

January 10, 2021

1 Abstract

The 2015 GECCO competition presented an opportunity to apply evolutionary computation to solve a prevalent difficult problem in the field of wind farm optimisation. The computational complexity of modelling the wake effect on wind farm layouts is one issue plaguing the world of sustainable energy generation. In this report we demonstrate how applying various constraints on the representation of the problem, can yield good approximations for real world data. We focus on using the most pertinent data available to apply the maximum number of constraints, reducing complexity while maintaining a viable real-world result.

2 Introduction

The necessity of renewable sources of energy is paramount with our current climate crisis. By current estimates only around 27 % of energy is from renewable sources. This issue will be further exacerbated by the growing energy needs, which are estimated to double by the year 2050 (2). In this report we will focus on developing a model for wind farm optimisation which is a fertile area of research with practical applications that are in demand. We begin with a brief overview of the problem, where we discuss the parameters and simplifications we use. Following this we discuss the literature available on the topic and the competition where this problem was first presented. In the method section we describe the solution to the problem we present, with explanations of our algorithm. We finally conclude with a discussion of our result.

The theoretical efficiency of wind turbines has long been known from the research undertaken by Albert Betz. The calculation below describes the theoretical limit on the extraction of power from a turbine. This calculation is based on conservation of mass and the limit we can achieve is 59.3% conversion rate from kinetic energy in the wind to electrical energy produced. An extension to this topic can be found in the form of the Navier-Stokes equation, which is beyond the scope of this report due to the simplifications we make.

We reduced the problem complexity by considering some simplifications that can be made. We do not consider the economic costs of installing and maintaining any farm design and our approach is mainly based on following assumptions:

- Wind speed is assumed to be the same in the wind farm area and ground is assumed to be flat, which makes every turbine at same height
- The type of turbine used is to be the same for all turbines and each of them has a 127m rotor diameter and 99m hub height.
- Wind turbine location is characterized by its two dimensional Cartesian coordinates (x,y). The number of wind turbines being used is N.
- Wind is following only in one direction and we assume turbines are perpendicular to the direction of wind.

3 Methods

We decided to use a standard representation for our genetic algorithm. We created a class called GAOptimizer which is the main component of our method. GAOptimizer contains the starting conditions of our algorithm and the necessary functions to run the genetic algorithm. We use functions to create an initial population of solutions, which consist of arrays with entries that represent the presence or lack of turbine at the selected position. We apply a fitness function that calculates the total power output of any specific layout. The fitness function also considers the interplay that the wake effect has on the layout, as the presence of wind turbines changes the dynamics of the result. Therefore our fitness function is representative of a real world scenario, where a wind farm layout attempts to maximise the power output and minimize the loss due to wake effect. With this we can come to an evaluation and comparison of the fitness of any particular set of solutions.

We can then begin to work on a process to iterate through generations of solutions. We implement parameters and functions for mutation, crossover rate as well as elite selection. With these parameters we can extend the search space for possible solutions and decrease chances of being stuck in a local sub-optimal solution. With the parameter for elitism along with the fitness function, we create selection pressures that steer the solutions along maximising the power generated. We can then apply this method to run the proves for any number of iterations we require. We use a second class named Turbine which contains some real-world data we use for the calculations. In this way even when using a simplified model, we obtain reasonable results.

We use an API to retrieve information on how the wake effect impacts our solutions. As the gaussian effect is not a simple scalar value that effects all turbines equally, therefore we must consider how to deal with a complex problem from the realm of fluid dynamics. We use a simplified version known as the Bastankhah Gaussian wake model for the calculations. We calculate the estimated losses from the wake effect for each turbine downstream. Each turbine creates it's own wake effect therefore we iterate through this process to consider

all wake effects. One simplification which makes this process easier is our initial assumption of the wind only having one direction of travel.

4 Results

Graph: y = power output x = generation number

Possibly show two pictures for results: Layout of first generation + Layout of last generation

This way we can compare how the algorithm performed over time

5 Discussion

Compare our results to competition results, see how good our simplified model is compared to the more complex ones

How good is a simple genetic algorithm at increasing quality of our solutions?

Are the results of a more complex model more reliable and realistic?(probably)

What improvements could we make?

6 Conclusion

overall review of everything above, how we completed the problem

possibly suggestions for improvements if we had to work on this problem further, what blindspots are there we could fix?

7 Literature Review

We can also only include paper 1, and call a small section "competition overview" instead

Paper 1 (overview of the original 2015 competition):

<https://www.sciencedirect.com/science/article/pii/S096014811830363X>

In the original competition, we are presented with minimising the cost function which considers many parameters that have an impact on the economic viability of a wind farm. The authors also reward solutions with high number of turbines, to counteract the possibility of small layouts having good theoretical efficiencies but poor power outputs. Across all the different scenarios, the 3-stages memetic differential evolution (3s-MDE) approach proved to produce the best results, outperforming on four of five tested scenarios. This approach creates a surrogate model that creates a candidate solution, which is evaluated by the fitness function, starting with two initial turbines on the first instance.

A range of predetermined distances between turbines is considered to calculate the effect of adding additional turbines to the solution, as their presence will influence the power generated. This method provided significant improvements, especially compared to the GA described in the paper. When considering standard methods of optimising placement of each individual turbine placement the competition results presented provide a superior method of solving for this complex problem.

Paper 2: <https://iopscience.iop.org/article/10.1088/1742-6596/1037/4/042012>
(the paper for the gaussian wake)

In the following paper we are presented with a representation of the wake effect. The wake effect has a large influence the total energy production from wind farms. This is due to changes in wind speed as wind interacts with a turbine. While the flow of wind over a turbine occurs, perturbations and changes in direction of the wind occur. These perturbations have a significant impact on the ability of turbines downstream to collect wind energy. Wind turbines already have an inbuilt theoretical limit on collecting the kinetic energy from wind and turning it to useful electricity, in the form of Betz's Law. The authors propose a way to calculate this effect in equation [6]. The influence of this paper can be seen in our GaussianWake function in the WindFlo API, which provides a representation of this calculation. Paper 3 (may not include due to 4 page limit for this report): Possibly paper 4 (may not include due to 4 page limit for this report)

[heading=bibnumbered]