**Wind farm layout optimisation using a genetic algorithm**

Arif Malik, Bharath Shivamogga-Jairam, Ryan Shaw, Mingting Hong, Filip Tobolewski

**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.

1. **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 (1). 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. Our approach is based on some of the following assumptions:

* Wind speed is assumed to be the same in the entire 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). And 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.

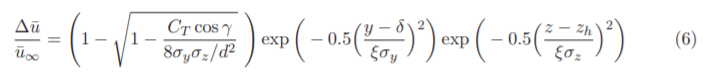
1. **Method**

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.

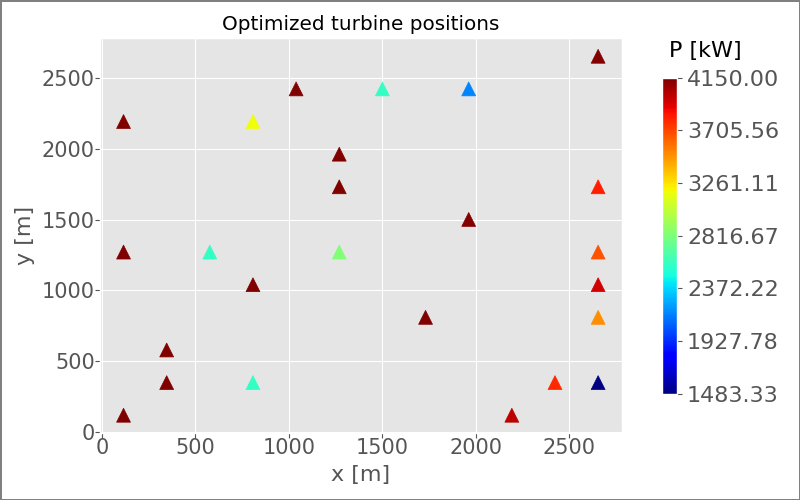
The mathematics behind the wake function used is largely described in this 2018 paper(reference). While the flow of wind over a turbine occurs, perturbations and changes in direction of the wind occur. This effect has a huge influence on the total energy production. These perturbations have a significant impact on the ability of turbines downstream to collect wind energy. Wind turbines have an inbuild 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 below. The influence of this paper can be seen in our GaussianWake function in the WindFlo API, which provides a representation of this calculation.



1. **Results**

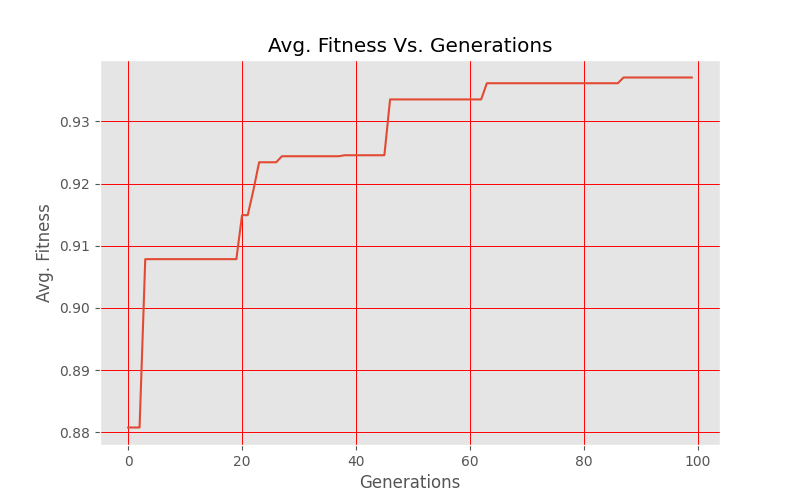
**4.1 Turbine placement solution**

The results of our algorithm are promising. On the optimized turbine position figure we can see the final turbine positions of our solution. We can see that some of the turbines are producing close to their theoretical limit. We can see the central turbine producing around 98% of it’s potential power output. We see that we can largely group the type of placement into two categories.

1. Turbines maximised for distance from other turbines, thus reducing the wake effect
2. Turbines placed in clusters, sub-optimal individual placement but good use of space available for maximizing average power output

Our solution is compromised of the two types of turbine placements defined above. The solution settled on a mixed approach, having some individual turbines of type 1 producing high power output. Other spaces of our solution compromise of clusters as seen in the bottom right and left of the diagram. The placement of these clusters will be largely dictated by maximising distance from other turbines and attempting to utilise as much of the field as possible. This is due to the algorithm recognising the need to minimise the wake effect as much as possible to produce good solutions.

The wind is defined in the -x direction. We see the presence of high-power producing turbines across the entire field. As we look the at results from right to left, we see that we have good turbine placement across the entire field. High power production does not seem to be location specific as this layout does very well to reduce the wake effect.

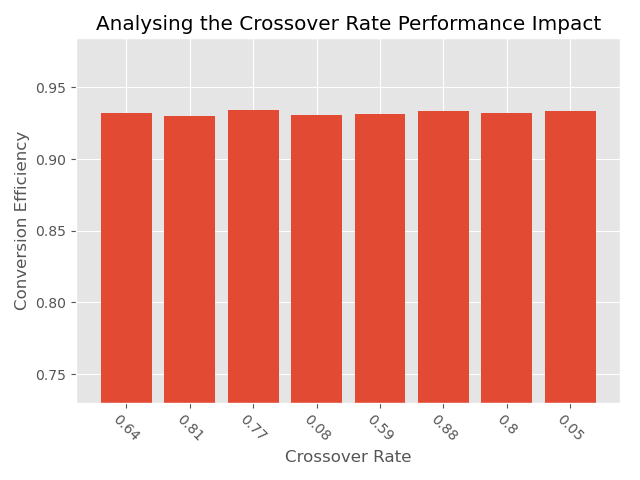
**4.2 Genetic algorithm over generations.**

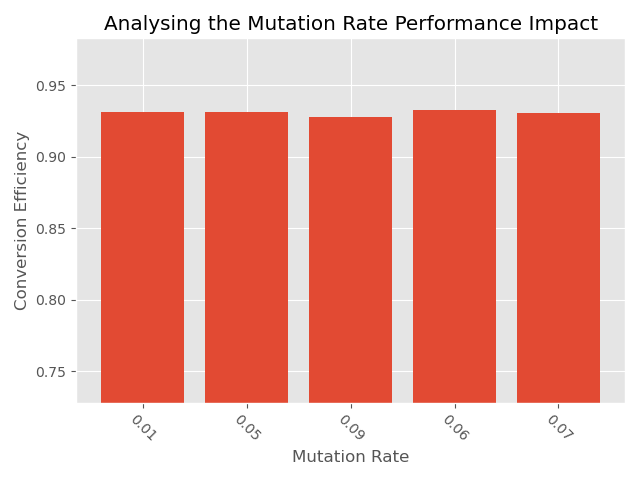
We can explore the value of the fitness function across the generations. We choose 100 generation so that we get result in reasonable computational time. The fitness function value begins at a high level and climbs to the maximum value in around 60 generation. The results of the genetic algorithm functions have the desired effect and we obtain a very good value for the fitness function.

For reliability we conducted 10 trial runs to ensure our results were consistent and not a result of luck.

1. **Discussion**

The problem described can have a huge scope as incorporating real world complications can increase the computational complexity significantly. We have shown that considering the problem in a simplified manner can achieve a good level of result.

We can experiment using different parameters, such as changing the crossover and mutation rate. This will have the effect of increasing the search space of solution at the cost of increasing computational resources. We first look at the effects of the choosing different crossover rates. From 8 trial runs we see that the crossover rate has little impact on the conversion efficiency.

We see that the change of the mutation rate also has a small impact on the conversion efficiency. One improvement we could make for these experiments is to increase the number of generations we use. Doing so would increase computational time therefore we decided against this as the results obtained are satisfactory.

Another extension we could consider is multidimensional wind directions. In our current model we reduce the wind to being single directional therefore it will always be optimal to have a large spaced out ‘front line’ of turbines that are very productive. Considering many different direction would improve the resulting layout.

* 1. **Particle Swarm Optimisation**

In PSO, a potential turbine layout is represented by a particle and a swarm is essentially identical to a GA population. A particle has a velocity that is updated each iteration based on its previous velocity, social weighting (global best), and cognitive weighting (personal best). The velocity then affects the position of the particle for each iteration, where its position corresponds to the parameters used for the cost function. However, we was unable to proceed as we could not figure out how to represent the search space in terms of turbine layouts.

1. **Conclusion**

The need for research in wind farm optimization is apparent. Key societal and ecological issues are at stake if we do not act now. Further research is pivotal in attempting to find solutions to these complex problems. We discuss the theoretical limitations of wind turbines from the basic properties of physics and move onto explaining the simplifications we make. In the method we see the logical structure of how we approached the problem and how we attempt to solve it.

A simple genetic algorithm approach for a wind farm optimization problem can have successful results. With our simplifications the model maintains the ability to give us good solutions as well as remain computationally efficient. In our results section we see the layout of turbines we generate and how the fitness function improves over the course of generations. We found that we can consistently derive results with around 0.93 fitness as defined by our fitness function showing a very good results as defined by the scope of our task.