Wind Farm Layout Optimization Problem by Modified Genetic Algorithm

Grant M. Williams
National Weather Center Research Experience for Undergraduates and Oklahoma State University
grant.williams@okstate.edu

October 23, 2014

Abstract

Wind energy is a rapidly growing source of energy for the United States, but there are still technical problems to resolve before it can become the major source of energy production. One of the biggest problems with land based wind farms is minimizing wake-turbine interactions within a constrained space and thus maximizing power. When wind blows through a turbine's blades, a choppy, turbulent wake is created that interferes with the ability of nearby turbines to produce power. Research has already been done on finding ways to model wind farms and place the turbines in a way that minimizes wake-turbine interactions, but current methods are either computationally intensive or require proprietary software. I present a modified genetic algorithm that is able to produce optimized results in a relatively short amount of computation time. The algorithm presented is able to make use of the computation power of graphical processing units and multiple processors and by doing so produces results much quicker than an algorithm run sequentially on a single processor.

1 Introduction

Wind energy is currently the fastest growing energy sector in the United States and has the potential to become a major producer of electricity on par with fossil fuels[7]. The Department of Energy has announced it would like 20% of the power made in the United States to come from wind energy by the year 2030 [7]. It is feasible the United States could reach the 20% goal in the intended time period, but much research still needs to be done to make wind energy a more viable option as a large scale source of electricity. There are a number of technical and societal issues with wind energy; however, one of the most daunting is optimizing wind farms to produce the most amount of power for the cheapest cost given constraints on size of the wind farm and a given budget to construct the farm. When the wind blows through a turbine's blades it creates a turbulent volume of air called the wake. The turbulent air in the wake is unable to generate as much power as a laminar wind, so wind farm companies want to minimize the wake-turbine interactions in order to maximize power production. The computation time to find a layout that minimizes wake-turbine interactions is immense and scales logarithmically with the number of turbines on the farm. Minimizing wake-turbine interactions, and therefore maximizing power, is akin to solving the well-known traveling salesman problem, a combinatorial optimization problem. The wind farm layout problem (WFLOP) is considered to be a Nondeterministic Polynomial Time (NP) Complete problem, which means the time to calculate a solution is $T(N_t!)$ where N_t is the number of turbines and T is a function of time. NP Complete problems are characterized by the fact no fast solution to them is known. With modern computational power, moderately sized versions of NP complete problems can take billions or trillions of years to compute. Past works have looked at finding a heuristic, or approximate, solution to the WFLOP. One heuristic model, the Genetic Algorithm, has been used to calculate the WFLOP, but due to computation limits, past implementations have had to assume a stable wind from a single direction among other simplifications. In this paper the author present a Genetic Algorithm that has been vectorized and modified so that it can run across multiple processors. By using multiple CPUs complex calculations may run simultaneously instead of sequentially, drastically decreasing computation by scaling the time to solve the problem with the number of processors available.

Researchers have taken many different approaches to solving the WFLOP; however, those focused around a heuristic model have produced accurate results in the shortest computation time. Minguez et al[6]. used a local search algorithm with a heuristic global search. An evolutionary model was presented by Kusiak and Song [3]. Eroglu and Seckiner used an ant colony algorithm to model the WFLOP [2]. Mosetti et al. used a genetic algorithm to minimize turbine wake interactions[5].

Section 2 introduces the Wind Farm Layout Problem, how power is produced by wind farms, and power adjustment as a result of the wake loss model. Next the model's parameters are introduced, the discretization of wind speed and direction is explained, and the constraints for the model are presented. In section 4, the optimization problem is presented and the simple and modified genetic algorithms are introduced in detail. The results are presented in section 5 along with the difference between the two cases modeled. Section 6 interprets the data introduced in section 5 and evaluates the models presented against those in literature. Finally, in section 7, the implications of the model are presented and considerations for future work with the model are presented.

2 The Wind Farm Layout Problem

The WFLOP is a complex optimization problem, and some assumptions were made. Enumerated below are the assumptions and parameters used in the model.

1. Wind farm designers often have a set number of turbines planned for a wind

farm, so in this model the number of turbines, N_t, is held constant.

2. The wind farm is a square of dimensions 2047 by 2047 meters. The boundary size of the farm is consistent with real wind farms and the number 2047 was chosen because it is easiest to store the location of the turbines as binary data; however, the model allows for easy changes in the boundary conditions of the wind farm.

3. The location of a wind turbine is defined by its cartesian coordinate after its location has been decoded from the binary population.

4. All wind turbines on the farm share the same parameters (i.e. the rated power, power curve, thrust coefficient, model, hub height, cut in speed and cut out speed remains constant between turbines).

5. For any location, height, or direction on the wind farm the wind speed, v, follows a Weibull distribution modeled by eq(1).

$$p_v\left(v,k,c\right)=\frac{k}{c}\left(\frac{v}{c}\right)^{k-1}e^{-\left(\frac{v}{c}\right)^k} \qquad (1)$$

where $P_v(v,k,c)$ is the probability density function, c is the scale parameter, and k is the shape parameter.

6. The wind direction is based off a probability from data of a nearby location to the wind farm. Where 0°is the eastern most part of the farm and 90°is the northern most portion of the farm. Wind speed is t a function of wind direction θ, therefore $k=k\left(\theta\right)$ and $c=c\left(\theta\right)$ for $0^\circ \leq \theta \leq 360^\circ$.

7. Finally the wind turbines must be placed within the boundaries of the wind farm and no two turbines may be placed within 4 rotor diameters of each other or $\sqrt{\left(N_i\right)^2+\left(N_j\right)^2} \geq 4D, i,j = 1,2,...,N_t, i \neq j$.

2.1 Power Production

The power produced by the wind farm is the total sum of all power from each individual turbine. Kusiak and Song show that the expected wind output of a single turbine can be modeled by eq(2) [3].

$$\begin{aligned}E(P,\theta) &= \int_0^\infty f(v)p_v(v,k(\theta),c(\theta))dv\\ &= \int_0^\infty f(v)\frac{k(\theta)}{c(\theta)}\left(\frac{v}{c(\theta)}\right)^{k(\theta)-1}e^{-\left(\frac{v}{c(\theta)}\right)^{k(\theta)}}dv\end{aligned} \qquad (2)$$

Where $f(v)$ is the power curve of turbine i.

A fully optimized wind farm will produce the maximum amount of power possible by the number of turbines present or $Power = N_t * P_{rated}$. Therefore the objective function of the model can be defined by eq(3).

$$max(\sum_{i=1}^{N_t} E(P_i)) \qquad (3)$$

Finally we can linearize the power curve of the wind turbine with a tolerable error. Eq(4) shows a piecewise function describing the new power curve of the wind turbines.

$$f(v) = \begin{cases} 0, v < v_{cut-in} \\ \lambda v + \eta, v_{cut-in} \leq v \leq v_{rated} \\ P_{rated}, v_{cut-out} > v \geq_{rated} \\ 0, v > v_{cut-out} \end{cases} \tag{4}$$

If the wind speed is below the cut in speed, $v < v_{cut-in}$, of the turbines the power produced is 0, while if the wind speed is between the cut in speed and the rated speed of the turbine, $v_{cut-in} < v < v_{rated}$, the power produced is described by the equation $\lambda v + \eta$. If the wind speed is greater than the rated speed of the turbine but less than the cut out speed, $v_{rated} < v < v_{cut-out}$, then the turbine is able to produce its maximum power, the rated power of the turbine P_{rated}. Finally, if the wind speed is greater than the cut-out speed of the turbine, $v > v_{cut-out}$, the turbine will lock the blades to prevent damage to the turbine and the turbine will produce 0 power. The turbines in the model presented have a cut-out speed of 25m/s.

2.2 The Wake Loss Model

Turbines on an actual wind farm are subject to power loss by wake-turbine interactions. The wake model chosen is the same model described in Kusiak and Song's paper and does not require computational fluid dynamics (CFD) solvers [3]. The wake model was chosen because the it is able to efficiently produce accurate wake interactions [3]. When a laminar flow of wind blows through the first row of turbine's blades a turbulent flow of air, the wake, is created. The wake is a function of the wind direction and speed; therefore, when modeling the wake effects it is important to calculate the wake deficiencies in order of the most upstream turbine to the most downstream turbine with relation to wind direction.

When calculating the velocity deficiency of a single turbine as a result of the wake of a single other turbine eq(5) is used.

$$velDeficit_{ij} = 1 - \frac{v_{down}}{v_{up}} = \frac{1 - \sqrt{1 - C_t}}{(1 + \frac{kd}{R})^2} \tag{5}$$

$velDef_{ij}$ is the velocity deficit created at turbine i by the wake of turbine j. v_{up} is the velocity of the laminar wind flow and v_{down} is the velocity of the wind after the wake effects have been considered. C_t is the thrust coefficient of the turbine, k is the wake decay constant, d is the distance between turbine i and turbine j as a projection along with wind direction and R is the radius of the wind turbine rotor.

The total wake loss at a turbine i is the sum of the individual wake losses turbine i experiences for all turbines i is within the wake of. eq(6) models the total wake loss at turbine i.

$$velDeficit_i = \sqrt{\sum_{j=1, j \neq i}^{N-t} VelocityDeficit_{ij}^2} \tag{6}$$

3 The Computational Model

The wake model described by eq(6) and the power model described by eq(4) and eq(3) were used to simulate the wind farm. Whenever the model calculates the power output of the farm it must first sort the turbines from most upstream to most downstream along a vector projection of the wind direction. Next the model checks to see if turbine i is in the wake of all turbines $j = 1, 2, ..., N_t$

that are upstream of turbine i using eq(7) from Kusiak and Song[3].

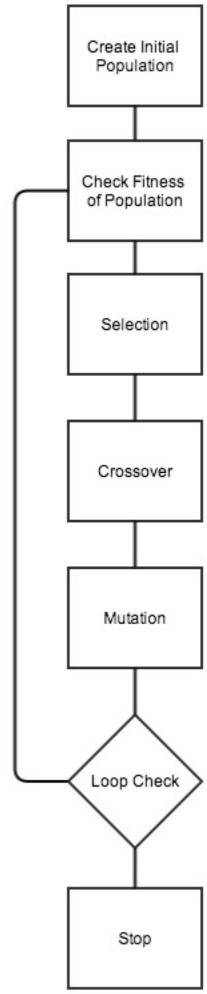
$$\beta_{ij} = cos^{-1} \left\{ \frac{(x_i - x_j)cos\theta + (y_i - y_j)sin\theta + \frac{R}{k}}{\sqrt{(x_i - x_j + \frac{R}{k}cos\theta)^2 + (y_i - y_j + \frac{R}{k}sin\theta)^2}} \right\} \quad (7)$$

if $\beta_{ij} < \alpha$ where $\alpha = arctan(k)$, then turbine i is within the wake of turbine j.

Once the model checks to see if turbine i is within any of the wakes created by the upstream turbines, the wake deficit of turbine i from each turbine upstream is calculated using eq(6) and by using eq(2) and eq(4) we can calculate the power at a single turbine. Finally we use eq(3) to calculate the total power of the wind farm

3.1 The Modified Genetic Algorithm

The genetic algorithm is a form of evolutionary programming that optimizes a function through manipulation of a population. The flow chart below shows an outline of how the genetic algorithm works



Figure(1)

The first step is to create an initial population by randomly filling an G x N_t matrix with binary data. In the case of this paper G, or the row vector referred to as the genome, is of length 22 and N_t equal to the number of turbines. The binary data stored in each row of the matrix is the location of an individual turbine.

After the initial population has been created the fitness of the first population is checked. The fitness function in the model presented is simply the amount of power produced by the wind farm as a whole. To evaluate the fitness function, the binary data in each row of the population matrix is converted to cartesian coordinates, the wake interactions are computed and finally the power of each turbine is calculated and summed as the total power of the farm.

Next the population goes through the selection stage. During the selection stage members of the population are chosen, based on their fitness, to mate with one another. In this paper a roulette selection was used. With a roulette selection the chance a single member of the population will be chosen to reproduce is directly proportional to how fit the population member is compared to the rest of the population. Eq(8) describes the fitness probability of any turbine i.

$$Selection = \frac{FitnessTurbine_i}{\sum_{i=1}^{N_t} FitnessTurbine_i} \quad (8)$$

Next the crossover, or reproduction, stage occurs. During this stage members of the population combine DNA, or bits of data from their genomes, to produce an offspring that has some similarities to its parents. For the model presented a uniform crossover was chosen, as it performed best during initial testing of different crossover types. In a uniform crossover 50% of the data for the offspring is chosen from each of the two parents. A function randomly selects half of the bits in parent 1 and the other half of the bits from are chosen from parent 2. The two groups of bits are recombined to create the offspring's DNA (the binary representation of the turbines cartesian coordinates).

The mutation stage is important because it helps the genetic algorithm to avoid converging on local extrema. During the mutation stage bits are randomly flipped from their current value to the opposite value (e.g. a 1 becomes a 0 or a 0 becomes a 1). The chance for mutation is a predetermined value, in the case of the model presented in this paper 0.25 was used.

During the iterations of the model if the fitness of a population ever comes within the acceptable boundaries set by the user the model will stop, otherwise the model will continue to run until it meets its iteration limit. The model presented is implemented with iteration count being the only stopping criterion.

Finally for the the WFLOP a parallelized and vectorized version of the genetic algorithm was implemented. By parallelizing the wake model of the wind farm, one is able to take advantage of a multi-processor system. Parallelization of the problem allows calculations of multiple parts of the wake deficiency to be calculated at the same time. Vectorizing the code also dramatically speeds up the model by avoiding for loops and instead relying on speedy matrix operations.

3.2 Discretization Of Wind

The wind direction and speed were discretized into small bins in order to cut down on computation time. For the model presented wind was broken down into 16 22.5° increments and wind speed was divided into 20 intervals of 0.5 (m/s). By discretizing the wind speed and direction the model is able to evaluate the power created by using summations instead of a continuous integral.

3.3 Model Parameters

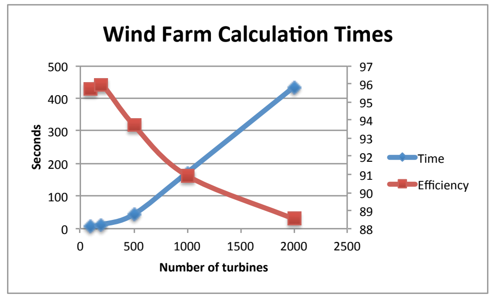
In order to stay consistent, the same parameters Kusiak and Song used were chosen for the model runs. The rotor radius is 38.5(m); cut-in speed is 3.5(m/s); rated speed is 14(m/s); rated power is 1500 (kW); For the linear power curve function, $\lambda = 140.86$, $\nu = -500$. Hub height is 80 (m). The thrust coefficient C_t is assumed to be 0.8, the spreading constant k is assumed to be 0.075 for land cases [3].

4 Results

The model chosen was the modified Genetic Algorithm. The model was run 10 times for a case with 100, 200, 500, 1,000, and 2,000 turbines. The model was run with 100 iterations for each case. The genetic algorithm specific parameters were held constant between runs. The genome length of the population was 22 and the mutation chance was 0.25.

Figure(2) shows a plot describing the difference between efficiency and the time required to computer the model. Efficiency is determined by equation(9) where the $TheoreticalMaxPower$ is $P_{rated} * N_t$, or the $TheoreticalMaxPower$ is defined as the maximum power each turbine could produce at wind speeds of v_{rated} and neglecting any wake interactions.

$$\frac{max(\sum_{i=1}^{N_t} E(P_i))}{TheoreticalMaxPower} \tag{9}$$



Figure(2)

5 Discussion

The algorithm presented was able to produce relatively optimized wind farms for a large number of turbines within small space. The computation time for even the 2,000 turbine case is not unreasonable. The computation time needed to calculate an optimized layout increases rapidly as the number of turbines (N_t) is increased. The algorithm performed very well for the cases involving a large number of turbines; however, given the wind farm size constraints, the algorithm most likely could have found a more optimal layout for the 100 and 200 turbine cases. The randomness of the placement of the turbines means that for 100 or 200 turbine cases it might be possible to achieve 100% efficiency, but the genetic algorithm was unable to do so; however, for the 500, 1,000, and 2,000 turbine cases the genetic algorithm performed very well and produced surprisingly efficient layouts given the changing wind directions. The algorithm seems to have promising potential in real life applications of wind farms. All code was written in Matlab 2013a and run on a compute with a 2.3Ghz i7 Intel processor with 16gb of ram.

6 Summary and Conclusions

A modified genetic algorithm using vectorized code and included a parallelized wake scheme was presented in this paper for an inland wind farm hosting a varied number of turbines. The model presented was able to produce efficient wind farm layouts for a large number of turbines despite the small space the farm was constrained to. The presented algorithm is robust and flexible enough to incorporate more advanced or accurate wake models in future studies. With the current, simplified, wake model, the algorithm was able to find globally optimized solutions within a reasonable time frame. Due to the random nature of heuristic algorithms, the genetic algorithm is some what inconsistent with its layouts; however, the efficiency of the wind farms over 10 runs of the algorithm is very promising.

In conclusion, the algorithm presented was able to produce efficient solutions to the WFLOP for a larger number of turbines than had been presented by other heuristic solutions in literature. Future works on the model might include a varying number of turbines with a second objective to minimize cost of the overall wind farm, or a more complex implementation of the wake model based off of computational fluid dynamics solvers. The model could also be run with considerations to geographic changes in height, or with considerations to terrain with obstacles preventing placement of turbines within the boundary of the wind farm and may produce more realistic modeling for real life wind farms.

7 Acknowledgements

I would like to thank my mentors April Taylor and Dr. Renee McPherson for all of their help and guidance on my project, my fellow REU student Robert Conrick for his help in implementing the wake model, and Dr. Daphne LaDue for her direction in the REU program.

Some of the computing for this project was performed at the OU Supercomputing Center for Education and Research (OSCER) at the University of Oklahoma (OU).

This material is based upon work supported by the National Science Foundation under Grant No. AGS-1062932.

8 References

[1] R. J. Barthelmie, S. C. Pryor, S. T. Frandsen, K. S. Hansen, J. G. Schepers, K. Rados, W. Schlez, A. Neubert, L. E. Jensen, and S. Neckelmann. Quantifying the impact of wind turbine wakes on power output at offshore wind farms. *Journal of Atmospheric and Oceanic Technology*, 27(8):1302–1317, 2010.

[2] Yunus Erolu and Serap Ulusam Sekiner. Design of wind farm layout using ant colony algorithm. *Renewable Energy*, 44: 53–62, 2012.

[3] Andrew Kusiak and Zhe Song. Design of wind farm layout for maximum wind energy capture. *Renewable Energy*, 35 (3):685–694, 2010. ISSN 09601481. doi: 10.1016/j.renene.2009.08.019.

[4] W. E. Leithead. Wind energy. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1853):957–970, 2007.

[5] G. Mosetti, C. Poloni, and B. Diviacco. Optimization of wind turbine positioning in large windfarms by means of a genetic algorithm. *Journal of Wind Engineering and Industrial Aerodynamics*, 51(1):105 – 116, 1994. ISSN 0167-6105.

[6] Beatriz Prez, Roberto Mnguez, and Ral Guanche. Offshore wind farm layout optimization using mathematical programming techniques. *Renewable Energy*, 53: 389–399, 2013. ISSN 09601481. doi: 10.1016/j.renene.2012.12.007.

[7] Kathy O'Dell Ed DeMeo Bonnie Ram Steve Lindenberg, Brian Smith. 20% wind energy by 2030. Technical report, Department of Energy, 2008.