CHC and SA Applied To The Distribution Of Wind Turbines on Irregular Fields

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 and random search algorithm applied to distribution of wind turbines in a wind farm . The basic idea is to utilize CHC (a sort of GA) and Simulated Annealing 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, the terrain is irregular and we will analyze two study farm scenarios with a real wind distribution of Comodoro Rivadavia city in Argentina and we will apply both algorithms to analyze the performance of the algorithms and the behavior of the computed wind farm designs.

Keywords: CHC, Simulated Annealing, Optimization, Wind Energy, Metaheuristics.

1 Introduction

Wind energy is one of the most important alternative energies in the world. It is an economic, free, and clean energy and nowadays it can compete with other kinds of energy like fossil-fuel power production methods. The capital interest is to produce a maximum of energy 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 to minimize the *wake effect* [1] in orden to maximize the produced energy . The goal in this paper is obtain a better configuration of the wind farm by using the conditions of the wind and the terrain given by the enviroment. In this work, we include a real wind distribution from Comodoro Rivadavia in Argentina taken in 2008 [2]. For that we need effective algorithms, that should be first evaluated before utilization.

Simulated Annealing [3] and Distributed Genetic Algorithms [4] have been used in the past to solve this kind of problem. In a previous work we used CHC and GPSO considered constant North wind [5] and CHC y Simulated Annealing considered the real wind distribution and flat terrain [6]. Now, we compare two scenarios using the real wind distribution and we consider irregular terrain. We analize the best farm configuration found, the fitness value, the produced power, the efficiency, the performance of the algorithms in terms of their running, time and number of evaluations needed to obtain the best solution.

The rest of the article is structured as follows: Section 2 explains the wake model, the power model, and the cost model used. Section 3 will detail the real wind distribution, wind rose and field data. Section 4 describes CHC, SA and Random Search, the proposed algorithms. In Section 5 we will detail the objective function and the representation of wind turbine locations. In Section 6 we will detail the experimental studies and discuss on the results obtained; finally Section 7 summarizes 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. These are the basic components to deal with a realistic farm design, and they are combined together into an objective for the needed guidance of the function 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 [7]. 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 [7]:

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

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:

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

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

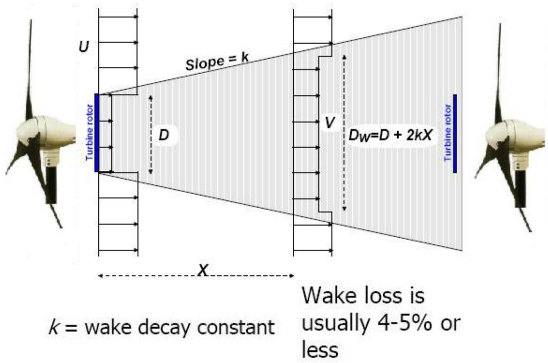


Fig. 1. Wake model for interaction between two wind turbines

2.2 Power Model

The previous wake model directly defines the power model, that is to be maximized. The power curve for the wind turbine under consideration is a Gamesa G47, whose power model (in KW) follows here:

$$P_i = \begin{cases} 0 & \text{for } U_x < 4m/s, \\ \rho \times A \times p(v) \times U_x^3 \times Cp & \text{for } 4m/s \le U_x < 12.5m/s, \\ 700 \times Cp & \text{for } 12.5m/s \le U_x \le 25m/s, \\ 0 & \text{for } 25m/s < U_x \end{cases} \tag{3}$$

where Ux is the wind speed on the wind turbine, $p(v)$ is the weibull probability of the wind v, ρ is the density of the environment $(1.23kg/m^3)$, A is the swept rotor area and Cp is the power coefficient of the wind turbine $(0.45$ in this case).

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

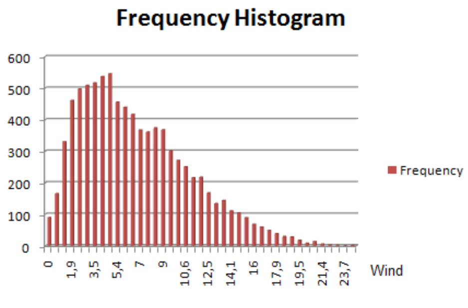
In our case, only the number of wind turbines influences the total cost to be minimized. The total cost per year for the entire wind farm, assuming a predefined and constant number of wind turbines, can be expressed as follows:

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

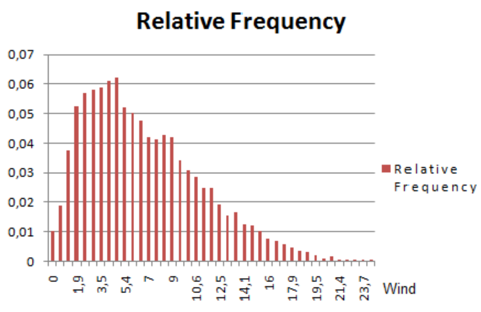
where $cost_{gy}$ represents the cost per wind turbine per year, and its value in this work is € $730,000$. We consider three different cost, the cost of installation (€ 800 per Kw installed), € 80000 per cost of foundation and € 90000 per cost of the tower.

3 Real Wind Distribution from Comodoro Rivadavia

In this section we introduce the real data obtained from Comodoro Rivadavia, Patagonia Argentina, and the process to obtain a good aproximation from the data for its later use in this work. Fig. 2(a) show the frequency histogram and the relative frequency of the actual wind. We can see that the most probable frequency of the wind is between 2 and 5 m/s, and high probability has a range between 5 and 12 m/s. The Weibull distribution is the most important probability distribution used in wind energy; it is usually used to approach the real wind data taken yearly (each 15 minutes) in our case. The process consists in obtaining a histogram of the wind with it frequency of ocurrence, relative and cummulative frequency. Then we apply a linear regression like least-squares to obtain a linear trend and calculate the parameters *k* and *b* of the Weibull distribution.



(a) Frequency histogram



(b) Relative frequency

Fig. 2. Absolute Frecuency and Relative Frecuency of the real wind distribution

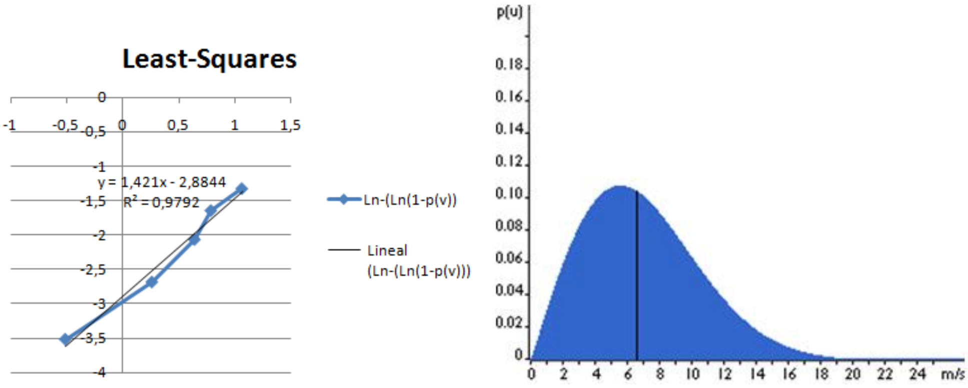
The probability of wind ocurrency is calculated as follows:

$$p(v) = (k/c)(v/c)^{k-1} \times e^{(v/c)^k}, \tag{6}$$

where *k* is a parameter form indicating if the wind speed tends to a particular value, and *c* is a parameter scale indicating how many winds are there in the environment. To obtain parameters *k* and *c* out of the natural measured data in the histograms we apply a linear regression whose form is *y* = *mx* + *b*, where $m = k$ and $c = e^{-b/k}$

We obtain the linear trends for the independent variable $x = ln(v)$ and the dependent variable $y = ln-(ln(1-p(v)))$, being *v* the wind speed and *p(v)* the cummulative frequency of the wind *v*, as shown in Fig. 3(a).

We obtained, with the least-squares approach, the parameter $k = 1.42$, the parameter $c = 7.53$, and 98% correlative coefficient, and then we have completed the Weibull distribution needed for our algorithms and shown in Fig. 3(b). Table 1 shows a comparison between real data and the Weibull distribution to show their acurracy.



(a) Least-Squares fitting for real data (b) Weibull distribution

Fig. 3. Weibull Distribution

Table 1. Comparison with Weibull Approach

Method	Mean Wind	Median
Real Data	6.79	6.1
Weibull Data	6.84	5.8

The resulting wind rose Fig 4 indicates the different frequency and direction of the wind. This rose is divided in to eight zones that indicate (in degrees) different cardinals point. In this scenario, the higher probability of ocurrence of wind direction is 270º (West direction). Thus our initial scenarios for evaluating the algorithms before a final real study will only consider in this work the wind coming from the West.

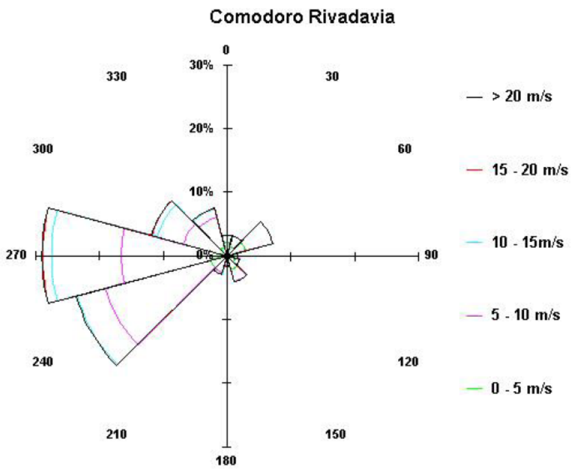


Fig. 4. Wind Rose of Comodoro Rivadavia (Patagonia Argentina)

4 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, a good feature found in a previous work [3][5][6].

4.1 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.5 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.

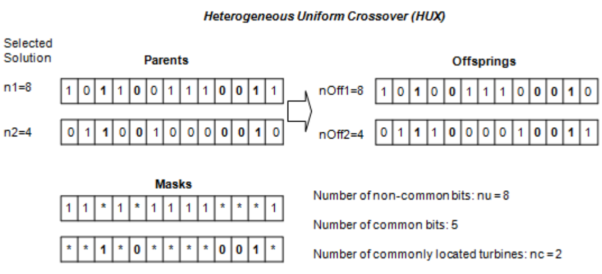


Fig. 5. Crossover HUX for CHC algorithm

4.2 Simulated Annealing

Simulated Annealing (SA) is a metaheuristic for global optimization, aimed at locating a good approximation to the global solution. Simulated annealing is a generalization of a Monte Carlo method for examining the equations of states and frozen states of *n*-body systems[6]. In SA only one tentative solution exists. The initial tentative solution is created randomly. The perturbation of a solution to get a neighbor solution is done by choosing one position where the wind turbine exists and move it to other location. If the new solution is better than the old solution, it becomes the present tentative solution. If not, it can be used anyway but with a probability regulated by a decreasing temperature parameter called Boltzmann probability $e^{-((s_n - s_b)/T)}$, where s_n is the present fitness value and s_b is the old fitness value, T is the temperature parameter whose initial value is 100. After that, the temperature is decremented in each iteration, thus decreasing the posibility of a worse solution is accepted. The iterative process finishes when a stop criterion is reached (e.g.,maximum number of step), and returns the solution found.

The pseudocode implementing our simulated annealing solver is shown in algorithm 2:

Algorithm 2 Simulated Annealing

```
s ← s0; /* Initial state */
sb ← s; /* Initial best solution */
k ← 0; /* evaluation count */
t ← 0; /* Initial temperature */
while k < kmax do
   sn ← neighbor(s); /* Pick some neighbor */
   /* Is this a new best? maximizing */
   if f(sn) ≥ f(sb) then
      sb ← sn;
   end if
   if Accept(sn, sb, t) then
      sb ← sn;
   end if
   t ← UpdateT() /* Update temperature */
   k ← k + 1 /* One more evaluation done */
end while
Return: Best solution found.
```

4.3 Random Search

The random search algorithm is a blind search, the search space is visited randomly choose a point in space and compared with the last point, if it is better than the previous point, we save it. The pseudocode implementing our random search solver is shown in algorithm 3:

Algorithm 3 Random Search

```
s ← random(); /* generate random point */
while k < kmax do
   sn ← random(); /* generate random point */
   /* Is this a new best? maximizing */
   if f(sn) ≥ f(s) then
      s ← sn;
   end if
   k ← k + 1 /* One more evaluation done */
end while
Return: Best solution found.
```

5 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, SA and RS for the problem.

5.1 Objective Function

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

$$profit = \left[st - (\frac{cost_{tot}}{P_{tot}}) \right] P_{tot} \pm G(x) \tag{7}$$

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.
The penalty function G(x) depends of the number of wind turbines included in the penalty zone, in this case we substract to fitness function the value calculated as follow:

$$G(x) = (\frac{k}{q}) * F(x) \tag{8}$$

where k is the number of the wind turbines included in the penalty zone and q is the total of places included in the restricted zone.

5.2 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. 6. 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 235 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.

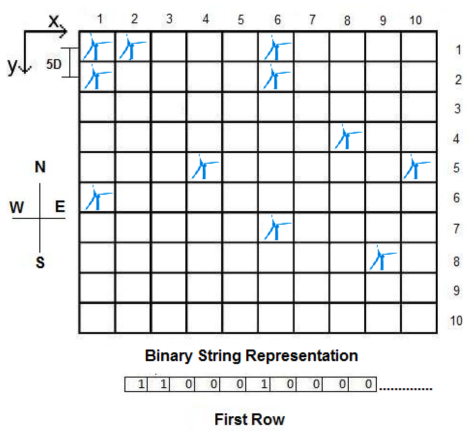


Fig. 6. Example of wind farm layout and the binary string representation

5.3 Customizing Algorithms for the Problem

In this problem, SA was developed as follows: the individual 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) the individual 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 SA, and the same criteria for the positioning of the wind turbines.

RS used the same criteria than SA and CHC to represent the random point.

6 Experimental Study

In this work we investigate two farm scenarios and we use the real wind distribution of Comodoro Rivadavia city. Our aim is to analyse two different kind of irregular terrain and try to generalize our conclusions to guide designer in similar configurations. Fig. shows the different kind of terrain used in this work.

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 2.

(a) Wind Turbine Property

Description	Parameter	Value
Nominal Power	P	700 *KWh*
Rotor Diameter	D	47 *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 SA

Description	Value
Temperature decay	0.99
Initial temperature	100
Probability of Mutation	0.3%

Table 2. Properties of wind turbines and parameters used in CHC and SA

6.1 Scenario *(a)*: NorthWest Irregular Terrain

For this scenario we have executed both algorithms (CHC and SA) and Random Search algorithm with the parameters shown in Table 2(b) and 2(c) respectively, and we obtained the best configuration of the farm ilustrated in the Fig. 7 and the numerical values shown in Table 3.

In this scenario CHC obtained better average fitness value, better power output and better efficiency. CHC needs less execution time and less evaluations to find the best solution than SA and RS. SA obtained better values than RS. We calcule the *p-value* with the *Kruskal-Wallis* test for the average fitness values and it value is $0.18e^{-04}$. This value is smaller than $0,05$, so we conclude that it exists statistical significance between average fitneses and that CHC is more accurate than SA. The *p-value* for the average execution time is $0.46e^{-07}$, it is smaller than 0.05, so we conclude that it exists statistical significance between average execution times and CHC is more faster than SA.

The configuration of the farm found for each algorithms is ilustrated in Fig. 7. We can see that the solution for CHC and SA uses 42 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.

Table 3. Results of scenario *(a)*

Description	CHC	SA	RS
Average Fitness Values (€)	$\mathbf{1,9658e+08}$($\pm$ 8,2909 e+06)	1,9625 e+08 ($\pm$ 8,3559 e+06)	1,5741 e+08 ($\pm$ 1,0266 e+07)
Average Power Output (KWH)	$\mathbf{12,624.69}$	12,406.89	10,421.32
Farm Coefficient (%)	$\mathbf{42.12}$	41.93	29.03
Number of Wind Turbines (N)	42	42	51
Average Execution Time (s)	$\mathbf{26.95}$	45.25	51.04
Average Evaluation of Best Solution Found	$\mathbf{2,335,749}$	2,526,919	3,221,191

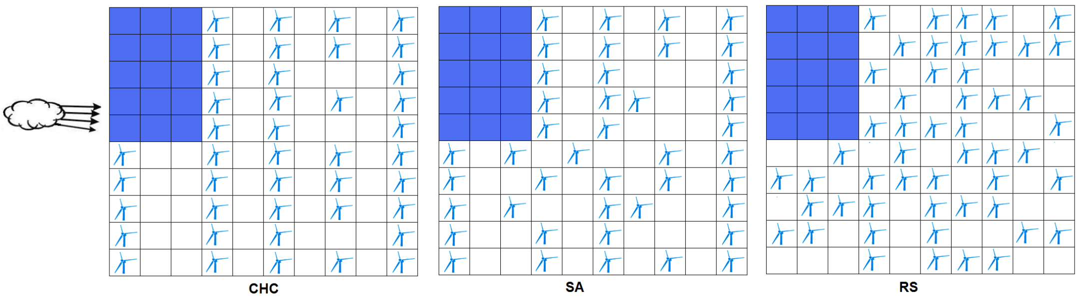


Fig. 7. Best configuration of the park for the three algorithms in scenario *a*

6.2 Scenario *(b)*: NorthWest and SouthEast Irregular Terrain

For this scenario we have executed both algorithms CHC and SA and RS with the parameters shown in Table 2(b) and 2(c) respectively, and we obtained the best configuration of the wind farm ilustrated in the Fig. 8, with the numerical values shown in Table 4

Table 4. Results of scenario *(a)*

Description	CHC	SA	RS
Average Fitness Values (€)	$\mathbf{1,4458e+08}$($\pm$ 7,1458 e+06)	1,4225 e+08 ($\pm$ 7,2296 e+06)	1,0014 e+08 ($\pm$ 1,1277 e+07)
Average Power Output (KWH)	$\mathbf{10,745.11}$	10,444.2	4,524.11
Farm Coefficient (%)	$\mathbf{43.85}$	41.44	16.15
Number of Wind Turbines (N)	$\mathbf{35}$	36	40
Average Execution Time (s)	$\mathbf{61.15}$	67.25	76.04
Average Evaluation of Best Solution Found	$\mathbf{4,154,111}$	4,214,245	4,222,985

In this scenario CHC obtained the best average fitness value, better power output and better efficiency again. SA needed less execution time as it needed less evaluations than RS. We calculed the *p-value* with the *Kruskal-Wallis* test for the average fitness values and it is smaller than 0.05, so we conclude that it 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.

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 for CHC is 35 against 36 for SA, they forming two rows in the center and in the opposite way with the wind sense.

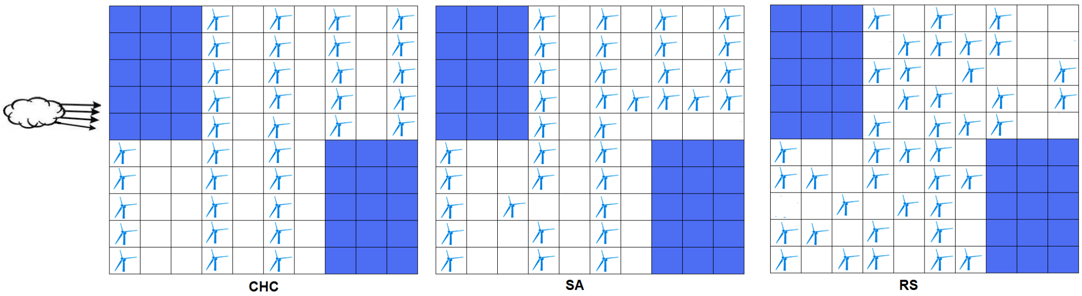


Fig. 8. Best configuration of the park for the three algorithms in scenario *b*

7 Conclusions and Future Work

We have here solved the problem of optimal placement of wind turbines in a wind farm with irregular terrain and the objective to maximize the power energy produced with the less number of wind turbines to reduce the overall cost. CHC and SA algorithms are very competitive. In the first scenario CHC obtained better values in average fitness values, average efficiency and average power output than SA. Both obtained similar final configuration of the wind farm but SA did it in more execution time and more number of evaluations. In the second scenario CHC obtained better preformance in the majority of metrics, SA obtained a good performance and RS obtained a bad performance in both scenarios. 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.

8 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. "http://www.eeolica.com.ar/"
3. M. Bilbao, E. Alba, "Simulated Annealing for Optimization of Wind Farm Annual Profits" - *2nd International Symposium on Logistics and Industrial Informatics*, Linz, Austria. 2009.
4. 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.
5. M.Bilbao, E. Alba, "GA and PSO Applied to Wind Energy Optimization" . *CACIC 09*, Jujuy, Argentina. 2009.
6. M. Bilbao, E.Alba, "CHC and SA Applied to Wind Energy Optimization Using Real Data". *CEC 10*, Barcelona, 2010.
7. 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.
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.