be minimized. The total cost per year for the entire wind farm, assuming a predefined and constant number of wind turbines, can be expressed as follows:

$$cost_{tot} = cost_{gy} N(2/3 + 1/3e^{-0.00174N^2}), \tag{5}$$

where $cost_{gy}$ represents the cost per wind turbine per year, and its value in this work is € $400,000$.

3 CHC and GPSO Algorithms

In this section we will explain the algorithms that we will use to solve the optimization problem of optimally design a wind farm. We have selected two well-known algorithms based in using populations of tentative solutions, a good feature found in a previous work [2].

3.1 CHC

The CHC algorithm was designed to work with populations coded as binary strings. CHC is a type of genetic algorithm that does not use mutation to produce new solutions; insteads it uses a mechanism called *HUX* crossover. The selection of individuals to complete the next generation is under only an elitist approach between parents and children.The *R* best solutions are retained and will be present in the next generation. When stagnation in the population is detected, a cataclysmic method of restart is used. The population tends to be homogeneous due to the absence of mutation and the elitist approach because there is no diversity; in order to solve this problem CHC implements a mechanism called *incest prevention*. The parents are selected randomly, but crossover takes place only if the individuals are not too close between them (Hamming distance) exceeds a certain threshold called *the threshold of incest*. As the population evolves, fewer individuals have the condition of not incest; in this case it is necessary to reduce the threshold. Every time that no change appears in the population (after one iteration) the threshold reduces in one unit.

The mechanism of crossover HUX also preserves diversity. This crossover copies in the two offspring all bits matched in both parents, and then copies half bits different in each offspring, such the Hamming distance between children and between children and parents is high. Once that the threshold of incest is 0, if *q* iterations pass without any new solution has entered the population, it means that the population has converged and the algorithm has stagnated, thus requiring a restart. All individuals except the best are modified by a mutation by bit inversion with very high probability (in our case is 50%). Fig.2 shows an example of crossover HUX. It generates a mask with the common bits from the parents and non-common bits are assigned randomly to each child taking into account that each one must take half of the bits not common.

The pseudocode of the CHC algorithm is shown in Algorithm 1.

Algorithm 1 CHC

$t \leftarrow 0$; /* evaluation */
$initialize(Pa, Distance)$ /*Initialize the population and the distances */
while $not\ stop\ criterion(t, Pa)$ **do**
    $Parents \leftarrow selected(Pa)$; /* Selected parent */
    $Offspring \leftarrow HUX(Parents)$ /* Crossover HUX */
    $evaluate(Pa, Offspring)$ /*evaluate Offspring*/
    $Pa \leftarrow elitism(Offspring, Pa)$
    if $Pa\ no\ change$ **then**
        $distance \leftarrow distance - 1$;
        if $distance == 0$ **then**
            $reset(Pa)$
            $initialize(distance)$
        end if
    end if
    $t \leftarrow t + 1$ /* One more generation */
end while
Return: best solution found.

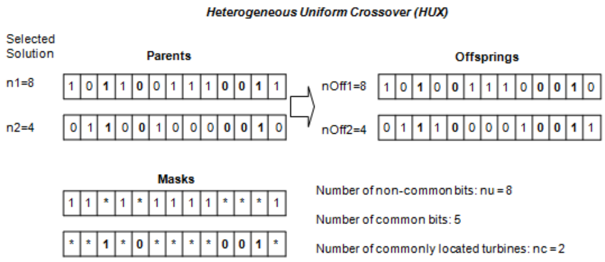


Fig. 2. Crossover HUX for CHC algorithm

3.2 Geometric Particle Swarm Optimization

The Geometric Particle Swarm Optimization (GPSO) enables us to generalize PSO to virtually any solution representation in a natural and straight-forward way, extending the search to other spaces, such a combinational ones. This property was demostrated for the cases of Euclidean, Manhattan and Hamming landscapes [7].

The key issue in this approach consists in using a multi-parental recombination of particles which leads to the generalization of a *mask-based crossover operation*, proving that it respects four requirements for being a *convex combination* in a certain space. This way, the mask-based crossover operation substitutes the classical movement in PSO, based on the *velocity* and *position update* operations, only suited for continuous spaces.

The pseudocode of the GPSO algorithm for Hamming spaces is shown in Algorithm 2. For a given particle *i*, three parents take part in the 3PMBCX operator (line 13). The current position x_i, the social best position g_i and the historical best position found h_i (of this particle). The weight values w_a, w_b and w_c indicate for each element in the crossover mask the probability of having values from the parents x_i, g_i or h_i respectively. A constriction of the geometric crossover forces w_a, w_b and w_c to be non-negative and add up to one.

Algorithm 2 GPSO

1: $S \leftarrow InitializeSwarm()$; /* Initialize Swarm */
2: **while** $not\ stop\ criteria$ **do**
3:   **for** $each\ particle\ x_i\ of\ S$ **do**
4:     $evaluate(x_i)$
5:     **if** $fitness(x_i) >= fitness(h_i)$ **then**
6:       $h_i \leftarrow x_i$;
7:     **end if**
8:     **if** $fitness(h_i) >= fitness(g_i)$ **then**
9:       $g_i \leftarrow h_i$;
10:     **end if**
11:   **end for**
12:   **for** $each\ particle\ x_i\ of\ S$ **do**
13:     $x_i \leftarrow 3PMBCX((x_i, w_a), (g_i, w_b), (h_i, wc))$
14:     $mutation(x_i)$
15:   **end for**
16: **end while**
17: **Return:** best solution found.

For Hamming spaces, which is the focus of this work, a *three-parent mask-based crossover* (3PMBCX) was defined as follows: given three parents *a*, *b* and *c* in $\{0,1\}^n$, generate randomly a crossover mask of length *n* with symbols from the alphabet $\{a, b, c\}$. Build the offspring o filling each position with the bit from the parent appearing in the crossover mask at the considered position.

In a convex combination, the weights w_a, w_b and w_c indicate for each position in the crossover mask the probability of having the symbols *a*, *b* o *c*. Fig. 3 shows an example of this kind of crossover.

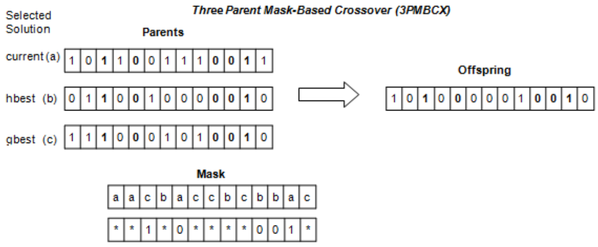


Fig. 3. Crossover 3PMBCX for the algorithm GPSO

4 Instantiating the algorithms for the Problem

In this section, we will explain how our approach works: we will introduce the fitness function, the representation used, and the customizing of CHC and GPSO for the problem.

4.1 Objective Function

The objective function that we are maximizing is the annual profit got from the wind farm, defined as follows [8]:

$$profit = \left[st - \left(\frac{cost_{tot}}{P_{tot}}\right) \right] P_{tot} \tag{6}$$

where st represents the estimated selling price for a KWh of electrical energy on the market in € (in this work it value is 0.1 €/KWh), P_{tot} represents the total expected energy output (kWh) of the wind farm per year, and $cost_{tot}$ is given by equation 5. The number of wind turbines is unknown and here also to be found by the used optimization algorithms.

4.2 Representations of Wind Turbine Locations

As other existing approches for the problem of Wind Energy Optimization we discretize the terrain in a matrix. A wind farm is logically divided into many small square like cells. Each cell in the wind farm grid can have two possible states: it contains a turbine (represented by 1) or it does not contain a turbine (represented by 0). A 10×10 grid is used here as the ground platform to place the wind turbines, and shown in Fig. 4. A binary string with 100 bits represents the location of the wind turbines in the wind farm. There are 2^{100} candidate solutions. The width at each cell, in the center of which a turbine would be placed, is equal to five times rotor diameter, $5D$ (or 220 m). Thus, the resulting dimension is $50D \times 50D$. The $5D$ square grid size also satisfies the rule of thumb of spacing requirements in the vertical and horizontal directions.

4.3 Customizing CHC and GPSO for the Problem

In this problem, GPSO was developed as follows: each particle *i* of the swarm consists of a binary vector x_i = (x_{i1}, x_{i2},... ,x_{in}) representing the terrain (10×10) where the wind farm will be installed; each element x_{ij} can have a wind turbine (represented by 1) or be empty (represented by 0). In this particular case (10×10) each partícle has a length (*n*) of 100 elements.

CHC was developed as follows: each individuals consists of a binary vector x_i = (x_{i1}, x_{i2},..., x_{in}) in the same representation than GPSO, and the same criteria for the positioning of the wind turbines.

5 Experimental Study

In this work we investigate three farm scenarios, in all of them we consider the case of uniform wind coming from the North, with different speeds for each case. Our aim is to analyse different wind farms and try to generalize our conclusions to guide designer in similar configurations. The first case we assume a wind speed of 12 m/s, in the second case a wind speed of 18 m/s, and the third case a wind speed of 25 m/s. We have selected these three scenarios based on the properties of wind profit of the mathematical model.

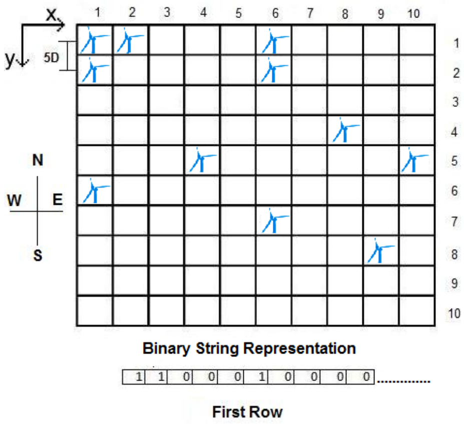


Fig. 4. Example of wind farm layout and the binary string representation

We show the different configurations for each case with the average fitness values, standard deviation of the fitness, total annual power output, average power output, number of wind turbines, average efficiency of the park, average execution time of each algorithm and the number of evaluation needs to find the better solution. We have also computed a statistical study comparing the average fitness values, and execution time of of each algorithm and we calcule the *p-value* with the *Kruskal-Wallis* test to conclude if it exists statistical significance between average fitness values and between average execution times. Each algorithm was executed 30 independent times with a stop criteria of $5,000,000$ evaluations. All the algorithms are executed in a MultiCore $2\times$ QuadCore 2 GHz and for the implementation of the algorithms we have used the library of optimization MALLBA [9].

For each scenario we used the properties of wind turbines and the parameters of the each algorithm shown in Table 1.

(a) Wind Turbine Property

Description	Parameter	Value
Nominal Power	P	750 *KWh*
Rotor Diameter	D	44 *m*
Trust Coefficient	Ct	0.88
Wake Decay Constant	k	0.11
Cut-in Velocity	V_i	13 *km/h*
Cut-Out Velocity	V_p	90 *km/h*

(b) Parameters of CHC

Description	Value
Population Size	128
Crossover	HUX
Cataclismic Mutation	Bit Flip 50%
Preserved Population	5%
Initial Threshold	25% of instance size
Convergence Value Q	1
Selection of Parents	Randomly
Selection of New Generation	Elitist

(c) Parameters of GPSO

Description	Value
Population Size	128
Size of the Swarm	100
Crossover	3PMBCX
Probability of Mutation	0.1%
Frecuency of Mutation	Bit flip 0.2%
Selection of Parent	x_i, g_i y h_i
Selection of New Generation	Elitist
Weight values w_a, w_b y w_c	0.2+0.1+0.7

Table 1. Properties of wind turbines and parameters used in CHC and GPSO

5.1 Scenario *(a)*: Wind Speed of 12 m/s

For this scenario we have executed both algorithms (CHC and GPSO) with the parameters shown in Table 1(b) and 1(c) respectively, and we obtained the best configuration of the farm ilustrated in the Fig. 6 and the numerical values shown in Table 2.

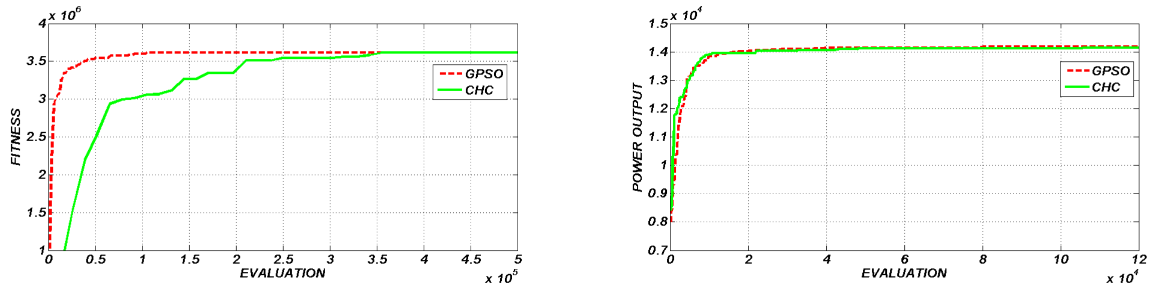
In this scenario CHC obtained better average fitness value, better power output and better efficiency.CHC needs more execution time and more evaluations to find the best solution than GPSO. GPSO obtained smaller values but with less execution time and evaluations. We calcule the *p-value* with the *Kruskal-Wallis* test for the average fitness values and it value is $2.28e^{-08}$. This value is smaller than $0,05$, so we conclude that it exists statistical significance between average fitneses and that CHC is more accurate and sligthly slower than GPSO. The *p-value* for the

Table 2. Results of scenario *(a)*

Description	CHC	GPSO
Average Fitness Values (€)	$\mathbf{3,608,160}$(± 10,985.8)	3,544,900 (± 39,926.4)
Average Power Output (KWH)	$\mathbf{14,205.13}$	14,132.89
Annual Power Output (MW)	124,471.36	124,471.36
Average Efficiency (%)	$\mathbf{91.28}$	90.86
Number of Wind Turbines (N)	30	30
Average Execution Time (s)	1.54	$\mathbf{1.15}$
Average Evaluation of Best Solution Found	259,735	$\mathbf{107,725}$

average execution time is 0.17, it is higher than 0.05, so we conclude that it does not exists statistical significance between average execution times.

Fig. 5 shows the evolution of the fitness (a) and the power output obtained (b). The configuration of the farm found for each algorithms is ilustrated in Fig. 6. We can see that the solution uses 30 wind turbines and they are aligned in rows keeping a constants distance between them, and in an orthogonal position with respect to the wind direction.



(a) Better fitness values (b) Better power output values

Fig. 5. Evolution of fitness values and power output for scenario *a*

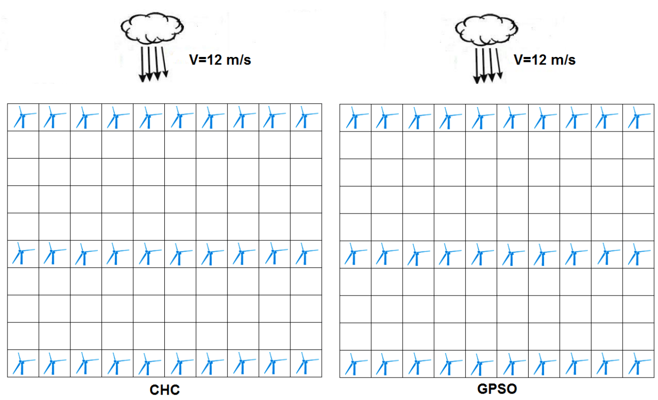


Fig. 6. Best configuration of the wind farm for both algorithms in scenario *a*

5.2 Scenario *(b)*: Wind Speed of 18 m/s

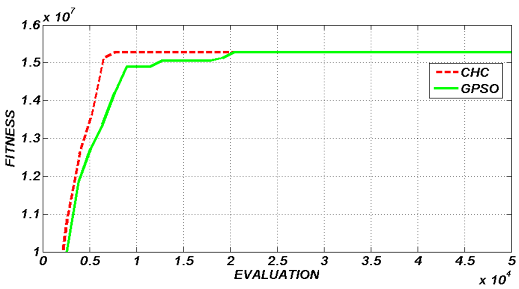
For this scenario we have executed both algorithms CHC and GPSO with the parameters shown in Table 1(b) and 1(c) respectively, and we obtained the best configuration of the wind farm illustrated in the Fig. 8, with the numerical values shown in Table 3

Table 3. Results of scenario *(b)*

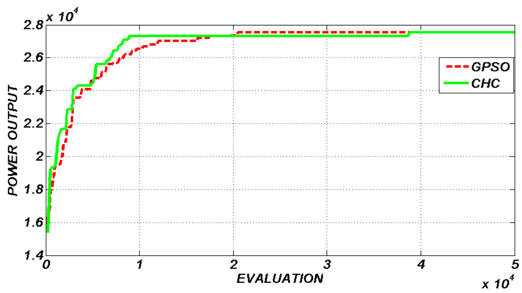
Description	CHC	GPSO
Average Fitness Values (€)	15,283,300($\pm$ 0)	15,283,300 ($\pm$ 0)
Average Power Output (KWH)	27,532.9	27,532.9
Annual Power Output (MW)	241,188.2	241,188.2
Average Efficiency (%)	91.77	91.77
Number of Wind Turbines (N)	40	40
Average Execution Time (s)	$\mathbf{0.12}$	0.25
Average Evaluation of Best Solution Found	$\mathbf{18,890}$	23,706

In this scenario CHC and GPSO obtained the same average fitness value, better power output and better efficiency. However CHC needed less execution time as it needed less evaluations than GPSO. We calculed the *p-value* with the *Kruskal-Wallis* test for the average fitness values and it results higher than 0.05, so we conclude that it does not exist stadistical significance between average fitnees values. The *p-value* for the average execution time is 0.002, it is smaller than 0.05, so we conclude that it exists statistical significance between average execution times.

Fig. 7 shows the evolution of the fitness (a) and the power output obtained (b). The best configuration of the wind farm found for each algorithms is ilustrated in Fig. 8, where we can see that the number of wind turbines are 40, they forming two rows in the center and in the opposite way with the wind sense.



(a) Better fitness values



(b) Better power output values

Fig. 7. Evolution of fitness values and power output for scenario *b*

5.3 Scenario *(c)*: Wind Speed of 25 m/s

For this scenario we have executed both algorithms, CHC and GPSO, with the parameters shown in Table 1(b) and 1(c) respectively, and we obtained the best configuration of the wind farm ilustrated in Fig. 10 and the numerical values shown in Table 4.

In this scenario CHC obtained again better values in most of metrics than GPSO, although the final configuration for the wind farm is the same for both algorithms. We have calculed the *p-value* with the *Kruskal-Wallis* test for the average fitness values and it results is $7.526e^{-05}$. This value is smaller than 0.05, so we conclude that it exists statistical significance between average fitnees values, then CHC is better than GPSO. The *p-value* for the average execution time is 0.023, it is smaller than 0.05, so we conclude that it exists statistical significance between average execution times.

Fig. 9 shows the evolution of the fitness (a) and the power output obtained (b). The best configuration of the wind farm found for each algorithms is ilustrated in Fig. 10, where we can see that the number of wind turbines are 50 and they all form the expected three rows in the center, in the opposite way than the wind direction.

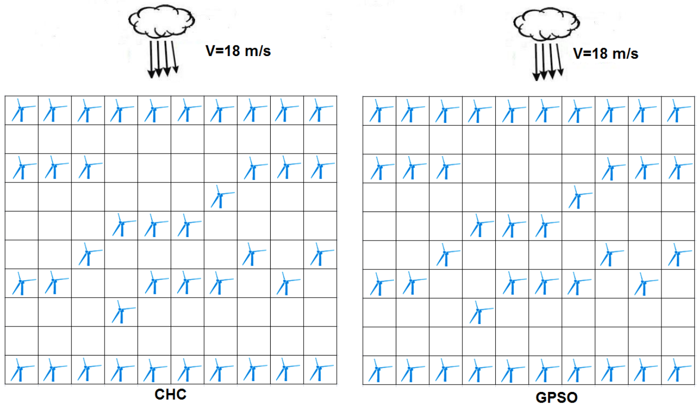
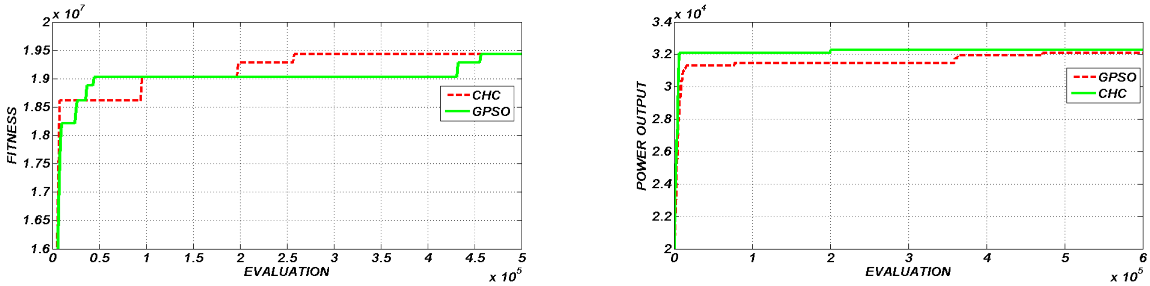


Fig. 8. Best configuration of the park for both algorithms in scenario *b*

Table 4. Results of scenario *(c)*

Description	CHC	GPSO
Average Fitness Values (€)	**19,345,000**(± 132,283)	19,094,000 (± 295,600)
Average Power Output (KWH)	**32,169.51**	31,882.98
Annual Power Output (MW)	282,654.54	282,654.54
Average Efficiency (%)	**85.76**	85.01
Number of Wind Turbines (N)	50	50
Average Execution Time (s)	**0.55**	1.54
Average Evaluation of Best Solution Found	**96,061**	201,024



(a) Better fitness values

(b) Better power output values

Fig. 9. Evolution of fitness values and power output for scenario *c*

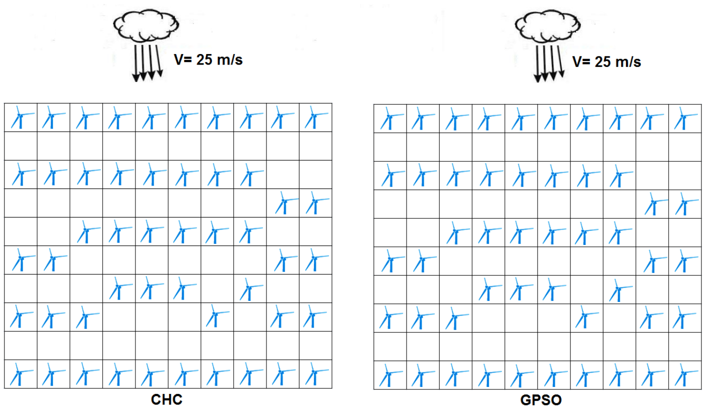


Fig. 10. Best configuration of the wind farm for both algorithms in scenario *c*

6 Conclusions and Future Work

We have here solved the problem of optimal placement of wind turbines in a wind farm with the objective to maximize the power energy produced with the less number of wind turbines to reduce the overall cost. Both algorithms are very competitive. In the first scenario CHC obtained better values in average fitness values, average efficiency and averare power output than GPSO. Both obtained the same final configuration of the wind farm but GPSO did it in less execution time and less number of evaluations. In the second scenario CHC and GPSO obtained the same preformance in the majority of metrics except in execution time and number of evaluation where CHC had better performance. In the third scenario CHC had a better performance than GPSO in all metrics, in this case both algorithms obtained the same configuration of the wind farm. We obtained the same configuration compared with previous work for the scenario *a*. Aparently cost function allow to find different solution and power function keeps similar evolution in both algorithms. In second and third scenario may need more time to find the optimal solution. As a future work we will consider additional farm models, including more real world factors, such as terrain effect and the esthetic impact. Also, we intend to study the scalability of this problem with bigger instances of the wind farm and new parameters of the wind turbines. Finally we plan to solve this problem as multiobjective consider two contrast function, the cost of design the wind farm and the produced energy.

7 Acknowledgements

We here acknowledge partial funding from Project UNPA-29/B105, University of Patagonia Austral Argentina, DIRICOM Project Nª P07-TIC-03044 and M* Project Nª TIN2008-06491-C04-01, University of Málaga Spain.
M. Bilbao acknowledge the co-operation of the University of Málaga for providing new ideas and constructive criticisms. Also to the University of Patagonia Austral, and the ANPCYT (National Agency to Promote Science and Technology) from which we receive continuous support.

References

1. J.F. Manwell, J.G. McGowan, and A.L. Rogers, *Wind Energy Explained-Theory, Design and Application*, 1st ed. , Reprint with correction, Jhon Wiley & Sons Ltd. ,2003, p.384 & p.44.
2. M. Bilbao, E. Alba, “Simulated Annealing for Optimization of Wind Farm Annual Profits” - *2nd International Symposium on Logistics and Industrial Informatics*, Linz, Austria. 2009.
3. H.S. Huang, “Distributed Genetic Algorithm for Optimization of Wind Farm Annual Profits”. *Intelligent Systems Applications to Power Systems*, 2007. ISAP 2007. International Conference on Volume , Issue , 5-8 Nov. 2007 Page(s):1 - 6.
4. E. Alba, G. Molina, F. Chicano, “Optimal Placement of Antennae using Metaheuristics” . *Numerical Methods and Applications (NM&A-2006)*, Borovents, Bulgaria. 2006.
5. E.Alba, J. García-Nieto, J. Taheri, A. Zomaya, “New Research in Nature Inspired Algorithms for Mobility Management in GSM Networks”. *Fifth European Workshop on the Application of Nature-inspired Techniques to Telecommunication Networks and other Connected Systems*, EvoWorkshops08, Springer-Verlag, pp. 1-10, Napoli Italy, 2008.
6. I. Katic, J. Hojstrup and N. O. Jensen, “A Simple Model for Cluster Efficiency”, *European Wind Energy Association Conference and Exhibition*, Rome-Italy, pp. 407-410, 7-9 October 1986.
7. A. Moraglio, C. Di Chio, R. Poli: “Geometric Particle Swarm Optimization”. In: Ebner, M., O’Neill, M., Ekárt, A., Vanneschi, L., Esparcia-Alcázar, A.I. (eds.)EuroGP 2007. LNCS, vol. 4445, Springer, Heidelberg (2007).
8. U. A. Ozturk and B. A. Norman, “Heuristic methods for wind energy conversion system positioning”, *Electric Power Systems Research*, vol.70, pp. 179-185, 2004.
9. E. Alba, F. Almeida, M. Blesa, C. Cotta, M. Díaz, I. Dorta, J.Gabarró, C. León, G. Luque, J. Petit, C. Rodríguez, A. Rojas, F. Xhafa,“Efficient Parallel LAN/WAN Algorithms for Optimization. The MALLBA Project”, *Parallel Computing* 32(5-6):415-440, 2006.