# Shell.AI Hackathon


<u>Team: Penguins 408</u>

Omkar Nath

Uma T.V.

Mishma Mariyam Raju

B. Bharatwaajan

**Indian Institute of Technology Madras**

# Contents

- Introduction
- Inspiration for our Approach
- The Working of the Algorithm
- How did we get our best result
- Pros and Cons of our Approach

# Introduction

# Problem Statement

- Given a windfarm of an area of 4km x 4km, we need to optimize the arrangement of 50 turbines in it to maximize the power generation
- Turbines are all identical
- Certain constraints apply on proximity and perimeter

# Our Approach

- Initially we started looking at various standard approaches, and implementing them
- This gave us inspiration, which led to the development of our own unique algorithm, which we call “Incremental Improvement”

# Inspiration for the Approach

# Random Search

- Random Search is an algorithm which basically takes a turbine randomly, and moves it in a random direction. If there is an improvement, the new configuration is saved
- Feng, J., & Shen, W. Z. (2015). Solving the wind farm layout optimization problem using random search algorithm. *Renewable Energy*, *78*, 182-192.

# The Idea

- Our algorithm works on a very similar intuitive concept.
- The idea we had was to see if we could use this approach, but make it more formalized, and well-defined so as to give a proper framework to it

# The Working of the Algorithm

The Turbine Improvement Algorithm - Explained

# The Turbine Improvement Algorithm

- It takes a starting configuration.
- A specific distance is chosen.
- Then the code loops through all the 50 turbines.
- This further loops over a set of angles covering all 360 degrees.
- The chosen turbine is moved by that distance in that direction.
- The AEP is calculated. If there is an improvement, the new AEP is saved.

# The Turbine Improvement Algorithm

- In this manner, basically every turbine is moved by some distance in every direction.
- Furthermore, this can be repeated for various different distances.
- Every iteration, the configuration is guaranteed to be improved
- By choosing an appropriate set of angles and distances, and the sequence, surprisingly good results can be obtained.

# Varying Parameters - Distance

- Distance: After much analysis, a list of optimal distances was created. Running the code over these distances can yield good results.

*[700.0, 600.0, 500.0, 400.0, 300.0, 250.0, 200.0, 160.0, 135.0, 100.0, 88.0, 66.0, 48.0, 40.0, 34.0, 24.0, 16.0, 12.0, 9.0, 7.0, 5.0, 3.0, 2.0, 1.0]*

# Varying Parameters - Angle

- Angle: The interval of two angles also matters. The best was found to be about 10 degrees, as further smaller values was not required.

# How did we get our best result

# The initial configuration

- The initial configuration we had for our best result was just to pack all the points around the perimeter as densely as possible
- This gives the algorithm a lot of manoeuvrability, and hence let it choose the positions of the turbines

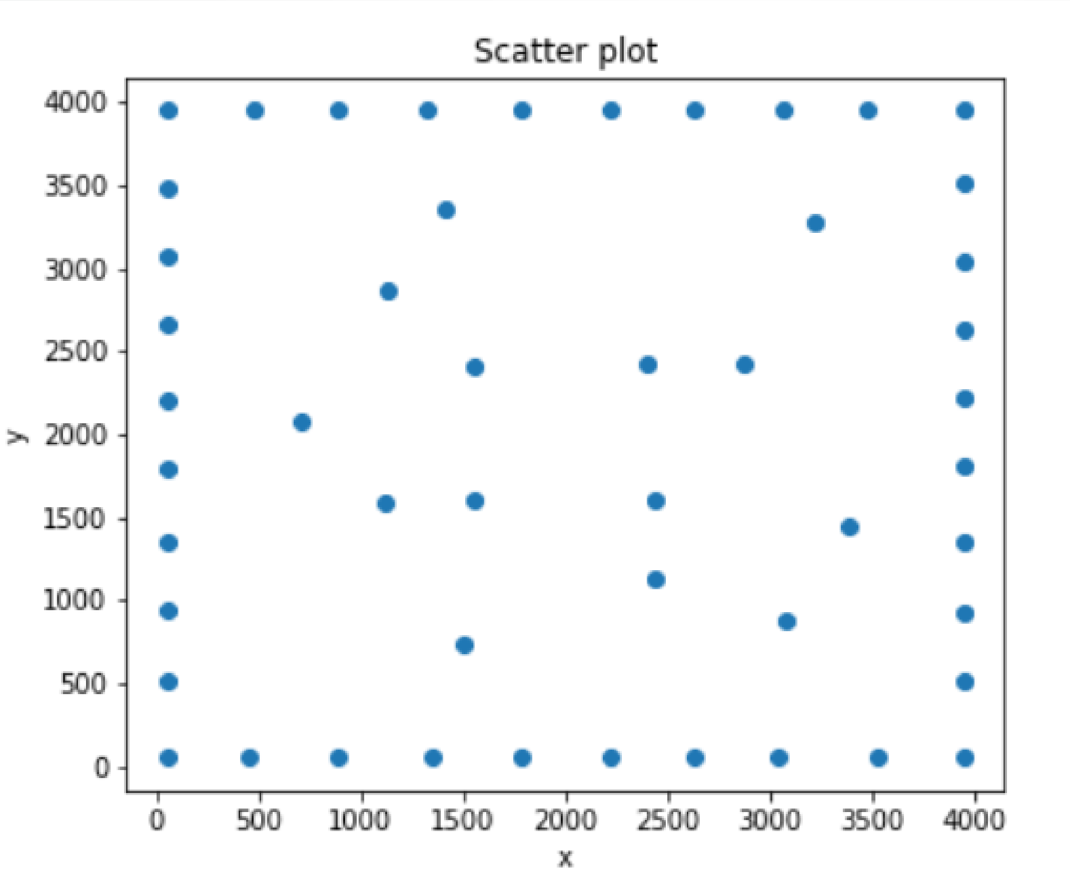
# Running the algorithm

- To keep improving results, the code was run multiple times over more finer distances and angles, to get better results. Some of these runs even took hours, to say improve by the last point.
- Note that the values given for the distances thetas are already very good. Using them, one can easily make any random configuration give a really good AEP in a matter of mere minutes. Even seconds.

# Running the algorithm

- Running the code over small distances and smaller thetas will give slightly better results.
- But the time complexity is too high for the almost negligible gains. For all intents and purposes, the simplified sets given in the code are sufficient

# Visualizing our Best Result



# Pros and Cons of Our Approach

# Advantages of our Code

- It is simple, and relatively easy to understand. Also makes intuitive sense.
- It works for any type of configuration, being able to drastically improve the AEP showing its robustness
- It is very time efficient, being very computationally light.
- It gives results, giving satisfactory AEPs on every run.
- The key to it is simply in the choosing of the various parameters of distances.
- Can easily be adopted for other scenarios of wind farm optimization.

# Areas of Improvement

- As only one turbine is moved at a time, it cannot handle situations where multiple turbines being moved as a group may lead to an improvement. This is hardly noticeable though, as the minor movements of the turbines can eventually lead to the same results.
- May not necessarily obtain the best configuration, though it will give you a really good configuration really fast.

Thank You