

# **APPROACH EXPLAINED**

## **Shell.AI Hackathon Wind Farm Layout Optimization Challenge**

***Team Name:* Team-7**

### ***Team Members***

**Akash Waitage**

**Awanit Ranjan (Team Leader)**

**Abhishek Choudhary**

**Vivek Kumar**

***College:* National Institute of Technology Karnataka  
Surathkal**

### ***Abbreviations:***

**WFL**-Wind Farm Layout

**GA** - Greedy Algorithm

**AEP**-Annual Energy Production

**Pec**-Perimeter Constraint

**Prc**-Proximity Constraint

**Problem Statement.** In this hackathon, the challenge is to optimize the placement of  $N$  turn , 50 wind turbines of 100 m rotor diameter and 100 m height each on a hypothetical 2D offshore wind farm area such that the AEP (Annual Energy Production) of the farm is maximized. The farm area for this problem is square in shape having dimensions: length  $L_x = 4$  km, width  $L_y = 4$  km. The

orientation of the farm is such that one of its edges is parallel to the geographical North direction.

*There are two constraints that must not be violated by a wind farm layout to be considered valid.*

**Pec** - All the turbines must be located inside the perimeter of the farm, while maintaining a minimum clearance of 50 meters from the farm Boundary.

**Prc**- The distance between any two turbines must be larger than a given security threshold to ensure a longer lifetime of the turbine rotors. This the minimum distance between two turbines is called  $D_{min}$  and needs to be 400 m.

### **List of Assumptions.**

- Identical Turbines
- Homogeneous Free Wind Distribution.
- No Partial Wakes
- Wake Modeling Assumptions

### **Data Provided:**

We have provided with different types of datasets: (i) wind data, (ii) turbine power curve data, and (iii) a turbine locations data (for testing).

# Our Approach:

As our sole goal is WFL Optimization thus first thing which came in mind was to move with Greedy Algorithm (GA). (Why Greedy is explained in the following section.)

## Detailed Explanation:

The idea was to divide Wind Farm continuous space into discrete points. So let's say we divide the complete Wind farm area into  $n$  points. Where it has  $t$  points in each row and there are  $t$  such rows( i.e a square) So we have  $t*t=n$  point. With each point separated by its adjacent point by a minimum distance of  $d$ (say).[ $d$  later will be treated as a hyper-parameter for tuning our grids points or making our grid more or less dense.]

Let  $x \in \{0, 1\}^n$  (As grid will be a two dimension so reducing its dimensionality by going in row-major form ) be a vector of binary variables indicating turbine locations among  $n$  possible points in the wind farm.

The goal of the proposed idea was to select  $N$  feasible turbine locations from  $n$  possibilities such that resulting layout is maximized for required AEP. The directional power output of the wind farm(AEP), was considered as an objective function taking into account the Wake effect following Jensen's Model.

But how do we guarantee Prc and Pec constraints?

As stated above we have to choose  $N$  points from  $n$  possible points for such that our objective function is maximized. So at each time during selection of points we are keeping a turbine at a distance of at least of safety distance(400m) with the rest of the turbines which are already selected.

Putting in another way suppose we have selected  $x$  points among  $n$  plausible points then our next selection of  $x+1$  th point will be based on two criteria first it should be at a distance greater than safety distance allowed between turbine from all  $x$  points selected so far. And the second criteria follows only after the first one and which is that placing that  $x+1$  th turbine at that points should yield maximum power for  $x+1$  turbines from rest of the  $(n - x)$  points.

Now this guarantees to give an optimal solution satisfying Prc constraint.

But how do we guarantee Pec ?

This was a cakewalk for us as we only have to consider points of the grid between  $(0+50)$ -  $(4000-50)$  i.e  $[50-3950]$  for both  $x,y$  coordinates.

Why do we feel this approach was correct?

Let assume  $G$  to be one of the optimal set of  $N(=50)$  points for turbines .  
Let  $H$  be the set of  $N$  points which we got by Greedy. Now let's start from the initial state .

Let assume that our starting point  $s_i$  in  $H$  (which we choose randomly as there will be no wake effect for  $N=1$  and also rotors of turbines can move ) belongs to  $G$  so upto now we are at optimum .

Hypothesis:

Now if  $N$  was 2 then then our second point of choice was such that it is placed greater than or equal to the safety distance and maximizes our output so this point should be in  $G$  else  $H$  will be optimum.

So by Mathematical Induction let's assume this is valid for  $k$  point i.e our  $k$  points in  $H$  matches with that of  $G$  where  $k < N$ .

Now for the next turbine's point that is the  $K+1$ th turbine point be  $(x,y)$  (say) we have again chosen a point based on greedy which maximizes our output[following Pec and Prc constraints]. So again by hypothesis if this  $(x,y)$  did not belong to  $G$  then let  $K+1$ th point that belongs to  $G$  be  $(x',y')$ . Now as we got  $(x,y)$  as point to be qualified candidate for producing maximum output (AEP) say  $A$  but  $G$  has  $(x',y')$  producing output power(AEP) as  $A'$  so if  $A' < A$  then obviously our  $H$  will becomes optimum but this will be a contradiction to our assumption that  $G$  is optimum. If  $A' > A$  then greedy will obviously choose  $(x',y')$  instead of  $(x,y)$  so this possibility is ruled out thus we are left with  $A' = A$  in this case it may be possible that both points  $(x',y')$  and  $(x,y)$  are producing power(AEP) output same upto a scale  $10^{-6}$  or  $10^{-7}$  so in this case too  $H$  will also be one of optimum solutions.

So by strong form of mathematical Induction we conclude Greedy will indeed give optimum solution if the first chosen point belongs to the optimum set.

Thus our job was tailored to tuning of hyperparameters (d) min distance in grid between adjacent points responsible for denseness of the grid and the starting point .

At later stage we thought of the possibility if we choose the optimum point then what will happen in the case when we run short of the area for finding

the rest of the points but fortunately this will not be the case here because for accommodating 50 turbines we have given the area which was very much larger so during the running of code each time we always get a point such that we are capable of putting rest of the points i.e we never run short for the area.

We feel that our approach was of a scalable nature if the size of the wind farm is increased or even if value of number of turbines( $N$ ) is increased (but in this case if we are running out of the area then we have to consider a condition which is discussed in the above paragraph which simply was to place the point in such a way that it follows the three conditions the first condition is that the point selected should be more than the safety distance from the rest of the points selected earlier the second condition is that point selected it will be in such a way that in future we are capable of putting rest of the points and the third and the last condition would be that placing the turbine at this point produces (or optimises or maximize our objective function.) so in this case too this procedure works well provided that wind nature is not random, the terrain is smooth and flat.

Now during running of code in beginning and mid phase of Hackathon We face some problems, that is the code takes 2-3 ,5-6 and even more than that to produce the output depending on how dense the grid is selected.( that is the hyperparameters which we are talking about in our paragraphs so far) .

Thus for solving this problem by a lot of discussions among the team members we make a next move that first we will run the code or train our model for dense points so that the points which we will get after the result will be analysed on the basis of hyper parameters which was given to the code for tuning on the basis of this result we make an important observations that at each tuning process the points get clustered around the boundary and in the middle region there was some sort of pattern and also by having online submissions on Hackerearth platform we get to know that average wind nature for online dataset is very much similar to that of

average wind nature for the years 2007, 2008 and 2017 for the dataset provided to us so clubbing all these observations we develop a strategy of not to scan the whole area of the farm for selecting the points but rather a part of it (we can say this as a divide and conquer strategy) i.e we will scan in the region it's more likely to be placed in optimum case (this observation was based upon our details analysis on output of several results in which different hyper-parameters were given as input as discussed above in this paragraph ) this takes less time to produce output and also its increases our mean AEP too. So we run loops taking each point and optimizing it in a certain small area of its likelihood ness. Same procedure is done for rest 49 points and this will be done for several iterations until the gradient of power (increase in objective function values) starts decreasing monotonically . Thus at that point code will be stopped and files having coordinates of 50 points with max- output (MAX AEP) will be saved.