Shell.ai Hackathon for Sustainable and Affordable Energy

Windfarm Layout Optimisation Challenge

COMPETE, CODE AND WIN!

REGISTER NOW

shell.in/hackathon

Contents

1 **Introduction** 1

2 **Problem Statement** 1

2.1 **Problem Formalization.** 3

2.2 **List of Assumptions.** 3

3 **Material Provided** 4

3.1 **Data** 4

3.1.1 **Wind Data** 4

3.1.2 **Power Curve** 4

3.1.3 **Test Turbine Locations** 6

3.2 **Wind Farm Evaluator** 6

3.2.1 **A Note About Code Alteration** 7

4 **Wake Effect Modeling** 7

4.1 **Wake Combination.** 9

5 **AEP Algorithm** 9

6 **Solution Submission and Evaluation Criteria** 11

7 **References** 12

1 Introduction

The energy transition and digitalisation are two mega-trends that will affect the world in the coming decades. A planet with more people and rising living standards will need more and cleaner energy solutions. We must reduce carbon emissions to tackle climate change. Shell believes that digitalisation and AI are critical enablers to support our ambition to be a net zero energy company. We see great potential in producing renewable power from wind. This hackathon, powered by Shell.ai and, provides an opportunity to work on a challenge commonly met with wind farm layout designs: how to find the most optimal and profitable layout of turbines inside a wind farm.

Optimal layout of wind turbines in a wind farm carries huge business importance. An unoptimized or suboptimal layout can typically cost upto 5-15 % of AEP (Annual Energy Production) [1], which eliminates the business case. Simultaneously, it also has the potential to steer the energy portfolio further towards sustainable and cleaner energy.

The key problem with an unoptimized layout is the combined influence of arranged wind turbines on the wind speed distribution across the limited area of farm. As a wind turbine extracts energy from the incoming wind, it creates a region behind it downstream where the wind speed is decreased – *wake region.*

> *Note that wind turbines automatically orient their rotors to face the incoming wind from any direction.*

Due to the induced *speed deficit*, a turbine placed inside the wake region of an upstream turbine will naturally generate reduced electrical power. This inter-turbine interference is known as *wake effect.* In order to deal with it, strategies to layout a wind farm in an optimal fashion should be employed such that the power losses incurred due to the combined wake effect is minimum.

Optimizing the layout of a wind farm is an interesting and complex optimization problem. The key challenge arises due to the high dimensionality, complex multimodality and discontinuous nature of the search space. Hence, optimizing the layout analytically is impossible. A smarter approach towards solving this problem is through optimization strategies and computer algorithms.

2 Problem Statement

Before we present the problem statement, we would like to mention clearly that in practice wind farm layout optimization is a complex problem, and in here we are only presenting a simpler version to test numerical optimization capabilities. We specify the assumptions and simplifications at suitable places in this text.

Problem Statement. In this hackathon, the challenge is to optimize the placement of N_{turb}, 50 wind turbines of 100 m rotor diameter and 100 m height each on a hypothetical

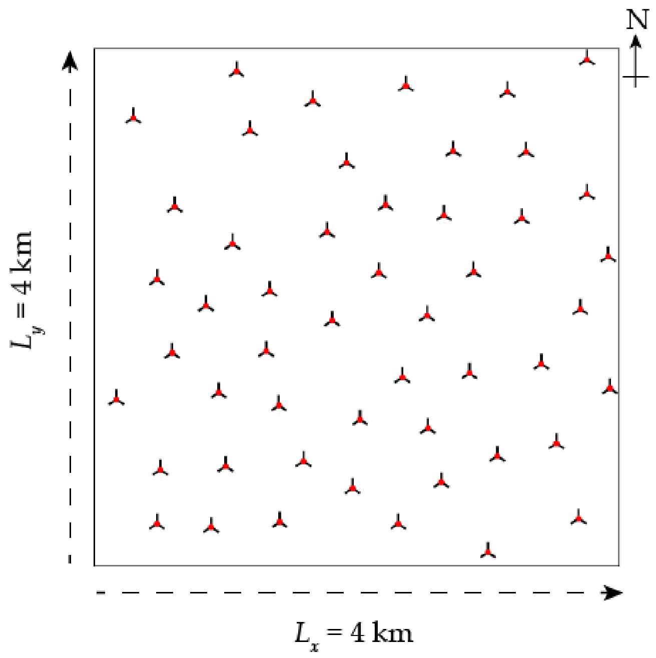

Figure 1: A valid example layout.The orientation of farm is such that one of its edges is parallel to the geographical North direction. Here it can be assumed that the positive y and x axis shown are aligned with the geographical North and East directions respectively. This diagram is only for illustration. Here, the turbine rotor size does not represents the actual scale.

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 farm is such that one of its edges is parallel to the geographical North direction. See Figure 1. There it can be assumed that the positive y and x axis shown are aligned with the geographical North and East directions respectively.

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

- *Perimeter Constraint.* All the turbines must be located inside the perimeter of the farm, while maintaining a minimum clearance of 50 meters from the farm boundary.
- *Proximity Constraint.* The distance between any two turbines must be larger than a given security threshold to ensure a longer lifetime of the turbine rotors. This minimum distance between two turbines is called D_{min} and needs to be 400 m.

Submissions failing to satisfy the above constraints get automatically rejected.
A valid example layout is shown in Figure 1.

2.1 **Problem Formalization.**

Although, a formalized mathematical formulation of this optimization problem is not necessarily required here, we still provide it below for assisted understanding.

Let us first denote the xy coordinates of a turbine placed at a position i by (x_i, y_i) (units in meters), $i = 1, 2 \dots N_{turb}$.The goal is to:

$$\text{maximize}(AEP),$$

,given the following constraints:

$$x_{\min} \leq x_i \leq x_{\max}, \quad y_{\min} \leq y_i \leq y_{\max} \quad i = 1, 2 \dots N_{turb},$$

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \geq D_{\min} \quad i, j = 1, 2 \dots N_{turb} \quad \text{and} \quad i \neq j,$$

$$,$$

, where AEP denotes Annual Energy Production of the farm after accounting for the wake effect among turbines. As a result of perimeter constraint, $x_{\min} = y_{\min} = 50$, $x_{\max} = y_{\max} = 3950$. N_{turb} is the total number of turbines and D_{min} is the minimum distance that any two turbine pairs must maintain. As mentioned before, N_{turb} is 50 and D_{min} is 400 m for this problem. The method for calculating AEP is presented in section 5.

2.2 **List of Assumptions.**

Towards solving this problem, you are allowed to make the following assumptions:

- **Identical Turbines.** You are required to assume that all the turbines are identical and have same specifications i.e. turbine type, turbine height, rotor radius, rated power, rated wind speed, power curve, thrust coefficient etc.
- **Homogeneous Free Wind Distribution.** If there is no turbine placed, and hence no wake effect, then at any given time, the wind can be assumed to be equally distributed all over the farm area i.e. at any location within the allocated farm area, the wind flows with same free stream speed and in the same direction.
- **No Partial Wakes.** Turbine rotor center (or turbine coordinates) represents effective turbine locations. We ignore the wake effect if location of a target turbine does not falls inside the wake region of a wake producing turbine.
- **Wake Modeling Assumptions.** Standard assumptions of *Jensen's wake model* are made for modeling the inter-turbine wake effect.

3 Material Provided

For this challenge, we provide two sets of materials: *(i)* the required data, and *(ii)* codes for evaluating AEP.

3.1 Data

We have provided three different types of datasets: *(i)* wind data, *(ii)* turbine power curve data, and *(iii)* a turbine locations data (for testing). We describe these data sets below.

3.1.1 Wind Data

First and foremost, we have provided a stack of seven `.csv` files containing approximately half-hourly wind data gathered from an anonymous location for seven different years i.e. 2007, 2008, 2009, 2013, 2014, 2015 and 2017. The names of these files are suffixed with the corresponding year - `wind_data_<year>.csv`. The wind data was collected at 100 m height from MSL (Mean Sea Level). Note that the turbine height for this problem is also 100 m. Inside these files, there are 3 columns:

- `date` - date, time of data recording
- `drct` - direction towards which the wind flows, measured in degrees. North (0° or 360°), East (90°), South (180°) and West (270°). Please note that entries in this column are multiples of 10 i.e. 360°, 10°, 20°. . . . 340°, 350°
- `sped` - wind speed in meters per second. Range of data [0, 30].

> *Note that the provided wind data is 'model-ready' and there is no requirement for any kind of pre-processing or data cleaning steps.*

The usual method to visualize a wind distribution data is by creating *wind rose diagrams*, with appropriate size of wind speed and wind directions bins. Creating a wind rose diagram is simple and some information is provided here about how to create one.

The wind rose diagram for the data in the file `wind_data_2007.csv` is shown in Figure 2. The way to interpret this diagram is straightforward. Roughly 3% of the total observations have `drct` values (270°). So on and so forth for the other directions. Within a particular direction sector, segment lengths are 'chunked' out depending on the percentage of wind speed observations that fall out in a particular speed bin (similar to stacked histogram). Colored according to the legend on the right.

3.1.2 Power Curve

We next provide a file `power_curve.csv` that contains the turbine's information about its electrical power output and *thrust coefficient* values at different wind speeds. There are 3 columns in the dataset:

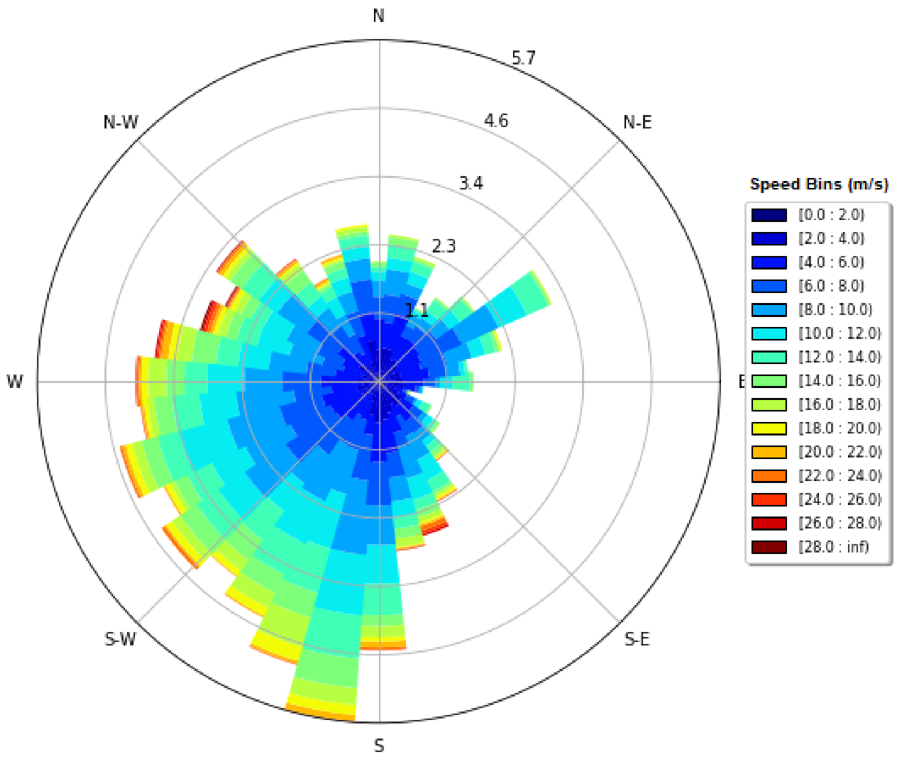

Figure 2: Wind rose diagram for the data in the file wind_data_2007.csv. Radial directions are partitioned into 36 equal sectors of 10° each and wind speed is binned into bins of 2 m/s (see legend on right).

- `WindSpeed(m/s)` - wind speed in meters per second
- `ThrustCoeffecient` - thrust coeffecient of the turbine
- `Power(MW)` - power generated by the turbine in megawatts (MW)

Thrust coefficient is a non-dimensional number used in wake effect modeling that represents force on the turbine blades due to the incoming wind. The power and thrust coefficient data gets eventually used during AEP calculations as 'look-up' table.

In Figure 3, we plot the power (in red) and thrust coefficient (in blue) versus the wind speed. In there, we show the *cut-in* wind speed, *cut-out* wind speed, *rated wind speed* and *rated power* of the turbine are labeled. We explain these terms below:

- Cut-in Wind Speed - Speed at which the blades start rotating and generating power.
- Cut-out Wind Speed - Speed at which the turbine shuts down to avoid exceeding load limits.

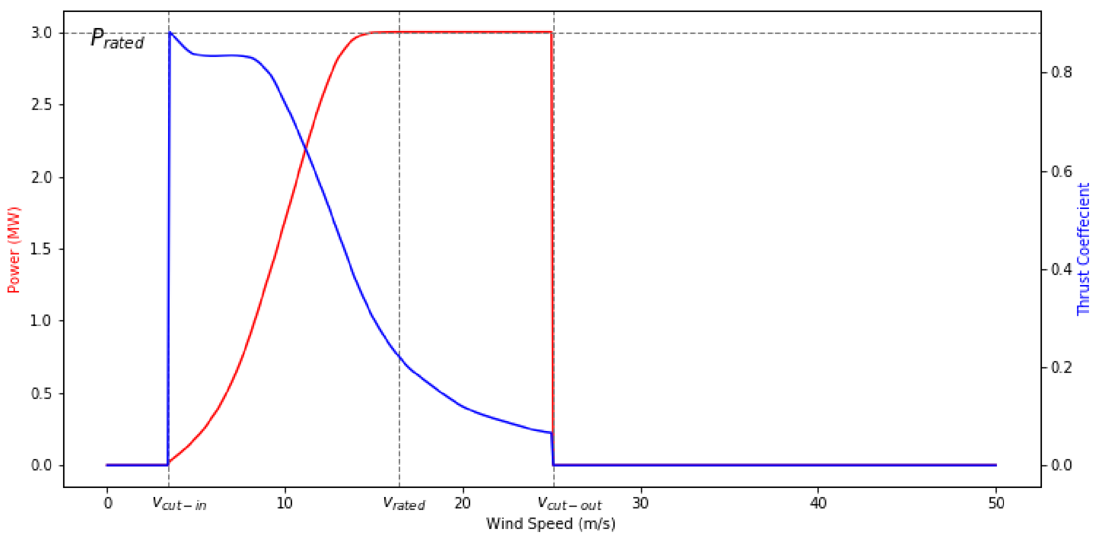

Figure 3: Power curve and thrust coefficient of the turbine.

- Rated Wind Speed - Speed at which the turbine is able to generate electricity at its maximum, or rated capacity.
- Rated Power (MW) - Maximum power that a turbine can generate in megawatts.

> *Take a note that the turbine specifications provided above and other eg. turbine rotor diameter, turbine height etc. are not provided in separate data files. We hard code these specifications in the wind farm evaluator codes.*

3.1.3 Test Turbine Locations

In the file `turbine_loc_test.csv`, we have provide a test data file containing the x and y coordinates of the location of 50 turbines. You can use this data to perform test runs of wind farm evaluator codes.

3.2 Wind Farm Evaluator

We provide to you two Python and one MATLAB file for calculating AEP of a wind turbine layout: `Farm_Evaluator.py`, `Farm_Evaluator_Vec.py` and `Farm_Evaluator.m`. Calculation wise, all three of these codes perform the same job i.e. they take as input the data files for turbine locations, turbine power curve and wind data, and finally return AEP of a wind turbine layout in Gigawatt hours (GWh) as the output. We have provided codes in two commonly used languages to assist different programming language choice.

`Farm_Evaluator.py` is written in a conventional manner and `Farm_Evaluator_Vec.py` is the *vectorized* version of the former, making it roughly 10 times faster. Due to its speed benefits, we advise participants to use `Farm_Evaluator_Vec.py` while building their optimizers. `Farm_Evaluator.py` can be used to better understand how the AEP calculations are happening. We provide in Section 5, the details of functioning of these codes, which involves the method to model inter-turbine interference - wake effect (Section 4).

The Python codes are developed in Python 3.7 release. Although much of the calculations inside makes use of pre-installed Python libraries: `Numpy` and `Pandas`, the user might need to manually install two other libraries that are used: `Tqdm` and `Shapely` to successfully run the codes.

☞ 3.2.1 A Note About Code Alteration

If a participant chooses to work in Python or MATLAB, then our advise is not to make changes inside the wind farm evaluator codes. Otherwise, this may end up resulting in different AEP output. As a prior safeguarding measure, we have explicitly mentioned in the files itself, which sections of the codes should not be modified.

There could however be certain cases in which the participants might desire to alter the code sections or develop a fresh piece of code for themselves, while making sure that the methodology of calculations remains unchanged. Some of these cases can be: *(i)* a possible performance (speed) bottleneck in the evaluator codes is located, *(ii)* the participant prefers a different programming language to work in, or *(iii)* the participant prefers to work in some algebraic modeling language like AIMMS, AMPL, GAMS, Pyomo, JuMP etc..

4 Wake Effect Modeling

In Section 1, we very briefly described wake effect in a wind turbine layout as the inter-turbine interference, due to which a turbine downstream of some upstream turbine may experience deficit in wind. This depends on: *(a)* whether the downstream turbine is inside the wake region of the upstream turbine, and *(b)* how far is the downstream turbine located from the upstream turbine.

PARK Model. There are several ways through which, it is possible to model the wake effect among wind turbines. We have used the classical Jensen model, also known as PARK model here [2]. The Jensen wake model is the most popular model in this area due to its high simplicity and practicality and is also the standard implementation in the wind resource assessment of many commercial softwares .

According to the PARK model, the equation for deficit due to a wake inducing upstream

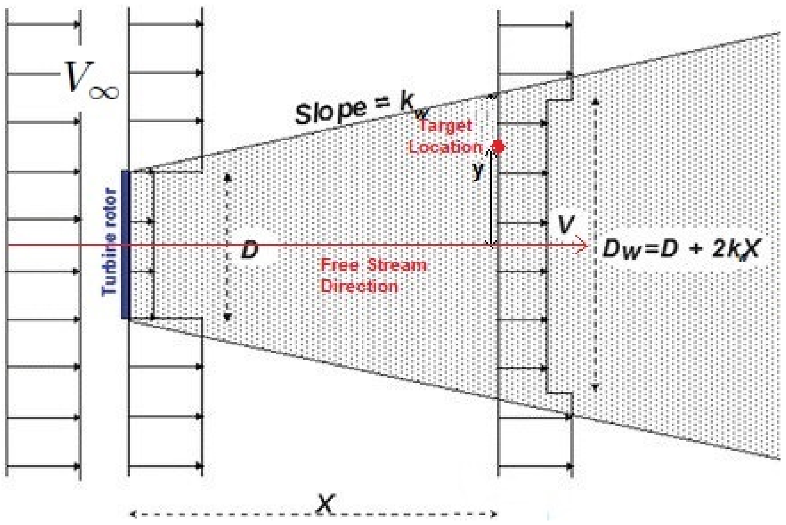

Figure 4: Wake expansion and deficit as modeled by the PARK model of N.O. Jensen. Image adopted from [3].

turbine is given by:

$$\frac{\Delta V}{V_\infty} = \begin{cases} (1-\sqrt{1-C_T})\left(\frac{D}{D+2k_w x}\right)^2 & \text{, if} \quad x > 0 \quad \text{and} \quad y \le (D+2k_w x)/2 \\ 0 & \text{, otherwise} \end{cases} \tag{1}$$

, where the description of the variables above in the equation is provided below.

ΔV	- Reduction in wind speed at a downstream distance x.
V_∞	- Unhindered free stream wind speed in m/s.
C_T	- Thrust coeffecient. Is estimated based on the value of V_∞
D	- Wake inducing turbine rotor diameter in meters.
k_w	- Wake decay constant. Typically has value 0.05 for offshore cases
x	- Distance of target location from wake generating turbine, along free stream
y	- Distance of target location from wake generating turbine, perpendicular to free stream

A diagrammatic representation of wake expansion and deficit as described by the PARK model is shown in Figure 4.

Note that we do not take partial wakes into consideration and ignore the wake effect if the turbine center of a target turbine is not falling inside the wake region of a wake generating turbine.

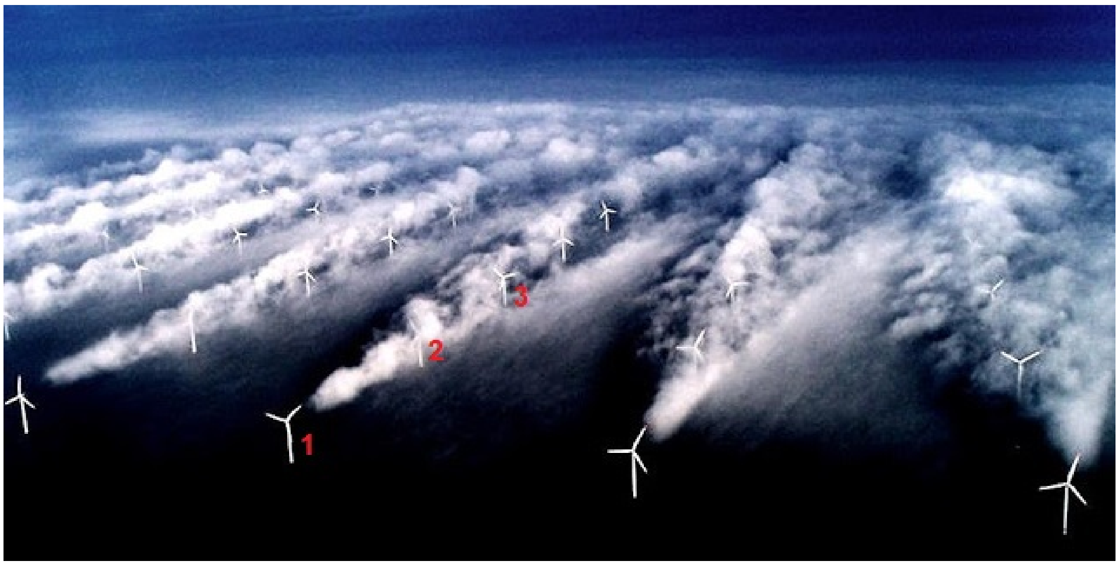

Figure 5: Turbines experiencing multiple wakes. As an example, turbine 3 is experiencing wake effects from both turbine 1 and 2. Image adopted from [4].

4.1 Wake Combination.

In a wind farm, it usually happens that a single turbine may be experiencing wake effects from multiple turbines (Figure 5). In such cases, we first independently calculate the wind speed deficit due to wake effect from each contributing upstream turbine on a target downstream turbine (using Eq. (1)) and then the total speed deficit suffered by this target turbine is calculated by using the square root of the sum of squares of the deficit from each upstream turbine:

$$\left(\frac{\Delta V}{V_\infty}\right)_{\text{total}} = \sqrt{\sum_{i=1}^{n} \left(\frac{\Delta V}{V_\infty}\right)_i^2} \tag{2}$$

, where n is total number of wake causing upstream turbines and $\left(\frac{\Delta V}{V_\infty}\right)_i$ is the individual deficit due to i^{th} turbine among n.

5 AEP Algorithm

We used the above described method of wake modeling in the provided wind farm evaluator codes for calculating the AEP of a turbine layout. In this section, we break down the algorithm inside these codes. We suggest to follow along the Python file `Farm_Evaluator.py`.

1. Read input files.
Implemented in the function - `getTurbLoc`, `loadPowerCurve`, `binWindResourceData`

- `turbine_loc_test.csv`. (x, y) locations of 50 turbines.
- `power_curve.csv`. Power and thrust coefficient data.
- `wind_data_<year>.csv`. Wind data.

2. Construct wind instances and calculate their probabilities.
Implemented in the function - `binWindResourceData`
We first need to 'discretize' the entire wind resource data into small *wind instances*. To do this, we bin wind direction and speed values into bins of sizes 10° and 2 m/s respectively and 'trap' number of data points inside these binned wind instances. This helps us to estimate the probability of occurrence of these wind instances, which we do by dividing the number of data points 'trapped' by the total number of data points. We denote the probability of occurrence of j^{th} wind instance by p_j We present below the discretized wind instances in a tabular format below with their corresponding direction and speed bins (denoted by s).

	0<=s<2	2<=s<4	....	....	26<=s<28	28<=s<30
drct = 360	..	..	..	..	..	..
drct = 10	..	..	..	..	..	..
drct = 20	..	..	..	..	..	..
....	..	..	..	..	..	..
drct = 340	..	..	..	..	..	..
drct = 350	..	..	..	..	..	..

We remind that that entries in `drct` column of the wind data provided are in multiples of 10 i.e. 360°, 10°, 20°. . . . 340°, 350°. 360° and 0° are one and the same thing. We have in total 540 wind instances.

$$\sum_{j=1}^{540} p_j = 1.0$$

> *Note that the direction and speed bin sizes chosen here are coarser than the desired resolution degree. We have chosen their sizes for computational reasons while not compromising heavily on wake effect modeling.*

3. Iterate over the wind instances.
Step 1. Rotate the frame of reference according to the wind flow direction. Implemented in the function - `rotatedFrame`
For the given wind flow direction, θ (in radians), convert the euclidean turbines (x, y) coordinates to *downwind-crosswind* coordinates, (x', y'). The shift is done so that the

wind flow direction aligns with the positive x-axis.

$$x' = x\cos\left(\theta - \frac{\pi}{2}\right) - y\sin\left(\theta - \frac{\pi}{2}\right)$$
$$y' = x\sin\left(\theta - \frac{\pi}{2}\right) + y\cos\left(\theta - \frac{\pi}{2}\right)$$

Step 2. Calculate effective wind speed at each turbine location using Jensen's PARK model. Implemented in the function - `jensenParkWake`
For the given free wind speed V_∞, now calculate the speed deficit experienced by a given turbine using Eq. (1) and (2) of the PARK model. The effective wind speed (V_{eff}) at each turbine location can then be computed as:

$$V_{eff} = V_\infty \left[1 - \left(\frac{\Delta V}{V_\infty}\right)_{\text{total}}\right]$$

Step 3. Estimate power production by the wind farm for this particular wind instance. Implemented in the function - `partAEP`
Use V_{eff} and power curve data from `power_curve.csv` to estimate the power produced by each turbine and obtain their sum. We denote the power produced by the wind farm for this particular j^{th} wind instance as P_j

Step 4. Repeat steps **Step 1.** to **Step 3.** for the next wind instance and record the respective P_j.

4. Calculate Wind Farm AEP. Implemented in the function - `totalAEP`
Multiply P_j with the corresponding frequency of occurrence of the wind instance (p_j). We provide the formula below for 540 wind instances. It is multiplied by a factor of 8760, which is the total number of hours in a year. A factor of 10^3 in the denominator converts the power to gigawatt-hours (GWh).

$$AEP = \frac{8760}{10^3}\left(\sum_{j=1}^{540} p_j P_j\right)$$

6 Solution Submission and Evaluation Criteria

Submission File Format. The final submitted solution should be a two-columned `.csv` file with *x,y* locations of 50 turbines.

> *In the solution file, keep the first row as column names x and y.*

Evaluation Criteria. We reserve annual wind data of:

- 3 years for public leader board, and
- 2 years for private leader board participant ranking.

Note that these annual wind datasets and the ones provided to you are all for different years, although they come from the same (anonymized) location.

> *On the leaderboards, the final score of a submission gets calculated as **mean AEP** score (in GWh), which is also our evaluation criteria for this Hackathon.*

7 References

[1] R. Barthelmie et al., “Modelling the impact of wakes on power output at Nysted and Horns Rev,” in *European Wind Energy Conference*, 2009.

[2] I. Katic, J. Højstrup and N. Jensen, “A simple model for cluster efficiency,” in *European Wind Energy Association Conference and Exhibition*, 1986.

[3] G. Giebel, C.B Hasager, “An Overview of Offshore Wind Farm Design,” Springer, Cham, 2016.

[4] P. Dvorak. ”Optimizing energy production: Addressing rotor wakes at wind farms”, Windpower Engineering and Development, 2016 [https://tinyurl.com/y68n8dbb]