Multicore Parallelization of CHC for Optimal Aerogenerator Placement in Wind Farms

Martin Bilbao[1], Guillermo Leguizamon[2]

[1] Universidad Nacional de la Patagonia Austral, Caleta Olivia, Argentina
mbilbao@uaco.unpa.edu.ar
[2] Universidad Nacional de San Luis, Argentina
legui@unsl.edu.ar

Abstract. In this paper, we study a parallelization of CHC algorithm (Crossover elitism population, Half uniform crossover combination, Cataclysm mutation) to solve the problem of placement of wind turbines in a wind farm. We also analyze the solutions obtained when we use both, the sequential and parallel version for the CHC algorithm. In this case we study the behavior of parallel metaheuristics using an island model to distribute the algorithm in different cores and compare this proposal with the sequential version to analyse the number of evaluation to find the best configuration, output power extracted, plant coefficient, evaluations needed, memory consumption, and execution time for different number of core and different problem sizes.

Keywords: Wind Energy, Weibull Distribution, Wind Power, Evolutionary Computation, Metaheuristics

1 Introduction

Nowadays, the importance of environmental care is one of the most addressed issues in society and the main agenda of governments. For that we are being increasingly using renewable energy with a special care on avoiding the emission of gases to the atmosphere. Today a kind of energy that is moving around the world is the wind energy, which basically transforms the wind energy into electrical energy. It is designed for large-scale wind farms. A wind farm is a set of wind turbines distributed so that maximize wind energy contained in the wind and turns it into usable electricity. The capital interest is to produce a maximum of energy at the same time as reducing the total cost of the wind farm and choose it position is a strategic decision to minimize the *wake effect* [1] in order to maximize the produced energy. The goal in this paper is obtain a better configuration of the wind farm using parallel metaheuristics and analyze different scenarios to understand the behaviour of our algorithm.

Over the past years other kind of algorithms where used to solve similar problems, Simulated Annealing and Distributed Genetic Algorithms [2][3]. In a previous work we used CHC and GPSO considering constant North wind [4], and CHC and Simulated Annealing considering the real wind distribution and flat terrain [5]. Now, we design a parallel algorithm to solve two big scenarios using constant wind and regular terrain. The objective of this study is introducing differents version of a parallel algorithm to solve more complex and realistic instances. We analyze the best farm configuration found, fitness value, produced power, efficiency, performance of the algorithms in terms of their running, time and number of evaluations needed to obtain the best solution, speed-up and parallel efficiency.

The rest of the article is structured as follows: Section 2 explains the wake model, power model, and cost model used. Section 3 will detail the algorithm used. Section 4 describes the algorithms for wind farm design, objective function and representation of wind turbine locations. In Section 5 we will detail the analysis of sequential algorithm and its results. In section 6 we describe the parallel design of the algorithms and technologies used for this problem. In section 7 we will explain the results obtained for the parallel design and finally Section 8 summarizes the conclusions and future work.

2 Wind Energy Optimization Problem

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.
Design a wind farm efficiently is a very important work to generate the maximum energy possible, it depends of the relative distribution of the wind turbines for obtaining an optimal geometry of the wind farm. Wind turbines receive lower wind speed and less energy captures if they are located behind others. This effect called *the wake effect*[1] can be reduced by optimizing the geometry of the wind farm. The environment could has irregularities as lakes, mountains, forbidden places. Therefore, we need to design a wind farm with that conditions, so in this work we pursue the following goals, maximize the power output extracted and minimize the cost of installation.

2.1 Wake Effect Model

The used model in this work is similar to the wake decay model developed by Katic's[6]. Depending on the farm geometry the wind turbines that are upwind of other wind turbines result in lower wind speeds that the downwind turbines, as show in Fig. 1. The *velocity deficit*[2] measures this effect as shown in equation 1

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

where U_0 is the initial free stream velocity, U_x is the velocity in the wake at a distance X downstream of the upwind turbine (eq. 2), 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:

$$U_x = U_0[1 - \sqrt{\sum_{i=1}^{N} (1 - \frac{U_i}{U_0})^2}]. \tag{2}$$

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 [7], whose power model (in KW) follows here:

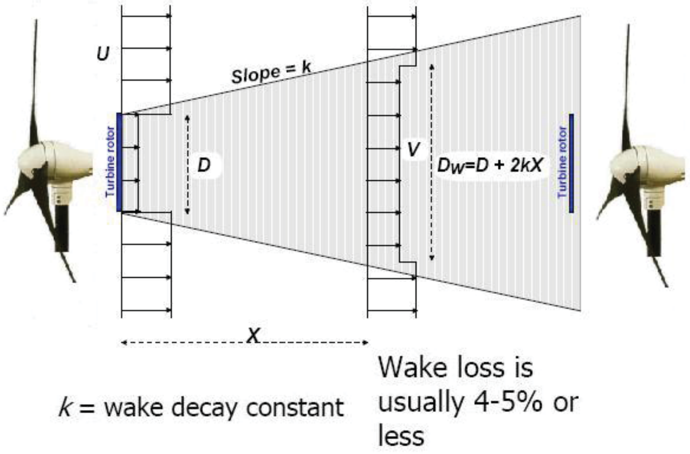

Fig. 1. Schematic of wake model

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

where Ux is defined in eq.2, 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 (0.40 in this case).

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

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

where N is the total number of turbines. For that, the power curve of this wind turbine is showed in Fig. 2

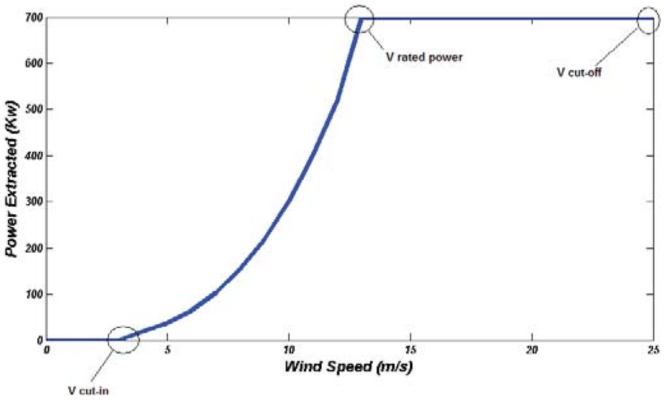

Fig. 2. Power curve of wind turbine

where velocity cut in is when the wind turbine starts to generate power, velocity rated power is when the wind turbine generate the maximum power, and velocity cut out is when the wind turbine stop to generate power.

2.3 Cost Model

In our case only the number of wind turbines is necessary for calculating and hopefully minimizing the total cost. The total cost per year for the entire wind farm assuming a predefined and constant number of wind turbines can be expressed as follows:

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

where $cost_{gy}$ represents cost per wind turbine per year, in this case it value is €400.000. We not consider the cost of installation, cost of foundation and cost of maintenance.

3 Metaheuristic Algorithms

Metaheuristics [14] are guided search techniques used in the optimization, where potential solutions are improved by incorporating an objective function. These techniques do not guarantee to find the optimal solution to the problem, but guarantee to find a good solution in reasonable computational times. For that in NP-hard problems the metaheuristics are a good option to find acceptable solutions very quickly.

3.1 CHC

In this section we describe the basic principles of the CHC [13] using in this work to resolve the problem. CHC is a sort of genetic algorithm and that have provided in the past good solutions in problems like RND (Radio Network Design) that share some points in common to our work[8]. 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 restarting 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%).

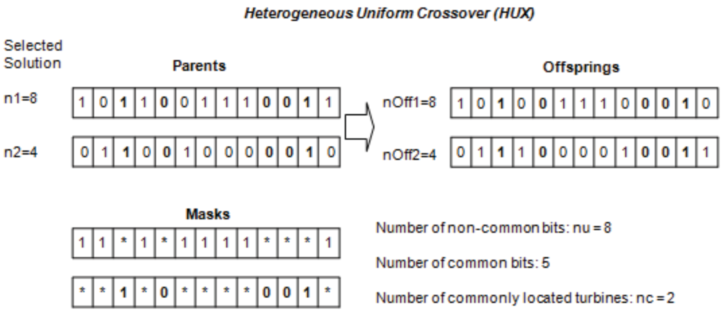

Fig. 3. Crossover HUX for CHC algorithm

Figure 3 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 respective pseudocode of the CHC algorithm is shown in Algorithm 1.

4 Wind Farm Design

In this section we show one of the most important things in this kind of algorithms, the representation of the potential solutions, each problem has a representation that best fits, we consider the binary representation to represent into the wind farm each solution because is very simple to match the real problem with our problem. Also we define the fitness function to be maximized in this case.

4.1 Representations of Wind Turbine Locations

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 wind turbine (represented by 1) or it does not contain a wind turbine (represented by 0). A 20×20 and 30×30 grid is used here as the ground platform to place the wind turbine. A binary string with 400 in the first case and 900

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 parents */
    $Offspring \leftarrow HUX(Parents)$ /* 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.
```

bits in the second case represents the presence or absence of the wind turbine in the wind farm. There are 2^{400} and 2^{900} candidate solutions respectively. The width at each cell, in the center of which a turbine would be placed, is equal to five rotor diameters, $5D$ (or 235 m) giving thus, the resulting dimension is $100D \times 100D$ in the first case and $150D \times 150D$. The $5D$ square grid size also satisfies the rule of thumb spacing requirements in the vertical and horizontal directions. The wind turbine considered in this paper has the next properties: rotor diameter 47 m, trust coefficient 0.88 and wake decay constant 0.11.

4.2 Objective Function

The objective function that we are maximizing is the annual profit got from the wind farm, defined as follows[9]:

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

where st represents the estimated selling price for a KWh of electrical energy on the market, in this case it value is 0.1, P_{tot} represents the total expected energy output (kWh) of the wind farm per year, and $cost_{tot}$ is given by[eq. 5].

5 Analysis Sequential Algorithms for Wind Farm Design

We study the scalability in two cases, dimension of problem 20×20, and 30×30, and we show graphics of evaluations to need for each size, evolution of fitness, different configurations of the wind farm, graphics of scalability and result tables with the average fitness, standard deviation, average power output, average efficiency, average time execution and number of wind turbines. The only case that we study is the case of uniform wind direction with a wind speed of 12 *m/s*; the wind direction comes from the north of the wind farm. The only change in wind speed for this case would occur in the wake of the wind turbines. The stop criteria for each algorithm is $5,000,000$ evaluations. We used Mallba library[10] to design the algorithms and we have executed this problem in CPU Multi-Core $2\times$Quad-Core 2.00 GHZ. Number of individuals in population was 128, probability of Crossover was $0,8$ and 0.5 the probability of reinitialize the population.

5.1 Case *a*: Sequential CHC applied for Wind Farm of dimension 20×20

For the first case with dimension 20×20 we obtained the next values, fitness: $e1.8466e + 7$, standard deviation: $1.13e + 3$, power output: $49,913.15$ kw, efficiency: 0,80 and 80 numbers of turbines. For the performance of the algorithm we analyze the number of the evaluations needed, the time execution and memory used during the execution of sequential CHC, this values was 4,836,777 evaluations, 103.09 seconds of execution time and 416 MB of RAM memory used. The configuration of obtained layout for the wind farm is as follows:

5.2 Case *b*: Sequential CHC applied for Wind Farm of dimension 30×30

For the second case with CHC algorithm we obtained the next values fitness: $e3.4857e + 7$, standard deviation: $1.81e+3$, power output: $94,596.12$ kw, efficiency: 0,74 and 192 numbers of

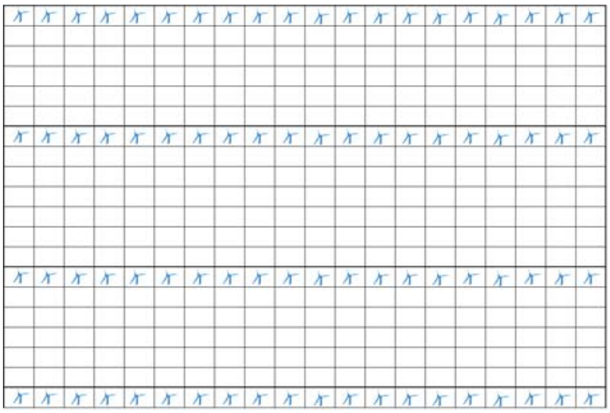

Fig. 4. Scenario 20 x 20

turbines. For the performance of the algorithm we analyze the number of the evaluations needed, the time execution and memory used during the execution of sequential CHC, this values was 4,394,530 evaluations, 256.28 seconds of execution time and 521 MB of RAM memory used. The configuration of wind farm obtained with CHC is as follows:

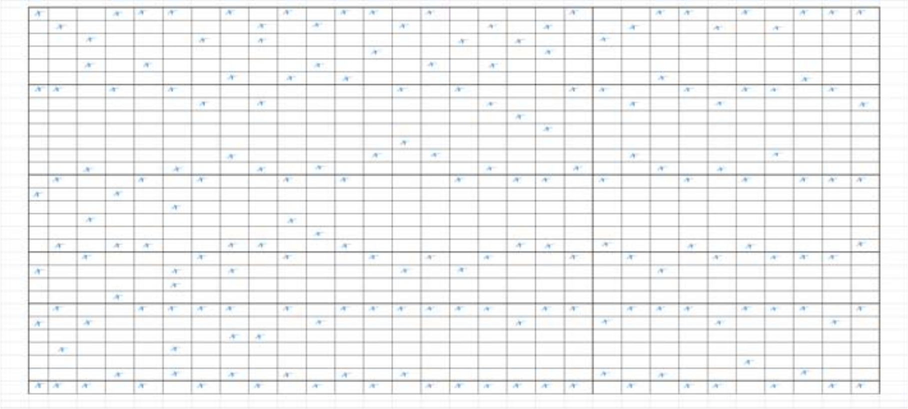

Fig. 5. Scenario 30 x 30 with Sequential CHC

6 Parallel CHC algorithm

In section 3 we explained the generic CHC, now we introduce a new version of CHC, in this case we will divide the population in several small populations called *islands* using a distributed island model[12]. In this model exists several islands that will be executed in different cores. Each population will communicate with a operator called migration. With this operator we send solutions to other population or island to introduce diversity into the population. Each island execute one CHC algorithm with a population size equivalent to *p/n* where *p* is the total population and *n* is the number of island in the model.

For this problem we use the distributed island model with ring connected and we need to define a new operator for work with different island, that operator is called *migration operator* which is responsible for managing the exchange of solutions between islands, for that we define a new operator with this features.

6.1 Analysis of Parallel Algorithms for Wind Farm Design

Now we solve the problem of wind farm design with parallel CHC, in this case we use the model island introduced in the previous section and a multicore processor with eight cores (and eight islands). We will analyze four cases for each instance (20×20 and 30×30), the first case we use only one core with eight island, the second case we use two cores with eight island (four island for each core), the third case we use four cores (two island for each core), and the last case we use eight cores (each island in each core). We study the effort of the algorithm studying the evaluations needed to find a solution, the time execution, the average fitness, standard deviation, average power output, average efficiency, and number of wind turbines. The stop criteria for each algorithm is $5,000,000$ evaluations. We used Mallba library[10] to design the algorithms and we have executed this problem in CPU Multi-Core $2\times$Quad-Core 2.00 GHZ. For migration operation we use the following values: a-synchronized operation mode, period: 10 evaluations and 5 individual of rate.

6.2 Case a:Dimension 20×20 Parallel CHC

For the first case with Parallel CHC algorithm we obtained the values in table 1.

Table 1. Dimension 20×20 for Parallel CHC (average values)

Cores	Fitness(€)	Standard Deviation	Power Output(kw)	Efficiency(%)	Number of Turbines
1	1.8804e+7	98,537.68	50,357.02	0.75	80
2	1.8808e+7	61,096.24	50,575.46	0.75	80
4	1.8813e+7	93,383.71	50,590.40	0.75	80
8	1.8819e+7	99,246.99	50,635.86	0.75	80

We analyze the number of the evaluation needed, the time execution and memory used during the execution of parallel CHC, this values are shown in the table 2.

Table 2. Performance of Parallel CHC (average values)

Cores	Eval. needed	Time Execution (Sec)	Memory used (MB)
1	766,459.97	95.47	339
2	501,392.23	44.65	301
4	438,985.16	23.78	272
8	219,577.60	11.73	208

Is important to note that the performance has improved in all parameters except in the number of turbines respect at the sequential version of CHC. The configuration of wind farm obtained with CHC is as follows:

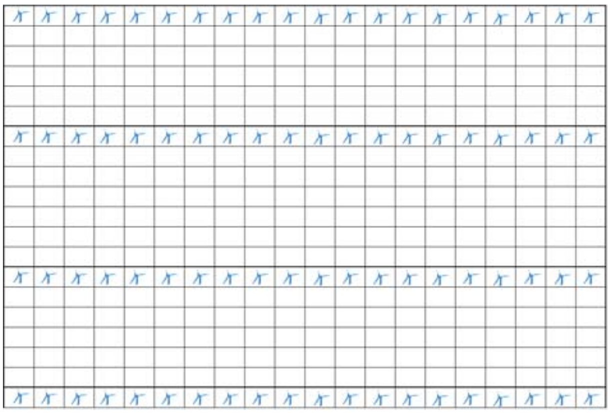

Fig. 6. Scenario 20 x 20 with Parallel CHC

6.3 Case b: Dimension 30×30 Parallel CHC

For the second case with Parallel CHC algorithm we obtained the next values in table 3. We analyze the number of the evaluation needed, the time execution and memory used during the execution of parallel CHC, this values are shown in the table 4.

Table 3. Dimension 30×30 for Parallel CHC (average values)

Cores	Fitness(€)	Standart Deviation	Power Output(kw)	Efficiency(%)	Number of Turbines
1	3.4917e+7	153,384.50	94,930.20	0.75	180
2	3.4954e+7	160,900.72	94,933.22	0.75	180
4	3.4968e+7	133,226.84	94,938.51	0.75	180
8	3.5011e+7	144,202.54	94,948.51	0.76	180

Table 4. Performance of Parallel CHC (average values)

Cores	Evaluation needed	Time Execution (Sec)	Memory used (MB)
1	945,856.97	181.91	461
2	758,258.27	88.61	433
4	637,945.16	43.30	402
8	579,610.77	22.65	395

Is important to note that the performance has improved in all parameters except in the number of turbines respect at the sequential version of CHC. The configuration of wind farm obtained with CHC is shown in Fig 7:

7 Conclusions and Future Work

Test result of those cases study with constant wind speed and direction demonstrate that the performance of our Parallel CHC is optimal for designing wind farms. We have obtained very

Fig. 7. Scenario 30 x 30 with Parallel CHC

good configurations for this problem with parallel CHC and improve our sequential version of CHC. The speed-up obtained is super linear in some cases and good speed-up in the other cases, so we conclude that this version of parallel CHC is a good alternative to design wind farm. As future work we can add new farm models, including actual factors, such as terrain effect and a aesthetic impact, more winds scenarios, real data etc. Also, we intend to use new parallel algorithms to still enhance the performance of the present study.

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 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. M.Bilbao, E. Alba, "GA and PSO Applied to Wind Energy Optimization" . *CACIC 09*, Jujuy, Argentina. 2009.
5. M. Bilbao, E.Alba, "CHC and SA Applied to Wind Energy Optimization Using Real Data". *CEC 10*, Barcelona, 2010.
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. "www.gamesa.es"
8. E. Alba, G. Molina, F. Chicano, "Optimal Placement of Antennae using Metaheuristics" . *Numerical Methods and Applications (NM&A-2006)*, Borovents, Bulgaria. 2006.
9. 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.
10. Mallba Project: *Software Library in Metaheuristic (*1999 − 2002*)*, University of Malaga, University of La Laguna, University Polytechnic of Catalunia. http://neo.lcc.uma.es/mallba/easy-mallba/index.html.
11. Multicore Processor Architecture: "www.intel.com"
12. E.Alba, "Parallel Metaheuristics, A New Class of Algorithms", Wiley-Interscience (September 8, 2005).
13. Larry J.Eshelman, "The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination", Foundations of Genetic Algorithms pp. 265–283 (1991).
14. A Eiben and J Smith, "Introduction to Evolutionary Computing", Springer, pp.XII–287 (2015).
15. J.Serrano González and M.Burgos Payán and J.M Riquelme Santos and F. González-Longatt, "A review and recent developments in the optimal wind-turbine micro-siting problem", Renewable and Sustainable Energy Reviews, pp.133–144 (2014).
16. R.Barthelmie and L.Jensen, "Evaluation of wind farm efficiency and wind turbine wakes at the nysted offshore wind farm",pp.240–243, Wind Energy (2010).