Optimizing the Layout of 1000 Wind Turbines

Markus Wagner
Algorithms and Complexity
Max Planck Institute for Informatics

Kalyan Veeramachaneni
CSAIL
Massachusetts Institute of Technology

Frank Neumann
School of Computer Science
University of Adelaide

Una-May O'Reilly
CSAIL
Massachusetts Institute of Technology

Abstract

In this paper we demonstrate an accurate, efficient, and parallelizable optimization algorithm for the layout of hundreds, then 1000, turbines. It is modular and therefore allows different wake effect models to be incorporated. Its computational cost is a relation which depends upon how many candidate layouts it investigates and the complexity of its wake loss calculation. We demonstrate how well it maximizes energy capture and show how it allows one to examine how wake loss scales with energy capture and number of turbines.
Keywords: Wind farm design, output maximization, wake consideration, Covariance Matrix Adaptation, Evolutionary strategy

1 Introduction

Layout tools attempt to identify the best layout of wind turbines on a land or offshore area according to energy capture. They model free stream wind flowing through an area with sited turbines and calculate the energy output of successive turbines while taking wake effects and turbulence intensities into account. A key component of such tools is the "optimizer" algorithm used to efficiently search through a modest proportion of candidate layouts to identify the best one. With increasing frequency, wind farms are getting larger. For example, the Horse Hollow Wind Energy Center in Texas, USA operates with 735.5 megawatt (MW) capacity and consists of more than 300 turbines spread over nearly 47,000 acres (190 km^2). The layout of turbines in such large wind farms is challenged by large numbers of turbines, large farm areas, constraints on feasible sitings and expensive wake models because the number of siting combinations of turbines on a large area is huge (exponential!), constraints must be respected and the cost of calculating wake loss scales non-linearly with each additional turbine.

In this paper we demonstrate an accurate and efficient optimization algorithm for the layout of hundreds of turbines, in fact, up to one thousand. Such an algorithm could be incorporated as an "optimizer" component choice in a layout tool such as the publicly available OpenWind offering by AWS Truepower. Our algorithm is easily parallelized which makes it faster to use. It is modular which allows different wake effect models to be incorporated. Its cost can be stated as a relation depending on how many layouts it searches through and how expensive it is to calculate wake loss. We demonstrate how well it maximizes energy capture and minimizes the ratio of fixed cost to energy capture. We also calculate its value in calculating how wake loss scales with energy capture as additional turbines are sited.

2 Motivation

This work is primarily motivated by the complexities of placing a few hundreds or even a thousand turbines using a layout optimizer. Some of the complexities are:
Many infeasible solutions: Due to a constraint that

two turbines cannot be closer than 4 times their blade diameter within a specified area, a large number of infeasible layouts exist within the farm area. Traditionally this has been dealt with by dividing the area into cells and deciding whether or not to place a turbine at the center of a cell. However, cells reduce placement flexibility. The challenge is to search through candidate layouts without generating too many infeasible ones, while being able to place many turbines in a flexible manner.

Costly wake modeling: The computational expense of evaluating a candidate layout is high because, as the number of turbines increases, the cost of modeling wake effects for a given layout increases quadratically. While this cost can be somewhat addressed by parallelization, it implies that it is important to intelligently sample layouts so that increasingly better ones are found.

Multiple, Unavailable Optimization Criteria: Layout optimization frequently depends on multiple subjective assessments or criteria that are not provided to an automatic optimizer. To address this, it is useful for the optimizer to produce as many candidate layouts as possible that are of approximately equal value in terms of energy capture. Later, a human designer can review these choices, and use subjective and other reasons not disclosed to the optimizer to choose the best layout.

3 Related Work

Due to its inherent complexity, bio-inspired algorithms such as evolutionary strategies [11] (ES), genetic algorithms [2] (GAs) and particle swarm optimization [4] (PSO) have been used for layout optimization. Recent examples are [5, 8, 9, 10] which are included in [7]. Evolutionary algorithms, which form a sub-class of bio-inspired algorithms, mimic some fundamental aspects of the neo-Darwinian evolutionary process. They simultaneously search with a “population” of candidate solutions and associate an objective score as a fitness value for each one. They then select among the population to favor those solutions that are more fit. The next generation (i.e. a new population) consists of replicates of the fitter solutions which have been “genetically mutated and or crossed over” in a biological metaphor: the decision variables were perturbed such that they inherit some characters of their “parents” as well as change in random ways.

Wan et al, [8, 9, 10], use a cell based approach and compare three different bio-inspired algorithms, each applied to the same set of wind farm models and parameters. They use successively more expressive layout representations (and algorithms)[1] to relax where in a cell a turbine can be located: strictly in the middle, anywhere, or anywhere subject to proximity constraints with neighbouring turbines.

This contribution and Kusiak et al, [5], exploit an alternative to cell placement: each turbine's location is a decision variable pair of real-valued, spatial (x,y) coordinates. With this representation, many more layouts are possible. In [5], a multi-objective evolutionary strategy is used. The secondary objective is to minimize turbine proximity constraint violations. In contrast, our contribution dispenses with the second objective and merely discards infeasible solutions. Kusiak et al do not demonstrate the layout of more than 6 turbines. They confine the model farm area to a 500m radius and cannot identify even one feasible solution in it for a larger number of turbines.

Like Kusiak et al, we employ an evolutionary strategy (ES), albeit a more powerful and competent variant. In general, an ES is effective because it is easily parallelized and it “self-adapts” the extent to which it perturbs decision variables when generating a new candidate layout from an existing one. In the algorithm that Kusiak et al employ, each decision variable is perturbed by adding a normally distributed random value. There, the univariate normal distribution $N(\mu = 0, \sigma)$ is defined by an evolved standard deviation σ that has survived selection and been adapted during the course of the optimization. This implies a strong, but invalid, assumption by the self-adaptation: each decision variable is independent of the others, with respect to the objective, i.e. energy capture of the entire layout. This clearly is not accurate because turbines experience wake interactions and modifying one turbine's location has impact on others.

[1] from a binary to a real-coded GA, to PSO

Table 1: Symbol Definitions

Number of turbines	N
Wind velocity Wind direction	v $0^0 < \theta < 360^0$
Farm radius	r
Rotor diameter	R
Weibull distribution for wind speed Weibull shape parameter Weibull scale parameter Wind direction distribution	$p_v(v,k,c) = k/c(v/c)^{k-1}e^{-(v/c)^k}$ k c $P(\theta)$
Expected power of a single turbine	$E^i[\eta]$
Piecewise power curve of turbine	$\beta(v) = \begin{cases} 0 & v < v_{cut\_in} \\ \lambda v + \gamma & v_{cut\_in} \leq v \leq v_{rated} \\ P_{rated} & v_{rated} < v < v_{cut\_out} \end{cases}$

4 Problem Description

To demonstrate breaking the 1000 turbine barrier with Covariance matrix adaptation based evolutionary strategy (CMA-ES), we formulate the layout problem as follows. Let $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_n\}$ be the x and y coordinates of the n turbines.

Our goal is to identify a layout that maximizes the energy capture from a given farm

$$\underset{(X,Y)}{\arg\max}\, \eta(X, Y, v, \beta(v)) \tag{1}$$

where v is the wind speed, and the function $\beta(v)$, known as a power curve, gives the power generated by a specific turbine for a given wind speed. Wind speed v however is a random variable with a Weibull distribution, $p_v(v, c, k)$, which is estimated from wind resource data. This distribution also changes as a function of direction, θ which varies from $0^0 - 360^0$, yielding a probability density function for different θ given by $p_v^\theta(v, c, k)$. Additionally, wind flows from a certain direction with some probability $P(\theta)$. These different pieces of information are inputs to the algorithm and are summarized in Table 1. Due to the random nature of wind velocity, the objective function in eq. (1) is transformed to evaluate the *expected* value of the energy capture for a given wind resource and turbine positions. For a single turbine, this value can be calculated using

$$E^i[\eta] = \int_\theta P(\theta) \int_v p_v(v(\theta), c(\theta), k(\theta))\beta^i(v). \tag{2}$$

Eq. (2) evaluates the overall average energy over all wind speeds for a given wind direction, and then averages this energy over all the wind directions. However, during the resource assessment, the wind speed distributions are estimated for discrete wind direction bins. Hence the above integral is discretized along the wind direction. Furthermore, the wind speed is discretized in order to proceed with numerical integration. For more details, refer to [5].

4.1 Wake Modeling

The above formulation of expected energy capture, assumes identical wind resources, i.e., $p_v^\theta(v, c, k)$ and $P(\theta)$ at each turbine. However, a significant factor that diminishes efficient energy capture is the wake effect: the so-called “down wind exhaust” from one turbine alters the free stream inflow into a turbine behind it. When optimizing a layout, the wake affect is calculated as a modification of the estimated wind resource that is available for a turbine i due to its location and the location of other turbines. Like others [5], we make some simplifying assumptions for illustrative purposes in this paper. We use the modified Park wake model

[6]. We are aware that, for a large number of turbines, it is not as appropriate as the deep array wake model [1]. The latter could be additionally imposed without modifications to our algorithm. The procedure for the evaluation of the wake effects due to the Park model is shown below in Algorithm 1.

As can be seen from Algorithm 1, the wake effects on a turbine i change the wind resource available to it along different directions by reducing the *scale* parameter of the Weibull distribution estimated for the entire farm, which is also called the freestream wind resource. This is dependent on its location and the location of the rest of the turbines. Hence, eq.(2) is modified to reflect this to

$$E^{farm}[\eta] = \Bigg(\sum_i \int_\theta P(\theta) \int_v p_v^\theta(v, c_i(\theta), k_i(\theta), x_i, y_i, X, Y) \beta^i(v). \Bigg) \tag{3}$$

The goal of the optimization problem is to maximize eq. (3). In the following subsection, we present the constraints and assumptions we made for the optimization problem.

4.2 Constraints and Assumptions

We have the following constraints placed on our optimization function.

Upper bound on the area of the farm: This constraint dictates that we can only place a turbine within a certain area, which is a realistic constraint for most layout problems. For a circular farm with radius r and the origin as the center, this constraint is satisfied *iff* $sqrt(x_i^2 + y_i^2) \leq r, \forall i$. For a rectangular farm with length l and width w this constraint is satisfied *iff* $0 \leq x_i \leq l \,\&\, 0 \leq y_i \leq w, \forall i$.

Proximity constraint: This dictates the minimal distance within which two turbines can be set up. The constraint is satisfied *iff* $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \geq 4M^2R^2, \forall i \forall j$ where R is the rotor radius and M is a proximity factor usually decided ahead of the optimization based on the make and model of the turbines that is used. The equation expresses the proximity constraint as a function of the rotor radius, which is standard in wind industry. In addition to the above constraints, we assume that all turbines have the same power curves (approximated as piecewise linear functions) and that the same wind resource spans the entire farm. [2] The assumptions can be very straight forwardly revised to generate more realistic scenarios.

Algorithm 1 Procedure for evaluation of wake effects due to park model

Given $\{X, Y\}$ as turbine locations $C_T \Leftarrow$ thrust coefficient, $\kappa \Leftarrow$ spreading factor;
$a = 1 - \sqrt{1 - C_T}$, $b = \kappa/R$, $u \Leftarrow$ unit step function, $o = (x_i - x_j)cos\theta + (y_i - y_j)sin\theta$;
$d_{i,j} = \|o\|$, $\alpha = tan^{-1}\kappa$
for $i = 1$ to number of turbines **do**
  for $\theta = 0^0$ to 360^0 **do**
    for $j = 1$ to n-1 and $j \neq i$ **do**
      $\delta_{i,j} = cos^{-1}\{\frac{o + R/\kappa}{\sqrt{(x_i - x_j + (R/\kappa)cos\theta)^2 + (y_i - y_j + (R/\kappa)sin\theta)^2}}\}$
      $Vdef_{(i,j)} = u(\delta_{i,j} - \alpha)\frac{a}{(1 + bd_{i,j})^2}$
    end for
    $Vdef_i^\theta = \sqrt{\sum_j (Vdef_{(i,j)})^2}$
    $c_i(\theta) = c_i(\theta) \times (1 - Vdef_i)$
  end for
end for

5 CMA-Evolutionary Strategy

The Covariance Matrix Adaptation based evolutionary strategy (CMA-ES), summarized in Algorithm 2 including Equations (6) and (8), self-adapts the covariance matrix of a multivariate normal distribution.This normal distribution is then used to sample from the multidimensional search space where each variate is a search variable. The co-variance matrix allows the algorithm to respect the correlations between the variables making it a powerful evolutionary search algorithm. Consider a representation x_k for the k^{th} solution to the optimization problem that attempts to minimize the objective function $f(x)$. In each iteration, t, the algorithm samples λ number of solutions from a multivariate normal distribution given by

$$\mathbf{x}_k^{(t+1)} = \mathcal{N}(\mathbf{m}^{(t)}, (\sigma^{(t)})^2 \mathbf{C}^{(t)}) \forall k.$$

[2] For additional accuracy, these resources can be estimated for different parts in the farm.

Algorithm 2 Covariance Matrix Adaptation Based Evolutionary Strategy

for $t = 1$ to maxiter **do**

Sample $\mathbf{x}_i^{(t)}$ using Equation 5, evaluate energy capture for $\mathbf{x}_i, \forall i$, Select μ members of the population

Update the *mean* using

$$\mathbf{m}^{(t+1)} = \sum_{i=1}^{\mu} \omega_i \mathbf{x}_i^{(t+1)}, such that \sum_{i=1}^{\mu} w_i = 1 and w_i > 0 \tag{4}$$

Update standard deviation $\sigma^{(t+1)}$ using

$$\mathbf{p}_\sigma^{(t+1)} = (1 - c_\sigma)\mathbf{p}_\sigma^{(g)} + \sqrt{(c_\sigma(2 - c_\sigma)\mu_{eff}}\mathbf{C}^{(t)\frac{-1}{2}} \frac{\mathbf{m}^{(t+1)} - \mathbf{m}^{(t)}}{\sigma^{(t)}} and \tag{5}$$

$$\sigma^{(t+1)} = \sigma^{(t)} exp(\frac{c_\sigma}{d_\sigma}(\frac{||\mathbf{p}_\sigma^{(t+1)}||}{E||\mathcal{N}(0, I)||} - 1)) \tag{6}$$

Update covariance matrix using:

$$\mathbf{p}_c^{(t+1)} = (1 - c_c)\mathbf{p}_c^{(t)} + h_\sigma^{(t+1)}\sqrt{c_c(2 - c_c)\mu_{eff}} \frac{\mathbf{m}^{(t+1)} - \mathbf{m}^{(t)}}{\sigma^{(t)}} \tag{7}$$

$$\mathbf{C}^{(t+1)} = (1 - c_{cov})\mathbf{C}^{(t)} + \frac{c_{cov}}{\mu_{cov}}(\mathbf{p}_c^{(t+1)}\mathbf{p}_c^{(t+1)^T}) + c_{cov}(1 - \frac{1}{\mu_{cov}})\mathbf{C}_\mu^{(t+1)} \tag{8}$$

$t \leftarrow t + 1$

end for

where $\mathbf{m}^{(t)}$ is the mean, $\sigma^{(t)}$ is the standard deviation and $\mathbf{C}^{(t)}$ is the covariance matrix for a multivariate normal distribution represented by $\mathcal{N}$. t represents the iteration index. The goal of the algorithm is to then adapt $\mathbf{m}$, σ^2 and $\mathbf{C}$ as optimization progresses. The simplest type of adaptation can be achieved by selecting a subset of μ solutions that perform the best in terms of the objective function, and estimating the parameters of the multivariate normal distribution based on these solutions. This can be simply done using

$$\mathbf{C}_\mu^{(t+1)} = \sum_{i=1}^{\mu} w_i \frac{(\mathbf{x}_i^{(t+1)} - \mathbf{m}^{(t)})}{\sigma^{(t)}}(\frac{\mathbf{x}_i^{(t+1)} - \mathbf{m}^{(t)}}{\sigma^{(t)}})^T \tag{9}$$

More sophistication to this adaptation can be added as shown in eq. (8), such as using weighted sums of the matrices, and the adaptation of the step-size.

Rank μ update: This is summation of two terms , i.e., eq. (9) weighted by $c_{cov}(1 - \frac{1}{\mu_{cov}})$ and $(1 - c_{cov})(\mathbf{C})^{(t)}$. Thus this generates a weighted sum of covariance matrix from previous iteration, and the estimate of the covariance from μ best performing samples in the current iteration.

Cumulation:This captures the direction of the movement of mean as iterations progress. This is calculated using eq. (7) and setting $\mathbf{p}_c^{(1)} = 0$ intially. Note that the contribution of the previous iterations is controlled using a weight $1 - c_c$. $h_\sigma^{(t+1)}$ is a Heavside function that stalls the update of $\mathbf{p}_c^{(t)}$ if $||\mathbf{p}_\sigma^{(t+1)}||$ is large [3].

Step size control: This provides a mechanism to control the variation in $\sigma^{(t)}$. To achieve this an evolution path $\mathbf{p}_\sigma$ is evaluated using eigen decomposition of $\mathbf{C}^{(t)}$, which is $\mathbf{C}^{(t)\frac{-1}{2}}$, and the change in the means. This value is then used to determine the new value of $\sigma^{(t+1)}$ as shown in eq. (6). $E||\mathcal{N}(\mathbf{0}, \mathbf{I})||$ is the expectation of euclidean norm of a $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

Table 3 summarizes the CMA-ES parameters we selected. Initial values are set as follows: $\mathbf{p}_\sigma^{(0)} = \mathbf{0}$, $\mathbf{p}_c^{(0)} = \mathbf{0}$, $\mathbf{C}^{(0)} = \mathbf{I}$. Values for parameters $w_i, c_\sigma, d_\sigma, c_c, \mu_{cov}, c_{cov}$ are set to their default values as described in [3]. For a more detailed intuition about these parameters the reader is referred to [3].

Constraint handling: Our algorithm takes care of the constraints in the following ways. Initially, it places the N turbines on a regular $n \times m$ grid. There, the grid is constructed in such a way that the distance between the rows and columns is maximal, including the placement of turbines on the borders of the wind farm area.[3]

[3] If $n \cdot m > N$, then the last column is not filled completely.

This is a straightforward approach of placing a number of turbines within an area, which can be observed to be frequently used in practice. Additionally, this approach proved to serve as a very good starting point, as the initial distance between the turbines is maximized in a naive way. Thus, the wake effect is already reduced to some extent (when compared to tighter layouts), even without considering the directional distribution of the wind.

Furthermore, when turbines violate the wind farm border constraint, we fix such placements by setting these right back onto the borders. This repair significantly reduces the time that is spent on the creation of a new layout, as the high likelihood of violating this constraint would otherwise require a significant number of repeated trials of creation.

Finally, when a layout has a turbine which violates the proximity constraint, we replace the layout entirely with a new, randomly generated feasible layout which allows the optimization to continue. During our experiments, this circumstance was rare, as the turbines tend to be moved primarily by small distances per iteration.

6 Results and Discussion

We use the wind resources provided in [5]. The wind direction is binned in 15^0 intervals. Scenario 1 has the same *scale* and *shape* parameters for the Weibull distribution for all bins. Scenario 2 has the same *shape* parameter for the bins, but different *scale* parameters. The *shape* parameter k increases the spread of the Weibull distribution as it gets larger. In Scenario 1, the dominant wind directions are 75^0 - 90^0 (with $P(\theta) = 0.2$) and 90^0- 105^0 (with $P(\theta) = 0.6$). There is no wind coming from 0^0- 15^0, and 345^0- 360^0. For the rest of the bins the $P(\theta) = 0.01$.

Scenario 2 is more complex and realistic. The *shape* parameter is the same for all bins, however, the *scale* parameter is different for different bins and ranges from 4 - 10. Similarly, $P(\theta)$ also varies over the range 0.001 to 0.1839. It is more difficult to nominally identify competent layouts as there is no prominent wind direction. In Scenario 1 one can optimize for the prominent directions and not lose significant efficiency. In Scenario 2, one has to optimize the layout to work with minimum wake loss along all the wind directions.

Different metrics to evaluate multiple layouts are presented in Table 4

Table 4: Metrics used to evaluate multiple layouts.

Metric	Definition
$\mathcal{E}_{\mathbf{wlf}}$	Wake loss free power
$\mathcal{E}$	power achieved by layout optimizer
$\mathcal{E}_{\mathbf{loss}}$	Power loss due to wakes
$\mathcal{G}_{\mathbf{N}}$	Power gain achieved by layout optimizer via adding N turbines
$\mathcal{G}_{\mathbf{N}}^{\mathbf{wlf}}$	Wake loss free power capture of adding N turbines
$\mathcal{G}_{\mathbf{N}}^{\mathbf{loss}}$	$\mathcal{G}_N^{wlf} - \mathcal{G}_N$

6.1 Results

Case A: 2-6 turbines: In this case, we validate the accuracy of CMA-ES by showing how it is comparable to small scale results of [5]. Each optimization "run" of CMA-ES evaluated the same number of candidate layouts as [5] for fairness. Due to the stochastic property of ES, we run the ES multiple times and report 'best of runs' meaning the energy loss of the best layout found when all runs are compared and 'average best' which is the average energy loss of the best layout in each run. Using [5]'s scenarios, the plots of Figure 1 show that the Kusiak et al algorithm, called "SPEA-2" [5, 11], and CMA-ES are equivalently effective. We plot the maximum energy capture (before wake loss is subtracted), and the net energy capture (after subtracting wake loss). With 6 turbines, using scenario 2, the energy capture without wake effects would be $43894\ kW$. SPEA-2's layout loses $698\ kW$ to wake effects and CMA-EA's layout loses, on average $440\ kW$ (approximately 36% improvement).

Case B: 10-100 turbines: We choose Scenario 2 because it is a more complex wind resource. Then we attempt to place 10 to 100 turbines at 10 turbine increments in a 9 km^2 rectangular area. Figure 2(left) shows the energy capture under wake loss. Wake loss is indicated by the gap between max energy and energy capture. Figure 2(right) and Figure 4(left) show that the value of adding each additional set of 10 tur-

Table 2: Wind Scenario 1 and Scenario 2

l	θ^l	θ^{l+1}	Scenario 1			Scenario 2			l	θ^l	θ^{l+1}	Scenario 1			Scenario 2		
			k	c	$P(\theta)$	k	c	$P(\theta)$				k	c	$P(\theta)$	k	c	$P(\theta)$
0	0	15	2	13	0	2	7	0.0002	12	180	195	2	13	0.01	2	10	0.1839
1	15	30	2	13	0.01	2	5	0.008	13	195	210	2	13	0.01	2	8.5	0.1115
2	30	45	2	13	0.01	2	5	0.0227	14	210	225	2	13	0.01	2	8.5	0.0765
3	45	60	2	13	0.01	2	5	0.0242	15	225	240	2	13	0.01	2	6.5	0.008
4	60	75	2	13	0.01	2	5	0.0225	16	240	255	2	13	0.01	2	4.6	0.0051
5	75	90	2	13	0.2	2	4	0.0339	17	255	270	2	13	0.01	2	2.6	0.0019
6	90	105	2	13	0.6	2	5	0.0423	18	270	285	2	13	0.01	2	8	0.0012
7	105	120	2	13	0.01	2	6	0.029	19	285	300	2	13	0.01	2	5	0.001
8	120	135	2	13	0.01	2	7	0.0617	20	300	315	2	13	0.01	2	6.4	0.0017
9	135	150	2	13	0.01	2	7	0.0813	21	315	330	2	13	0.01	2	5.2	0.0031
10	150	165	2	13	0.01	2	7	0.0994	22	330	345	2	13	0.01	2	4.5	0.0097
11	165	180	2	13	0.01	2	9.5	0.1394	23	345	360	2	13	0	2	3.9	0.0317

Table 3: CMA-ES and Experiment Parameters. Population is expressed with two variables, μ defines the parent population size and λ the number of offsprings generated from the parent population each generation.

	N=2...9	N=10...100	N=200...500	N=1000
(μ, λ)	(20, 120)	(10,20)	(10,20)	(10,20)
(generations, runs)	(100,30)	(10000, 30)	(10000, 5)	(20000,1)
farm size (km)	r=0.5	l=w=3	l=10 w=20	l=10 w=20

bines slowly declines while the total wake loss rises considerably. The decline may be explained by additional interference due to squeezing more turbines into the farm. The net energy capture and wake loss rise from $73154\ kW$ and zero respectively with 10 turbines to $619133\ kW$ and $112405\ kW$ respectively with 100 turbines. As in the validation case, packing turbines more tightly into the same area creates higher wake loss. Figure 5(left) shows the displacement of 50 turbines from their initial positions at the end of a CMA-ES run. The turbines were placed in a grid initially.At the end of the run the turbines were displaced by a few meters. Figure 5(right) summarizes the displacements of turbines from their initial positions for 30 independent runs of CMA-ES. The turbines moved 15-20 meters from their initial placements.

Case C: Breaking the 1000 turbine barrier: What happens when 200 to 1000 turbines are located in a rectangle of 200 km^2? Figures 3 and Figure 4(right) show, for this turbine range, information similar to that of Figures 2 and 4(left). The net energy capture ascends in sequence $(1440, 2130, 2813, 3465)\ MW$ when turbine number grows from 200 to 500 by 100 turbine increments. The corresponding wake loss sequence is $(23, 64, 113, 193) MW$. At 1000 turbines, the net energy capture is just less than double that of 500 turbines: $6554\ MW$ because the wake loss rises from $193\ MW$ to $761\ MW$. The non-linear trend in wake loss, again, arises from packing more turbines into the same area.

6.2 Cost of the algorithm

One metric of evaluation is elapsed time. We ran a parallelized version of CMA-ES on 20 processors when running large layouts (for 1000 turbine problem). This is crucial because just one wake loss calculation for a 1000 turbine layout takes about 30 seconds (on an Intel Xeon E7530, 1.87GHz). A realistic version of this optimizer would account for many additional details, such

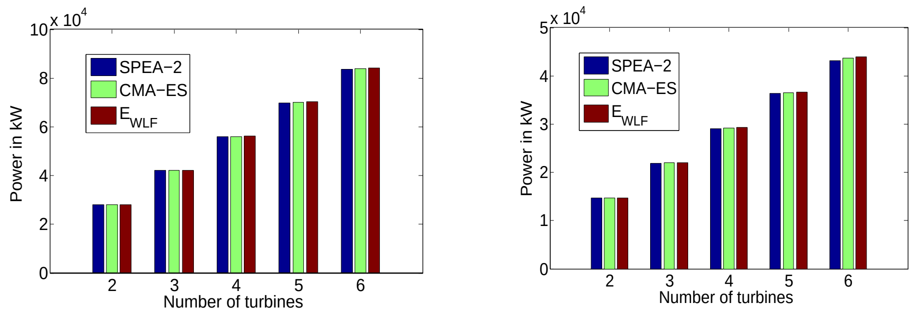

Figure 1: Comparison of Kuisak et al's SPEA-2 algorithm [5] and CMA-ES, left: Scenario 1, right: Scenario 2 . Results are not significantly different and are comparable.

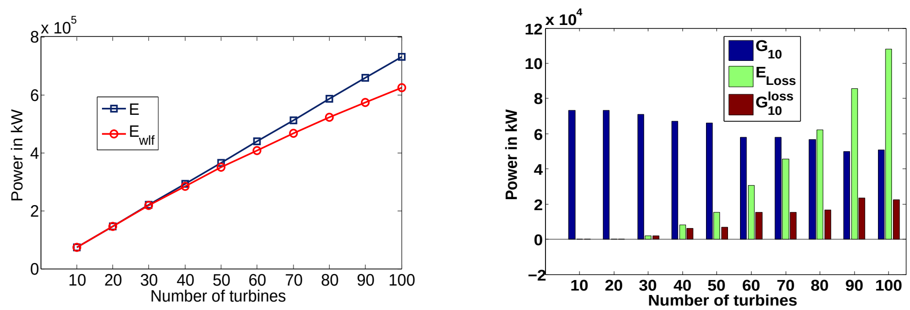

Figure 2: Performance of CMA-ES for 10 to 100 turbines under Scenario 2. Left plot shows energy capture climbs up as we add turbines. Right plot shows how adding each new set of 10 turbines helps despite the increase in wake losses.

as cabling costs, but the dominating factor in expense would still be calculating wake loss when evaluating the net energy capture of a layout as other become asymptotically insignificant. The wake loss calculation scales quadratically with the number of turbines. The cost of one layout evaluation must be multiplied by the total layout evaluations run by the optimizer. For CMA-ES this latter factor is the product of offspring pool size, μ, and the number of generations. For example, each layout of 1000 turbines, on average requires 30 seconds to evaluate net energy capture. If we run CMA-ES with an offspring pool of 20 for 20000 generations so the run requires, serially, roughly 12,000,000 CPU seconds or about 140 days. With parallelization, the elapsed time of the optimization was approximately 12 days implying the speedup is sub-linear. To optimize 200 and 500 turbines serially, it would have taken about 3 and 19 days respectively but, with parallelization on 2 processors, this averaged to 1.3 and 13 days.

7 Conclusions and Future Work

In this contribution, we have presented an advanced evolutionary algorithmic approach that learns the statistical properties of the better layouts and makes use of them to generate even better layouts. This property is advantageous for layout optimization because the optimal position of a turbine depends upon its neighbours'

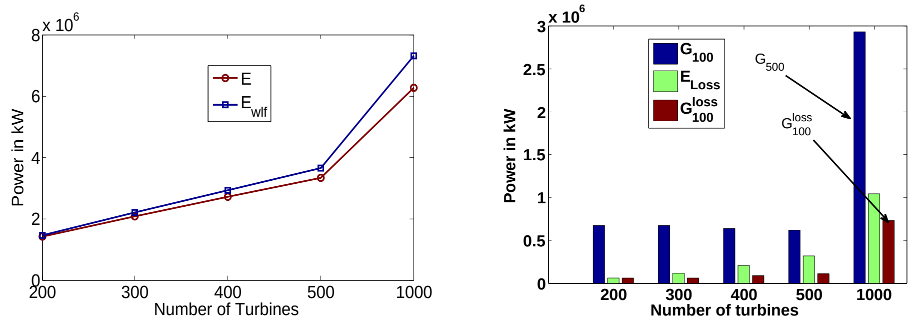

Figure 3: Performance of CMA-ES for 200 to 1000 turbines under Scenario 2. Left plot shows energy capture climbs up as more turbines are added. Right plot shows how adding each new set of 100 (500 between N=500 and N=1000) turbines helps despite the increase in the wake losses.

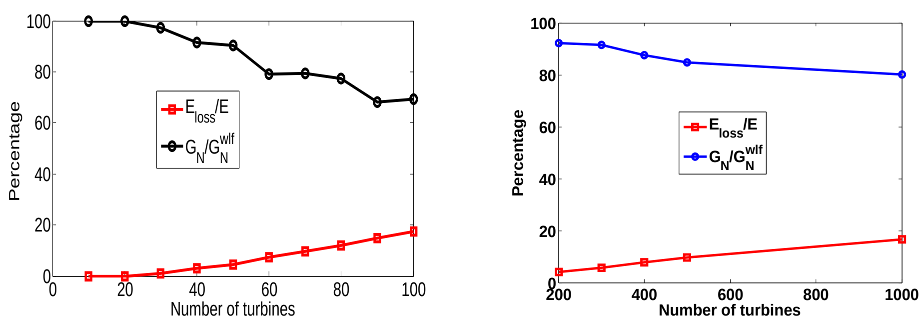

Figure 4: (a)The plot shows how the ratio of energy loss due to wake to total capture increases with each additional set of 10 turbines. As well as the gain achieved by adding each additional turbines starts to decrease. This is characteristic of the layout problem when more turbines are squeezed in the same area. (b) This plot shows the same metric evaluated for layouts consisting of 200 -1000 turbines.

positions. We demonstrated the algorithm on layout problems involving 100's and even 1000 turbines. The stochastic nature of the algorithm demands performing multiple independent trials. One fortuitous feature of this requirement is that the multiple trials provided different layouts that were equally competent in their energy capture. TThey cannot be confirmed as theoretically globally optimal but are competent and useful for practical purposes. The algorithm was parallelized on multiple cores to achieve significant speed-ups. As future work we will focus on the following:

Multiple objectives So far, our focus was on the optimization of a single objective (energy output). A natural next step is to incorporating additional objectives, such as minimizing the required amount of land and minimizing the connecting cables' lengths. These objectives are often in conflict with each other so the goal of solving a such a multi-objective optimization problem is usually to find a set of compromise solutions.

Realistic models More realistic wake models, such as the deep array wake model, will be incorporated. As such models are computationally costly, the value of their precision in accurate energy production prediction will be analyzed.

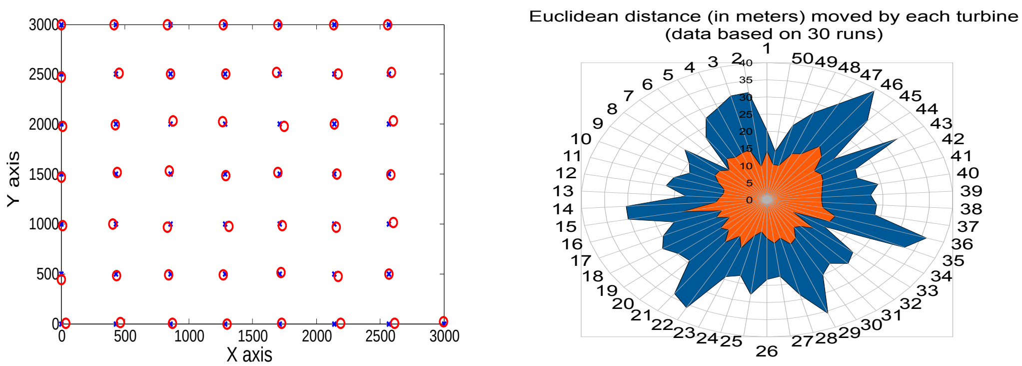

Figure 5: (left) Displacement of the turbines from the initial positions at the end of a CMA-ES run. Mean and standard deviation of displacement of turbines for 30 independent runs of CMA-ES (right)

References

[1] R. Barthelmie and L. Jensen. Evaluation of wind farm efficiency and wind turbine wakes at the nysted offshore wind farm. *Wind Energy*, 13, 2010.

[2] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. MIT Press, Cambridge, MA, USA, 1989.

[3] N. Hansen. The CMA evolution strategy: a comparing review. In J. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea, editors, *Towards a new evolutionary computation. Advances in estimation of distribution algorithms*, pages 75–102. Springer, 2006.

[4] J. Kennedy, R. C. Eberhart, and Y. Shi. *Swarm Intelligence*. Morgan Kauffman, 2001.

[5] A. Kusiak and Z. Song. Design of wind farm layout for maximum wind energy capture. *Renewable Energy*, 35(3):685 – 694, 2010.

[6] H. Neustadter. Method for evaluating wind turbine wake effects on wind farm performance. *Journal of Solar Energy Engineering*, pages 107–240, 1985.

[7] P.-E. Rethore. State of the art in wind farm layout optimization. *Wind Energy Research*, page 179, 2010.

[8] C. Wan, J. Wang, G. Yang, X. Li, and X. Zhang. Optimal micro-siting of wind turbines by genetic algorithms based on improved wind and turbine models. In *Decision and Control, 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 5092–5096, 2009.

[9] C. Wan, J. Wang, G. Yang, X. Li, and X. Zhang. Optimal siting of wind turbines using real coded genetic algorithms. *Proceedings of European Wind Energy Association Conference and Exhibition*, 2009.

[10] C. Wan, J. Wang, G. Yang, and X. Zhang. Optimal micro-siting of wind farms by particle swarm optimization. In Y. Tan, Y. Shi, and K. Tan, editors, *Advances in Swarm Intelligence*, volume 6145 of *Lecture Notes in Computer Science*, pages 198–205. Springer, 2010.

[11] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papailiou, and T. Fogarty, editors, *Evolutionary Methods for Design, Optimisation and Control*. CIMNE, Barcelona, Spain, 2002.