



Práctica Pre-Profesional
513498-1

Informe de resultados

Optimización del posicionamiento de
aerogeneradores para una granja eólica
offshore.

Tamara Paz García Latorre
Matrícula: 2020427330
Tutora de práctica: Natalia Aziares Aguayo

Lunes 5 de Febrero, 2024.

Índice

Resumen	2
Introducción	3
Marco teórico	4
Librerías Revisadas	4
Meteodyn WFO: The wind farm optimization software	4
SOWFA: Simulator for Wind Farm Applications	5
PyWake	6
TOPFARM	9
Otras librerías	11
Objetivos	12
Objetivo general	12
Objetivos específicos	12
Metodología	13
Selección de una librería	13
Diagrama de trabajo	13
Parámetros y Simulación	14
Ubicación	14
Parámetros	15
Aplicando la optimización	19
Resultados	20
CASO 1: Totalidad de datos	20
CASO 2: 40 % mejor de los datos	22
Discusión	24
Conclusión	25
Referencias	26

Resumen

Este informe recopila la información adquirida y las actividades realizadas durante mi práctica pre-profesional en la carrera de Geofísica de la Universidad de Concepción en el Centro de Investigación MERIC. Esta experiencia estuvo enfocada en recabar información para seleccionar una biblioteca del lenguaje de programación Python, o similar, con el objetivo de aplicar una optimización a la ubicación de aerogeneradores en una granja eólica off-shore de un caso de estudio en la región del Biobío, Chile, con la mira en adquirir experiencia e información para su aplicación en futuros proyectos.

La biblioteca utilizada fue PyWake, la cual se centra en la interacción del efecto estela con los distintos aerogeneradores calculando la producción de energía anual(AEP) de la granja, junto con la librería TOPFARM, que se especializa específicamente en la optimización de variables tal como las ubicaciones de los aerogeneradores para encontrar aquellas donde el AEP sea más alto. El resultado tras aplicar estas técnicas consiste en una grilla con las ubicaciones optimizadas de los aerogeneradores donde se logra un mayor AEP, de acuerdo a las condiciones de viento y diversos parámetros utilizados.

Introducción

Dentro del marco del cambio climático y la necesidad de transformar la matriz energética de los países hacia fuentes renovables la energía eólica ha demostrado tener un gran potencial y crecimiento en la última década, y es de esta misma que surge una nueva técnica que es la implementación de la energía eólica offshore, o energía eólica en alta mar, la cual se presenta como una innovadora solución para la generación sostenible de energía. En lugar de ubicar los aerogeneradores en tierra firme, la energía eólica offshore implica instalar estas estructuras en el mar, aprovechando los vientos constantes, fuertes y menos turbulentos (1) que se encuentran en aguas profundas.

Chile, con su extensa línea costera y condiciones climáticas propicias, se posiciona como un terreno con gran potencial para el desarrollo de la energía eólica offshore (2). La abundancia de vientos en las costas chilenas presenta un potencial significativo para la generación de energía, convirtiendo al país en un candidato idóneo para aprovechar esta tecnología emergente. Además, la instalación de parques eólicos en alta mar puede superar las limitaciones de espacio y logística asociadas con los proyectos en tierra firme, permitiendo el despliegue de aerogeneradores de mayor tamaño y capacidad.

Las ventajas de la energía eólica offshore son diversas, desde una mayor eficiencia derivada de vientos más consistentes hasta la reducción de impactos visuales y sonoros en comparación con proyectos en tierra. Sin embargo, la eficiencia de estas instalaciones depende en gran medida de la distribución de los aerogeneradores y es aquí donde la optimización se vuelve crucial para maximizar la producción de energía minimizando efectos adversos, tal como la interferencia entre turbinas que generan el fenómeno conocido como estela (3). La correcta disposición de los aerogeneradores se traduce en una mayor eficiencia operativa y, en última instancia, en un rendimiento más rentable y sostenible de la granja eólica offshore. A lo largo de este informe, se explora detalladamente el proceso de optimización del posicionamiento de aerogeneradores de un caso de estudio en Chile mediante el lenguaje de programación Python utilizando las librerías Pywake para la simulación de la granja eólica y TOPFARM para la optimización.

Marco Teórico

Librerías Revisadas

Meteodyn WFO: The wind farm optimization software

Meteodyn WFO (4) es un software de optimización de parques eólicos On-shore y Off-shore que selecciona los tipos de aerogeneradores y optimiza su disposición en función de varios parámetros personalizables. La optimización es posible basándose en un criterio de producción (producción, factor de capacidad u hora a plena carga) o en un criterio financiero (coste nivelado de la energía (LCOE) o coste por kWh). El criterio “Fitness” tiene en cuenta tanto la producción como el LCOE y el coste de la inversión.



Este software puede utilizar un mapa de recursos eólicos del sitio proporcionado por el usuario o una simulación de parque eólico realizada en el software Meteodyn WT. En caso de limitaciones del terreno, es posible crear zonas de exclusión que impidan la colocación de aerogeneradores en determinadas zonas. El software calcula rápidamente el diseño más adecuado según los ajustes definidos y la función objetivo seleccionada por el usuario. A continuación se describen los módulos que lo componen:

1. **Módulo de optimización de parques eólicos:** Mediante esta herramienta es posible agregar zonas de exclusión, crear diferentes grupos de aerogeneradores conteniendo cada uno de ellos un número mínimo y/o máximo de aerogeneradores, se pueden definir los tipos de aerogeneradores, sus coordenadas y su altura.

Dentro de este módulo se encuentran diversas características, tales como:

- Optimización del diseño del parque eólico: La disposición de los aerogeneradores puede optimizarse definiendo un número mínimo o máximo de estos, la producción o el factor de capacidad.
- Restricciones: El usuario puede agregar restricciones mínimas y máximas para las siguientes variables; ángulo de entrada, intensidad de turbulencia, velocidad extrema del viento, cizalladura, densidad del aire, velocidad del viento, densidad de potencia y altura.
- Optimización de la producción: Se optimiza la producción según tres criterios; Producción de energía, factor de capacidad y horas de plena carga

2. **Módulo de tipo de aerogenerador:** El usuario puede modificar los tipos de aerogeneradores disponibles para lograr la optimización del parque eólico y definir sus características técnicas como potencia nominal, diámetro del rotor, número de palas, control de potencia, velocidad de rotación mínima y máxima, entre otros.

Las fortalezas de MeteodynWFO se encuentran en su gran variedad de parámetros siendo estos personalizables, como las **tasas de eficiencia de producción de energía** según la disponibilidad de la turbina, la contaminación de las palas, errores de cálculo, influencia del clima, consumo de electricidad y pérdida de cables, fluctuación de la frecuencia de la red y apagones eléctricos, curva de potencia y efecto de estela. **Parámetros de costos de energía** como préstamo, valor residual, producción total de energía de por vida (TLE), operación y mantenimiento (OM), etc.

SOWFA: Simulator for Wind Farm Applications

SOWFA es un conjunto de solucionadores de dinámica de fluidos computacional (CFD), condiciones de contorno y modelos de turbina. Se basa en la caja de herramientas OpenFOAM CFD e incluye una versión del modelo de turbina acoplada con FAST. Esta herramienta permite a los usuarios investigar el rendimiento y la carga de aerogeneradores y plantas eólicas en toda la gama de condiciones atmosféricas y en el terreno (5).

Dentro de las características del paquete se encuentran:

- Solucionador de capa límite atmosférica
- Solucionador de planta eólica
- Clase de modelo de turbina de línea de actuador (estándar y acoplado con FAST)
- Clases de condiciones de límite atmosféricas/eólicas específicas de planta
- Utilidades para la inicialización del campo de flujo
- Utilidad para convertir datos de límites de muestreo previo en datos de flujo de entrada para el solucionador de planta eólica

SOWFA un paquete de software de código abierto acoplado y que permite a los usuarios investigar los efectos de los patrones meteorológicos, la turbulencia y el terreno complejo en el rendimiento de las turbinas y plantas eólicas. Simula la dinámica de fluidos a escalas que van desde la meteorología regional y el impacto de estas dinámicas en la respuesta estructural y del sistema de la turbina (6).

Trabajar directamente con el código fuente de SOWFA y OpenFOAM, requiere conocimiento de programación en C++. Pues ambas herramientas están escritas en C++ y utilizan este lenguaje de programación como base.

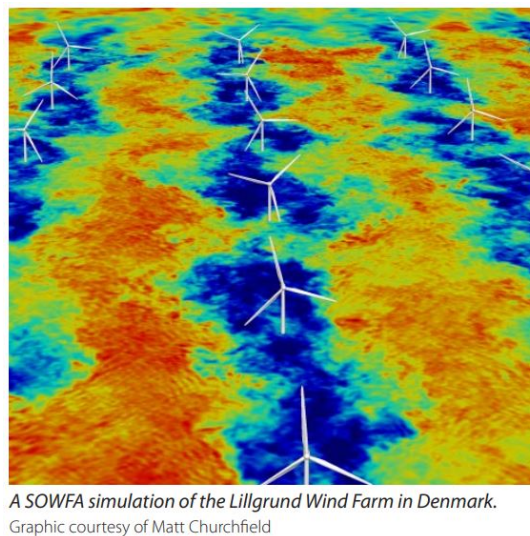


Figura 1: Simulator fOr Wind Farm Applications helps optimize layouts and controls of wind plant arrays
Fuente:(6).

PyWake

PyWake es una librería o herramienta de simulación de parques eólicos de código abierto y basada en Python desarrollada en DTU. Su objetivo principal es calcular la interacción de la estela en una granja eólica de forma computacionalmente económica. Entre sus características destaca la capacidad de calcular campos de flujo, producción de energía de turbinas de forma individual, producción anual de energía (AEP) de una granja eólica (7).



PyWake proporciona una interfaz unificada que permite la integración de modelos de parques eólicos con diferentes niveles de fidelidad, como modelos de ingeniería y CFD-RANS. Gracias a su intensa vectorización y el uso de bibliotecas numéricas, PyWake destaca por su rapidez y capacidad para manejar múltiples variables simultáneamente. La herramienta ofrece flexibilidad al permitir a los usuarios combinar diversos bloques de modelado AEP, adaptándose así a las particularidades de problemas del mundo real con mayor precisión. También es capaz de simular **granjas eólicas offshore** donde puede incluirse la distancia desde la costa y la profundidad del agua.

Por otro lado, PyWake se complementa de manera eficiente con la herramienta de optimización de parques eólicos TOPFARM. Esta integración facilita la **optimización de la distribución de turbinas en un parque eólico con el objetivo de maximizar la producción de energía**.

Toda la información expuesta proviene de su sitio web oficial donde se entregan más detalles en <https://www.pythonpool.com/pywake-library/>.

Entonces, PyWake puede ayudar a los usuarios a:

- Calcular la producción de energía de turbinas individuales y la producción anual de energía (AEP) del parque eólico.
- Calcular campos de flujo.
- Implementar optimizaciones junto con la librería TOPFARM.

1. Datos requeridos y resultados

Los inputs necesarios para correr una simulación en PyWake varían según el problema a resolver, según presentado en este informe son:

- a) La ubicación y distribución inicial de los aerogeneradores del parque eólico.
- b) Las curvas de potencia y CT de las turbinas eólicas, altura del buje y diámetro del rotor.
- c) Información sobre las condiciones de viento de la zona (datos históricos de velocidad y dirección).

Por otro lado, los principales outputs de PyWake son:

- a) La producción de energía de turbinas individuales.
- b) La producción anual de energía (AEP) de un parque eólico.
- c) Los campos de flujo en el parque eólico.

Los resultados salen en formato de datos de Python, que se pueden utilizar para análisis o visualización. Si se aplica una optimización con TOPFARM existirán otros outputs según el caso, por ejemplo, coordenadas optimizadas, alturas de buje, tipos de turbina, entre otros.

2. Arquitectura para la simulación:

El diagrama de trabajo y estructura de PyWake se muestra en la figura (2), el objeto principal en la arquitectura de PyWake es **WindFarmModel**, que se inicializa con un objeto **Site** y **WindTurbines**. Devuelve un objeto “SimulationResult” que contiene la velocidad efectiva del viento calculada, la producción de energía y el coeficiente de empuje de las turbinas individuales. Además, se basa en métodos para calcular el **AEP** y generar **mapas de flujo** de parques eólicos completos.

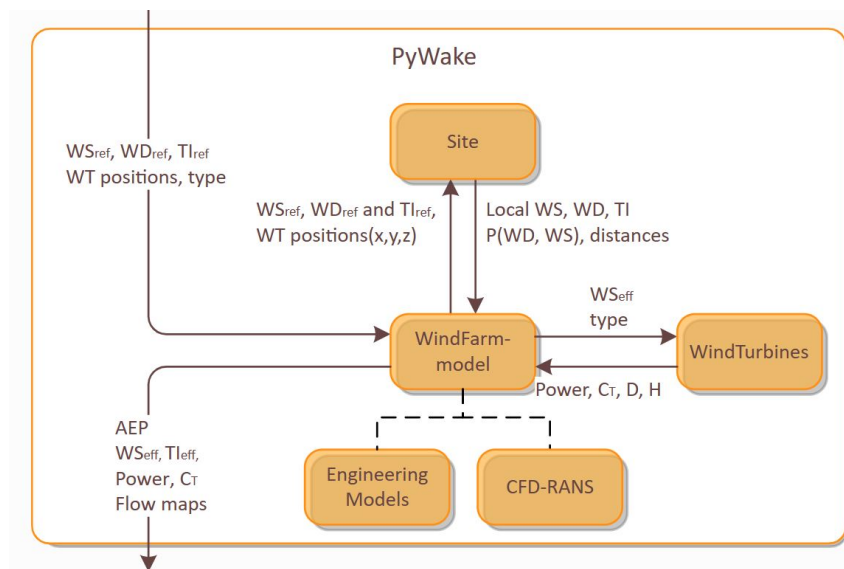


Figura 2: Diagrama de flujo de trabajo de PyWake (7).

3. Objetos principales:

- **Site**: Para determinadas posiciones de los aerogeneradores (x, y), velocidad del viento de referencia (WS_{ref}) y dirección del viento (WD_{ref}), el objeto Site proporciona las condiciones locales del viento en términos de velocidad del viento (WS), dirección del viento (WD), intensidad de la turbulencia (TI) y probabilidad de cada combinación de dirección y velocidad del viento. Además, el objeto Site se encarga de calcular la distancia a favor del viento, a contraviento y vertical entre aerogeneradores
- **WindTurbines**: En este objeto se encuentra la información del tipo de aerogenerador a utilizar, existen algunos predefinidos y también se pueden configurar según los requerimientos. Para un tipo de aerogenerador y una velocidad efectiva del viento (W_{seff}) determinados, el objeto WindTurbines proporciona la curva de potencia, la curva del coeficiente de empuje (CT), así como la altura del buje (h) y el diámetro (D) del aerogenerador al modelo.
- **WindFarmModel**: El modelo de parque eólico puede ser predefinido con opciones que abarcan todos los modelos de ingeniería disponibles que se han implementado a partir de la literatura. Cada uno de los modelos viene con configuraciones predeterminadas en términos de modelos de déficit de estela, modelos de superposición, modelos de turbulencia, etc. Pero también pueden personalizarse en cuanto a la elección de los modelos que utiliza. En este objeto se reúnen todos los demás para generar la simulación y cálculos.

Mientras que siempre es necesario seleccionar un modelo de déficit de estela todos los demás componentes o módulos son opcionales.

4. Opciones predefinidas

Los objetos para iniciar `WindFarmModel` tiene opciones predefinidas que se muestran en los siguientes puntos:

- a) **Site:** Los sitios predefinidos se pueden revisar en la página oficial y por otro lado se pueden visualizar gráficos de sus propiedades como la rosa de los vientos y la función de densidad de probabilidad de estos para tener mayores detalles.
 - **IEA37Site:** Utiliza la función `UniformSite` (Viento uniforme), tiene 16 turbinas eólicas en un sitio uniforme con una velocidad del viento fija de 9,8 m/s y, por lo tanto, las variables de datos solo dependen de la dirección del viento.
 - **Hornsrev1:** Utiliza la función `UniformWeibullSite` (velocidad del viento uniforme dependiente del sector y distribuida por Weibull.), tiene 80 aerogeneradores en un emplazamiento uniforme, por lo que las variables de datos dependen de la dirección y la velocidad del viento.
 - **ParqueFicticioSite:** Utiliza la función `WaspGridSite` (flujo de entrada no uniforme cuadrulado basado en archivos *.grd exportados de WAsP). Tiene 8 turbinas en un terreno complejo y, por tanto, las variables de datos dependen de la dirección y velocidad del viento y de la posición.

Existen otras funciones a demás de las expuestas con las que el usuario puede definir un Site personalizado.

- b) **WindTurbine:** PyWake contiene algunas turbinas predefinidas, por ejemplo, la V80 y la DTU10MW. También se pueden importar archivos de aerogeneradores desde archivos WAsP (.wtg). Las propiedades de estas pueden visualizarse en gráficos como la curvas de potencia y CT.

Por otra parte, el usuario puede definir su turbina mediante las siguientes funciones:

- **GenericWindTurbine:** Esta clase define un aerogenerador cuya potencia se calcula mediante un modelo analítico basado en el diámetro y la potencia nominal. Ofrece numerosas entradas opcionales, como altura del buje, diámetro del rotor, potencia y turbulencia, que por defecto son valores empíricos.
 - **WindTurbine:** En esta clase, es necesario proporcionar el coeficiente de empuje (CT), que describe la fuerza del viento sobre la turbina, así como la curva de potencia que representa la capacidad de generación eléctrica de la turbina en función del recurso eólico disponible (3). Además, se deben especificar la altura del buje y el diámetro del aerogenerador.
- c) **Engineering Wind Farm Models:** Los modelos de parques eólicos se derivan de dos clases base que deben tener especificada al momento de crear el objeto `WindFarmModel`. Estos dos definen el procedimiento que determina cómo se propagan los déficits de estela y bloqueo en el parque eólico:
 - **PropagateDownwind:** Itera sobre todas las turbinas en orden descendente, calculando la velocidad efectiva del viento y el déficit causado por cada turbina aguas abajo. Rápido, pero ignora los efectos de bloqueo aguas arriba.
 - **All2AllIterative:** Más lento pero capaz de manejar efectos de bloqueo. Itera hasta que la velocidad efectiva del viento converge, calculando el déficit causado por cada turbina en todas las turbinas. Permite un modelado más preciso con efectos de bloqueo.
 - d) **Wake deficit models:** Calcula el déficit de estela de una turbina a las turbinas de viento aguas abajo en el parque eólico. Modelos como `NOJDeficit`, `TurboNOJDeficit` y `FugaDeficit` pueden implementarse fácilmente, cada uno tienen sus características. Además, estos modelos se pueden combinar fácilmente con otros para describir mejor la estela detrás de la turbina.

TOPFARM

PyWake se integra con la herramienta de optimización de parques eólicos TOPFARM (8) un paquete Python de código abierto desarrollado por DTU Wind Energy que sirve como optimizador de parques eólicos tanto terrestres como marinos. Los cálculos de las interacciones de los parques eólicos se realizan a través de PyWake, que se encarga de calcular las pérdidas de estela y la producción de energía tanto de turbinas individuales como de parques eólicos completos con el uso de modelos de estela de ingeniería.

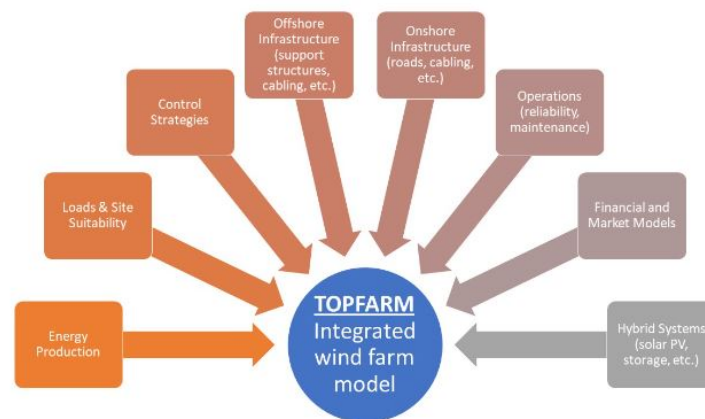


Figura 3: Diagrama de TOPFARM, Riva et al. ('Welcome to TOPFARM') (8).

En TOPFARM, la función objetivo se evalúa mediante el componente del modelo de costes, y puede representarse mediante objetivos de producción de energía o financieros. Actualmente este paquete de optimización puede proporcionar soluciones en:

- Optimización del diseño de parques eólicos para distintos tipos de turbinas y alturas de buje.
- Optimización del control activo (wake steering).
- Optimización de la disposición o control de la estela con restricciones de carga.
- Optimización con batimetría.
- Optimización de la disposición basada en LCoE.

Su arquitectura se divide en componentes, que interactúan entre sí para crear el objeto `TopFarmProblem` y posteriormente trabajan para encontrar la mejor solución posible a un problema de optimización.

Para mayor detalle e información visitar el sitio oficial de TOPFARM: <https://topfarm.pages.windenergy.dtu.dk/TopFarm2/index.html>.

Estructura de TOPFARM

El objeto principal de TOPFARM es el objeto `TopFarmProblem` tal como se observa en el diagrama de flujo (4) con el cual se realiza la optimización de ciertas variables utilizando a la vez parámetros definidos en:

- a) **Desing variables:** Variables que serán modificadas durante la optimización y que influyen en la función objetivo. Pueden utilizarse las posiciones de las turbinas, tipos de turbina, la altura del buje y los ángulos de guiñada.
- b) **Cost Component:** Es el objeto que calcula la función objetivo de la optimización (por ejemplo, AEP).
- c) **Drivers:** (opcional) Es el controlador de la optimización que se utiliza para realizar la optimización numérica. Hay controladores sencillos listos para usar y pueden realizar optimizaciones basadas o no en el gradiente. Sin embargo, también se pueden acoplar controladores de optimización personalizados a un flujo de trabajo. El controlador por defecto es el "EasyScipyOptimize", que utiliza por defecto el algoritmo "SLSQP" basado en gradiente de scipy.

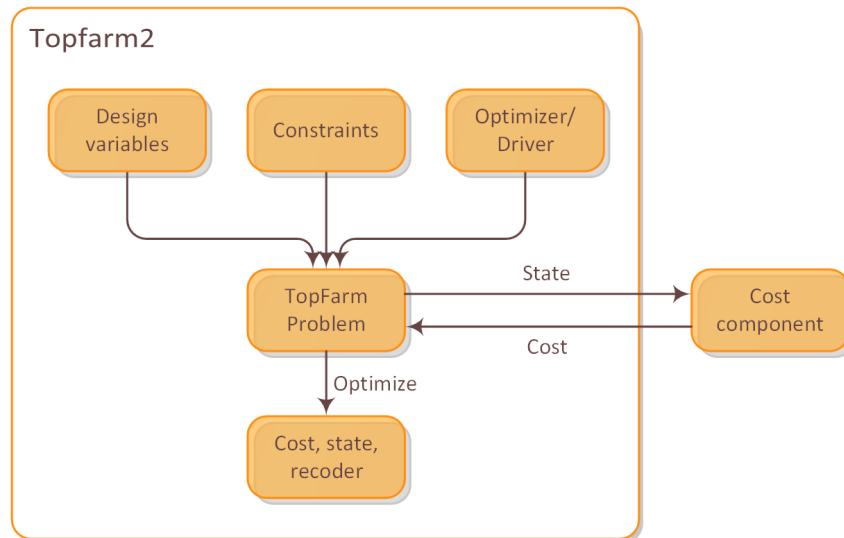


Figura 4: Diagrama de flujo de trabajo de TOPFARM (8).

- d) **Constraints:** (opcional) Son restricciones que se establecen sobre las variables de diseño. Las más comunes son las restricciones de espaciado entre turbinas o los límites del parque eólico y se pueden especificar condiciones adicionales.
- e) **Plotting component:** (opcional) La clase TopFarmProblem permite al usuario visualizar la optimización con un componente de trazado. Si se activa, muestra el estado del parque eólico a medida que se optimiza, incluido el estado actual de la función objetivo en cada iteración.

Definiendo estos parámetros y aplicada la optimización se entregan tres elementos como output:

- **cost:** Corresponde al resultado final de la optimización.
- **state:** Proporciona los valores finales de las variables de diseño utilizadas en la optimización.
- **recorder:** Presenta un registro de los conjuntos de variables de diseño que se probaron durante la optimización.

En el sitio oficial se encuentran diversos ejemplos de diferente complejidad y fragmentos de código para guiarse.

Otras librerías de interés

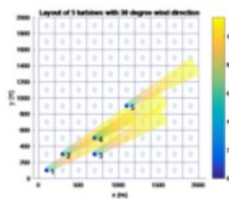
En los siguientes puntos se entregan enlaces de otros softwares o librerías encontradas en las cuáles no se profundizó debido a que son de pago o utilizan otro lenguaje de programación.

- a) **WFLO-code**: Por Nantiwat Pholdee. Corresponde a una serie de funciones desarrolladas en MATLAB que se encuentran en el repositorio <https://github.com/nantiwat/WFLO-code>.

El código fue desarrollado para la optimización del diseño de parques eólicos, utilizando meta-heurísticas detalladas por Kunakote, Sabangban, Kumar, Tejani, Panagant, Pholdee, Bureerat y Yildiz (2021) (9). Las estrategias metaheurísticas se describen en su artículo “Comparative Performance of Twelve Metaheuristics for Wind Farm Layout Optimisation”, publicado en Archives of Computational Methods in Engineering, páginas 1-14.

También está registrado en la página oficial de MATLAB:

<https://la.mathworks.com/matlabcentral/fileexchange/93315-wflo-code>.



WFLO-code

Versión 1.0.0 (202 KB) por Nantiwat Pholdee

wind farm layout optimisation based on metaheuristics

<https://github.com/nantiwat/WFLO-code>

Figura 5: Repositorio de WFLO-code (9).

- b) **WindfarmGA**: Paquete alojado en CRAN que corresponde a un algoritmo genético diseñado para optimizar parques eólicos de cualquier forma. Requiere un número predefinido de turbinas, un radio uniforme del rotor y una velocidad media del viento por dirección del viento.

Puede incluir un modelo de efectos del terreno, que descarga automáticamente un modelo de elevación “SRTM” y un raster “Corine Land Cover”. El modelo de elevación se utiliza para encontrar montañas y valles y ajustar en consecuencia las velocidades del viento mediante “multiplicadores del viento” y para determinar las densidades del aire a la altura de los rotores. El raster de la cubierta terrestre con un valor adicional de rugosidad de la elevación se utiliza para re-evaluar la rugosidad de la superficie y determinar individualmente la constante de decaimiento de la estela para cada turbina.

Los códigos se basan en el lenguaje de programación R, para mayor información se puede visitar:

<https://ysosirius.github.io/windfarmGA/>

<https://github.com/YsoSirius/windfarmGA>.



Figura 6: windfarmGA, Genetic algorithm to optimize the layout of windfarms (10).

Objetivos

Objetivo general

Revisión de librerías y su aplicación para la optimización en la distribución de aerogeneradores de una Granja Eólica Offshore en un caso de estudio en la región del Biobío, Chile.

Objetivos específicos

- a)* Familiarizarse con el funcionamiento y bases de la energía eólica offshore.
- b)* Comprender el funcionamiento y características de las librerías revisadas seleccionando la que mejor se adapte los requerimientos del problema.
- c)* Generar una primera simulación de la granja eólica con una distribución aleatoria de los aerogeneradores.
- d)* Aplicar la optimización a la ubicación de los aerogeneradores estableciendo los parámetros correspondientes para obtener una grilla con las coordenadas optimizadas.
- e)* Comparar la producción de energía anual previa y posterior a la optimización.

Metodología

Selección de una librería

Según lo recopilado en la revisión bibliográfica de las librerías que se encuentran en la sección de “Marco teórico” se elige la librería **PyWake** para realizar la simulación, dentro de la cual se incorpora información técnica de las turbinas y de las condiciones de viento de la zona de estudio para posteriormente utilizar la librería **TOPFARM** donde se realiza la optimización de las ubicaciones de los aerogeneradores. Esta elección se basa principalmente en que ambas librerías son de código abierto y fácil acceso, contando con información y ejemplos en sus respectivos sitios web. Al revisar la información necesaria para su uso se determinó que se cuenta con los datos inputs y los outputs son los esperados.

Diagrama de trabajo

El siguiente diagrama de flujo sintetiza la metodología seguida para llevar a cabo el objetivo. Aclarar además que de estos pasos el procedimiento simple y general para lograr una familiarización con las librerías consistió en correr ejemplos de los sitios oficiales e ir realizando modificaciones hasta lograr crear los diferentes objetos necesarios para la simulación. Este trabajo paulatino permitió incorporar parámetros más personalizados. Los códigos y datos utilizados se pueden encontrar en Github (11).

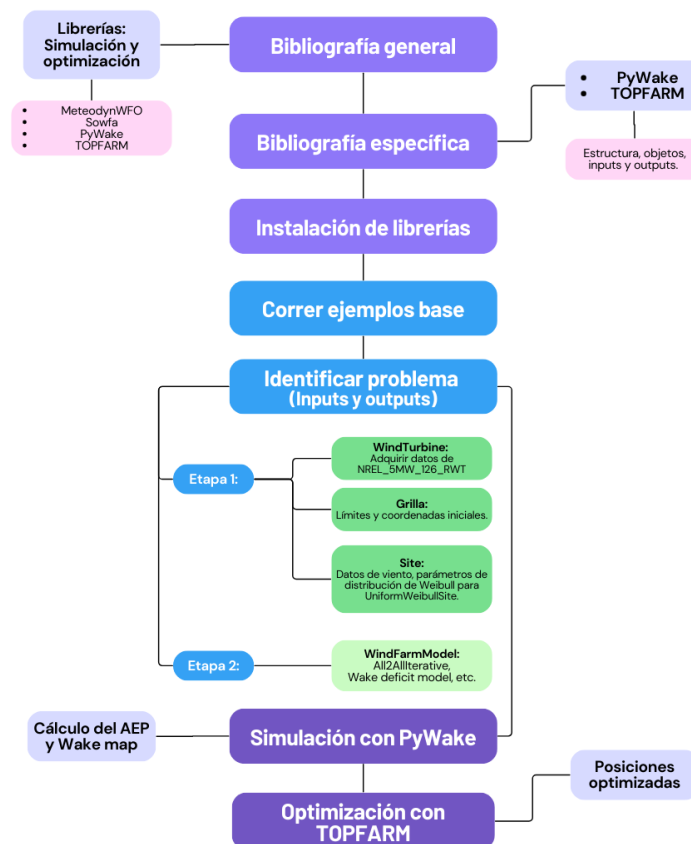


Figura 7: Diagrama de trabajo. Fuente: Elaboración propia.

Parámetros y Simulación

Ubicación:

La zona de estudio se encuentra en la costa de Chile, cerca de la ciudad de Concepción frente a la Península de Tumbes en la región del Biobío (ver figura 8). Las profundidades del fondo oceánico del área de emplazamiento se encuentran en un rango de [138-200] metros de profundidad y la velocidad promedio en el punto más ventoso del área tiene un valor de 9.6 [m/s], permitiendo una potencia media respectiva de 975 [W/m²] a una altura de 100 [m] (12).

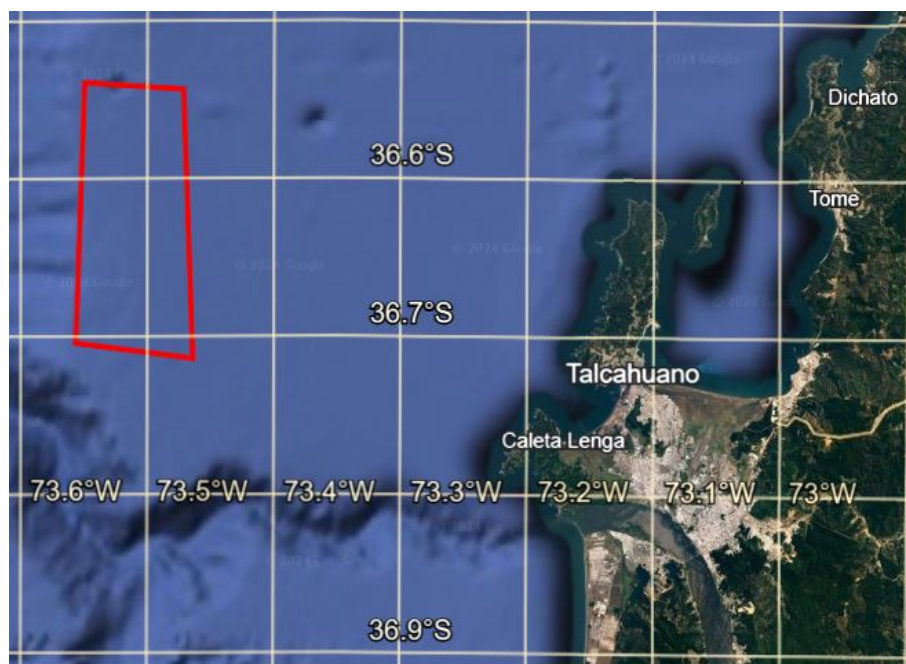


Figura 8: Posible ubicación del área de emplazamiento de la Granja Eólica Off-Shore. Área dentro de los límites del polígono rojo. Fuente: Captura de Google Earth:(13).

- a) **Grilla:** Los límites de la zona fueron entregados en un archivo .kmz de cual se extrajeron los vértices de un polígono que demarcaba el área de emplazamiento en coordenadas con formato decimales. Durante la ejecución de los ejemplos en PyWake los límites y ubicaciones se encontraban en coordenadas con formato UTM por lo que se realiza una conversión de los puntos dados utilizando la herramienta SIRGAS Chile: Convertidor de Coordenadas (14) para agilizar el proceso. Las coordenadas UTM pertenecen a la zona 18H con unidad en metros.

Los puntos originales y convertidos se entregan en la siguiente tabla y pueden observarse en la figura (9):

Punto	Coordenada	Decimal	Coordenada	UTM	Zona
A	Latitud:	-36.540058	Este x:	629805	18H
	Longitud:	-73.549853	Norte y:	5955176	
B	Latitud:	-36.544191	Este x:	636818	18H
	Longitud:	-73.471424	Norte y:	5954603	
C	Latitud:	-36.714829	Este x:	637218	18H
	Longitud:	-73.463581	Norte y:	5935661	
D	Latitud:	-36.705398	Este x:	629003	18H
	Longitud:	-73.555744	Norte y:	5936836	

Cuadro 1: Coordenadas originales en formato Decimal del área de emplazamiento y coordenadas transformadas a formato UTM.

- b) **Posiciones iniciales:** Se establecen 18 Aerogeneradores y sus posición iniciales son aleatorias al igual que la separación entre ellas. Notar que algunas se encuentran fuera de los límites del polígono (observar figura 9).

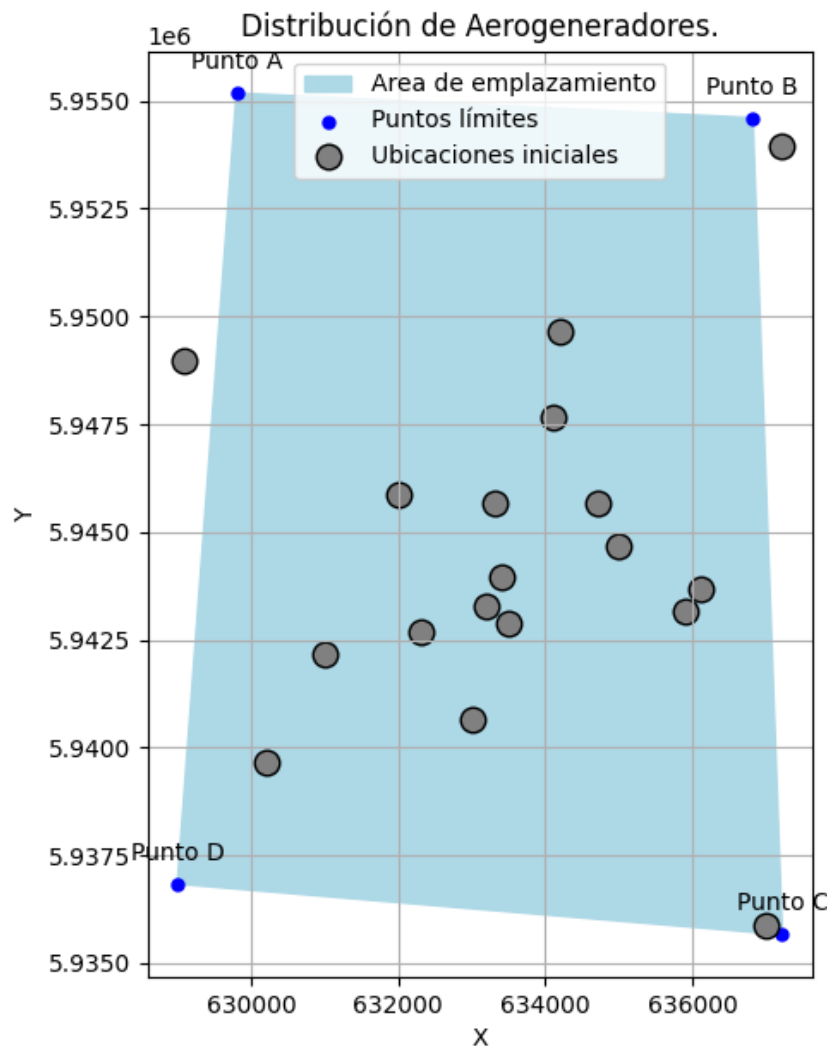


Figura 9: Disposición inicial de los aerogeneradores (círculos grises) en el área de emplazamiento (polígono celeste). Los puntos límites corresponden en color azul. Fuente: Elaboración propia mediante Python.

Parámetros

- a) **Tipo de turbina:** En este caso el tipo de turbina se define dentro del objeto [WindTurbine](#). Para ello se debe contar con información de la altura del buje, el diametro del rotor, curva de potencia y coeficiente de empuje del tipo de aerogenerador. Los dos últimos deben ser vectores con sus respectivas velocidades de viento.

Esta información procede de una turbina de referencia llamada **NREL_5MW_126_RWT** cuyos parámetros fueron obtenidos de la página [NREL Turbine Archive](#) (15). Con esto se ingresan los datos a una nueva clase de **WindTurbine** lista para utilizarse, con un diámetro de 126 metros, altura de buje de 90 metros y las curvas de potencia y CT que se muestran en la figura 10.

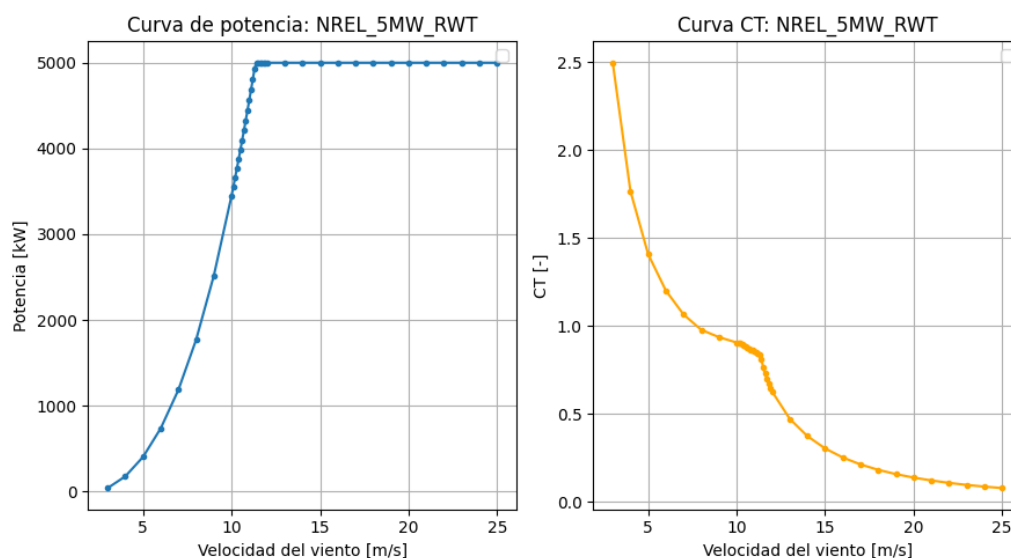


Figura 10: Curva de potencia (azul) y CT (naranja) de la turbina **NREL_5MW.126.RWT**(15), con un diámetro de rotor de 126[m] y una altura de buje de 90[m]. *Fuente: Elaboración propia mediante Python.*

- b) **Site:** Se ocupa la clase `UniformWeibullSite` utilizando datos de vientos descargados de ERA5 del dataset “ERA5 monthly averaged data on single levels from 1940 to present” (16). Estos son filtrados y procesados en MATLAB obteniendo los parámetros necesarios para definir el objeto.
- 1) **Datos de viento:** Para construir este objeto se obtienen datos mensuales de las componentes (u,v) de la velocidad del viento en el período 1940- 2022 a una altura de 100 metros, en la figura (11) se muestran los parámetros utilizados para la descarga de datos.

Open request form	
Request ID: 487a56dc-e071-4634-ae37-cdbdf50e3a39	
Product type:	Monthly averaged reanalysis
Variable:	100m u-component of wind, 100m v-component of wind
Year:	1940, 1941, 1942, 1943, 1944, 1945, 1946, 1947, 1948, 1949, 1950, 1951, 1952, 1953, 1954, 1955, 1956, 1957, 1958, 1959, 1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, 1971, 1972, 1973, 1974, 1975, 1976, 1977, 1978, 1979, 1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023
Month:	January, February, March, April, May, June, July, August, September, October, November, December
Time:	00:00
Sub-region extraction:	North -32°, West -76°, South -38°, East -70°
Format:	NetCDF (experimental)

Figura 11: Información de datos de viento descargados de ERA5 (16).

El procesamiento en MATLAB consiste en cargar el archivo “.nc”, filtrar entre las coordenadas [-36.75, -36.5] latitud y [-73.75, -73.25] longitud que encierra el área de emplazamiento (ver fig 12), calcular el ángulo azimuth y la magnitud de velocidad en cada instante de tiempo para posteriormente aplicar la función `wblfit` que obtiene los parámetros de forma y escala de Weibull, y también se debe obtener un vector con el sector de frecuencias de las direcciones. Mediante este procedimiento se obtiene el parámetro de escala y forma de la distribución de Weibull y el sector de frecuencias.

Los datos de vientos tienen una magnitud de velocidad promedio de 5.72 [m/s] a una altura de 100 [m] durante el periodo de tiempo, con una dirección promedio proveniente desde los 231.94 grados (o bien que se dirige en dirección a los 51.92 grados, Noreste). El 40 % mejor de los datos de viento cuenta con un promedio de 9.05 [m/s] y dirección proveniente de 204.78 grados.

Área de estudio, Granja Eólica Off-Shore [Zona 18H].

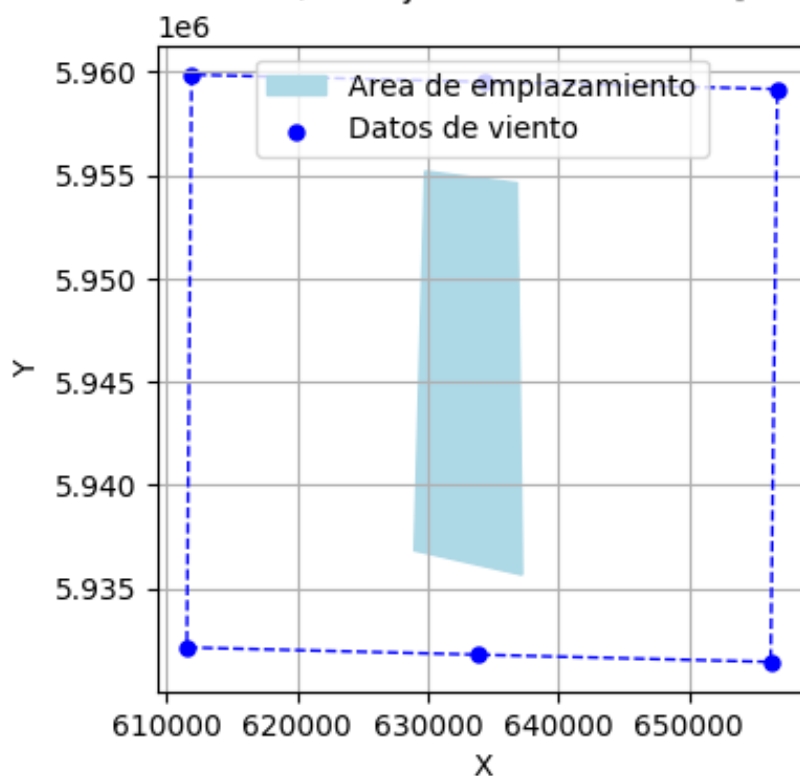


Figura 12: Puntos con datos de viento que encierran el área de emplazamiento con los cuales se calcula el promedio de magnitud de velocidad y dirección del viento para cada instante de tiempo.

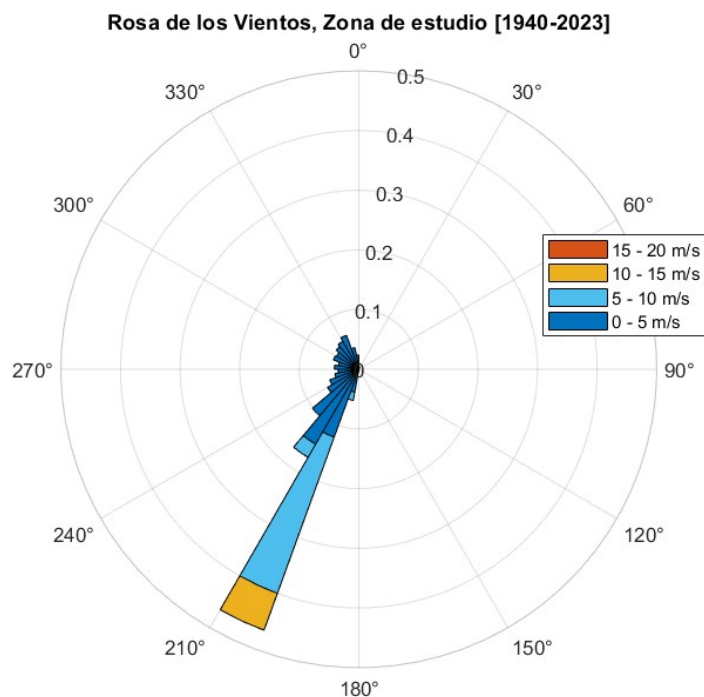


Figura 13: Rosa de los vientos de la zona de estudio entre 1940-2022 a una altura de 100 metros para el total de datos. Fuente: Elaboración propia mediante MATLAB

- 2) **Parámetros de Weibull:** La clase `UniformWeibullSite` de `PyWake` necesita como entrada los parámetros de forma y escala de la distribución de Weibull y un vector del sector de frecuencias de la dirección (desde donde proviene el viento) que nos entrega la función `wblfit` de MATLAB (17). Una vez obtenidos los parámetros estos se pueden visualizar con `PyWake` graficando el ajuste de la distribución de Weibull, la distribución de probabilidad de la velocidad según determinadas direcciones y nuevamente la rosa de los vientos.

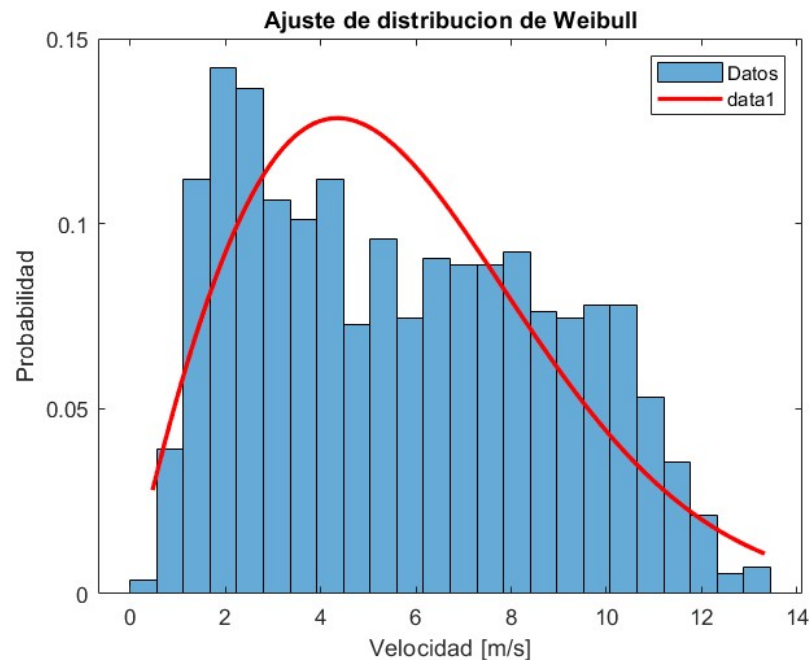


Figura 14: Ajuste de distribución de Weibull para los datos..

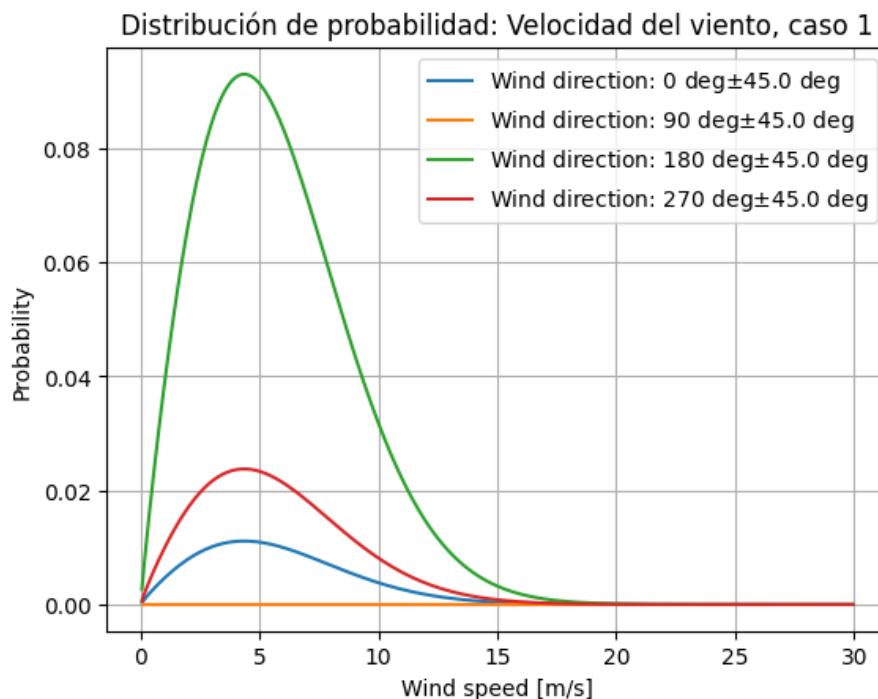


Figura 15: Gráfico de la distribución de probabilidades de la velocidad del viento en ciertas direcciones obtenida de acuerdo a los parámetros de Weibull.

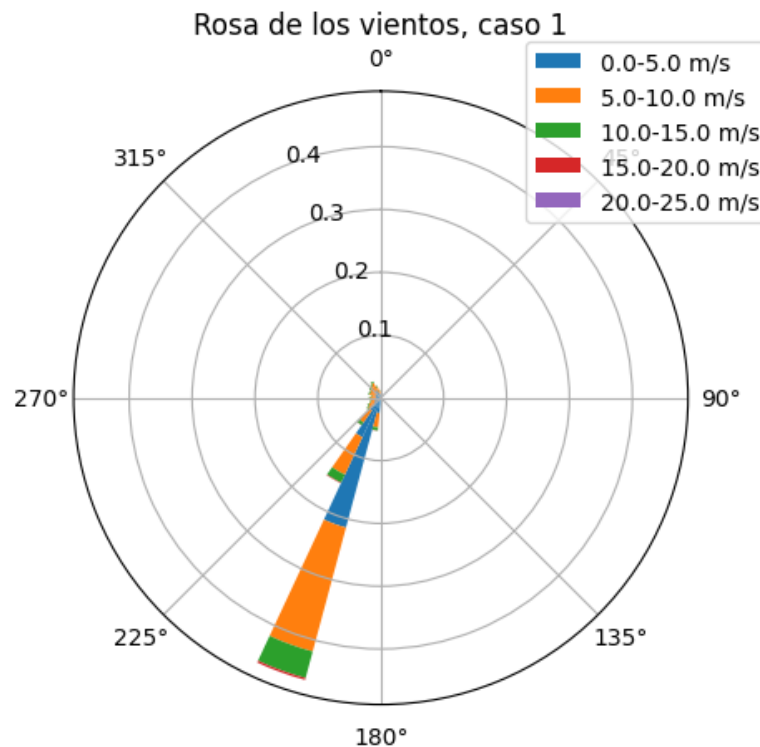


Figura 16: Rosa de los vientos de la zona de estudio entre 1940-2022 a una altura de 100 metros para el total de datos obtenida de acuerdo a los parámetros de Weibull.

c) **Wind farm model:**

El principal objeto que reúne los dos creados anteriormente es WindFarmModel. Empleando la turbina y Site creados además de seleccionar un modelo de ingeniería llamado All2AllIterative que añade el efecto de bloqueo en los aerogeneradores este objeto incorpora el modelo de déficit de estela FugaDeficit y un modelo LinearSum todo ello para simular el efecto estela aguas abajo que se produce cuando el viento pasa por cada uno de los aerogeneradores.

Aplicando la optimización

Se aplica el problema TopFarmProblem para maximizar el AEP de la granja eólica modificando las coordenadas de los aerogeneradores utilizando en su mayoría los parámetros por default del problema de TOPFARM, optimizador “EasyScipyOptimize” que utiliza el algoritmo “SLSQP” basado en gradiente scipy de forma predeterminada realizando 200 iteraciones y con parámetros como la tolerancia o `expect_cost` con valores donde el tiempo de espera no sea alto.

En el primer caso se realiza para todos los datos y direcciones de viento y en el segundo caso se realiza para el 40% mejor de los datos en un rango entre $[180, 225[$ grados.

Respecto a los parámetros más importantes se tiene:

- Variables de diseño: Corresponden a las ubicaciones iniciales aleatorias de los aerogeneradores mostradas anteriormente. A través de las distintas iteraciones que realiza TOPFARM se modifican y prueban diversas disposiciones de las turbinas para encontrar el mayor valor de AEP.
- Restricciones: Se utilizan los puntos límites (A, B, C, D) que delimitan un polígono que deca marca el área de emplazamiento (ver figura (9)).

Resultados

CASO 1: Totalidad de datos

a) Simulación con PyWake

La simulación da como resultados un AEP total de 169.724 [GWh] para el caso de la totalidad de datos de viento y los siguientes gráficos que ayudan a visualizar las propiedades y características de la granja simulada.

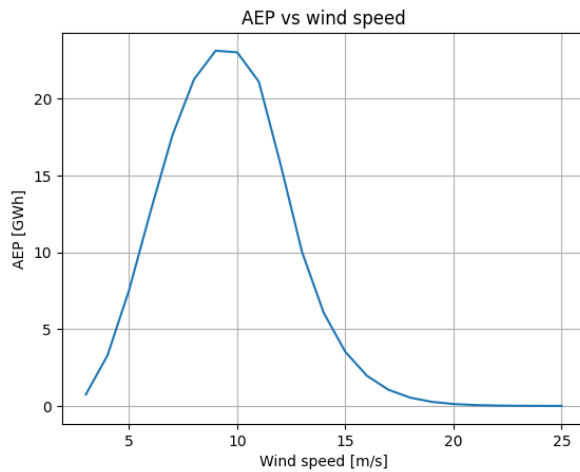


Figura 17: AEP en función de la velocidad del viento.

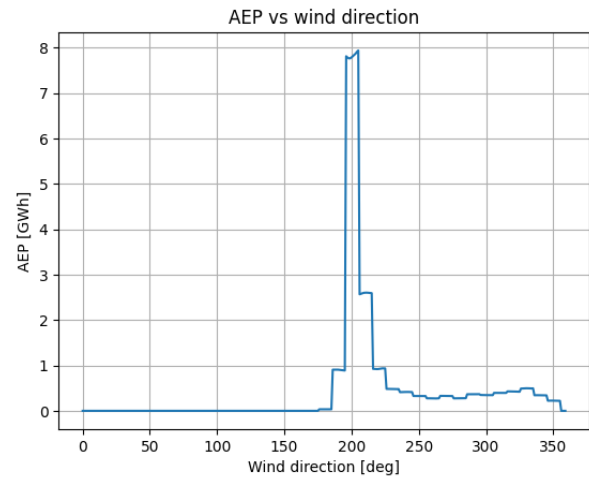


Figura 18: AEP en función de la dirección del viento.

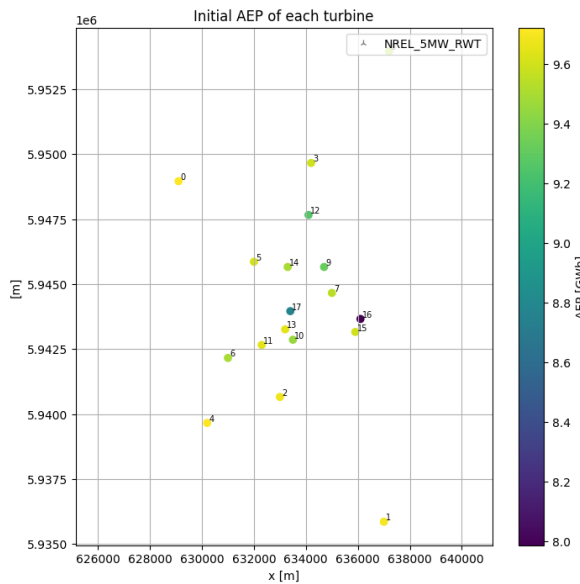


Figura 19: AEP de cada turbina, utilizando todo el rango de direcciones.

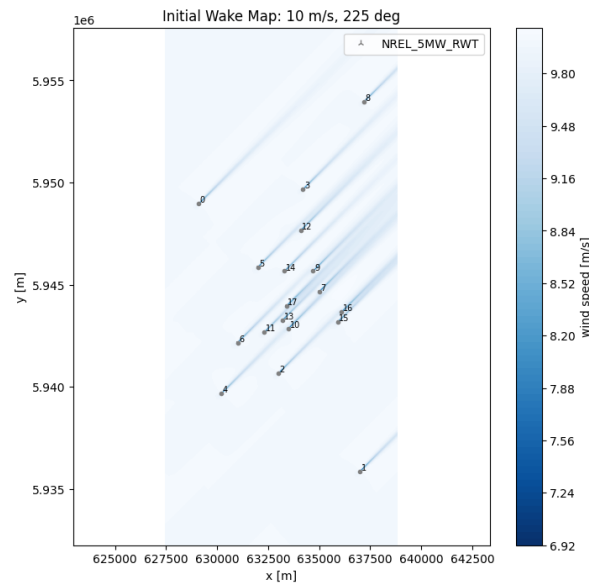


Figura 20: Mapa de flujo para visualizar la estela, caso con 10m/s y dirección del viento de 225 grados.

b) Optimización con TOPFARM

El AEP optimizado es de 174.36825 [GWh] el cual aumentó un 2.76 % respecto del valor inicial.

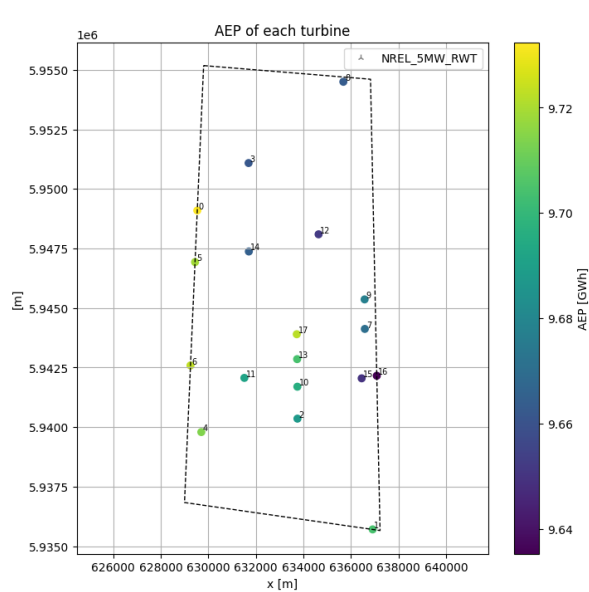


Figura 21: AEP de cada turbina, posiciones optimizadas.

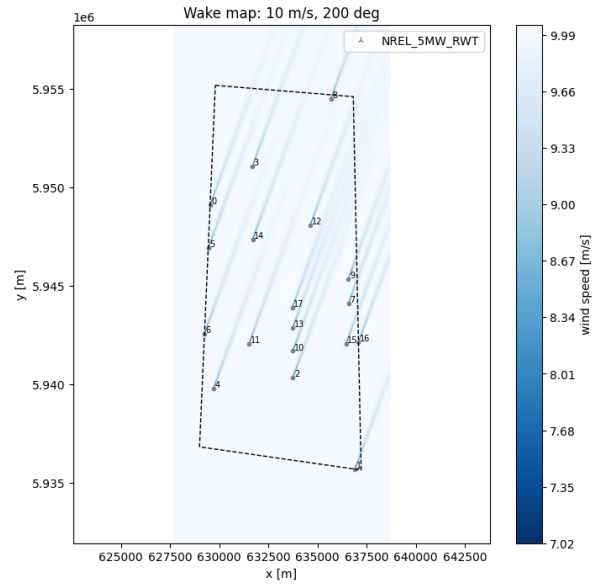


Figura 22: Mapa de flujo con las posiciones optimizadas, con 10m/s y dirección del viento de 200 grados

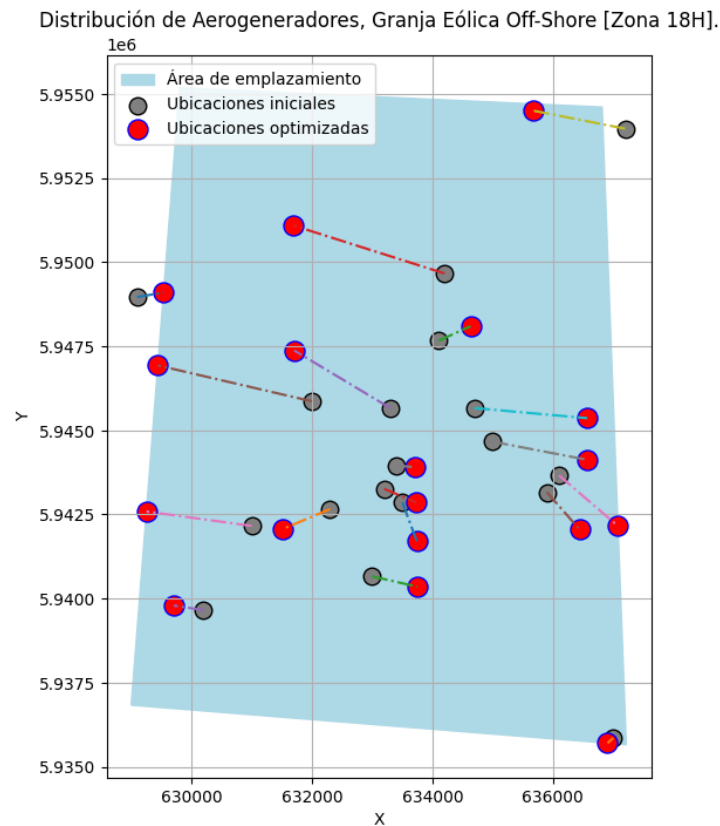


Figura 23: Distribución inicial y optimizada de los aerogeneradores de la granja eólica offshore.

CASO 2: 40 % mejor de los datos

a) Simulación con PyWake

Para este caso al utilizar el 40 % mejor de los datos de viento da un AEP total de 412.4495 [GWh]. Los primeros dos gráficos a continuación corresponden a las propiedades del Site y los cuatro restantes provienen de la simulación.

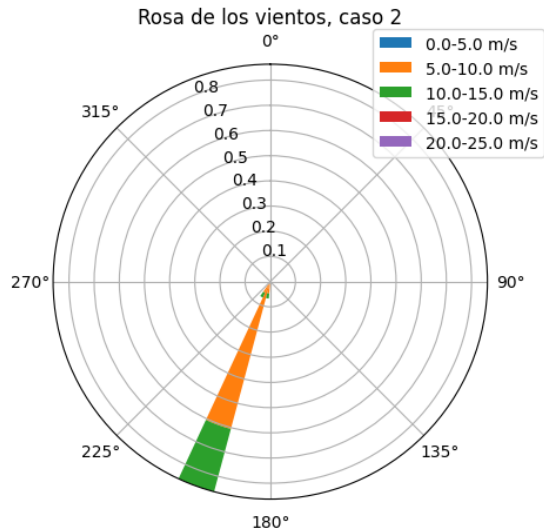


Figura 24: Rosa de los vientos, caso para el 40 % mejor de los datos de viento.

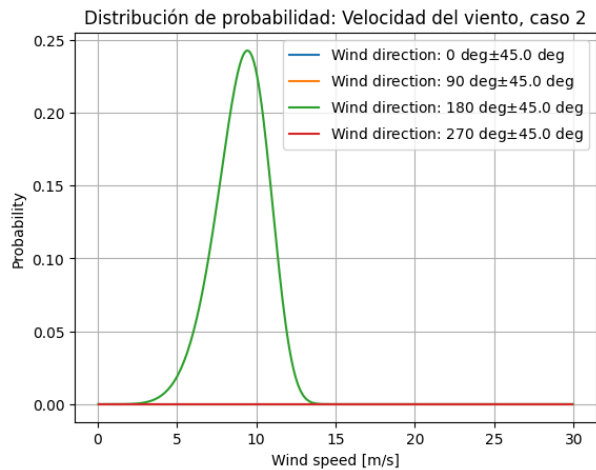


Figura 25: Distribución de probabilidades de la velocidad según direcciones para el caso 2.

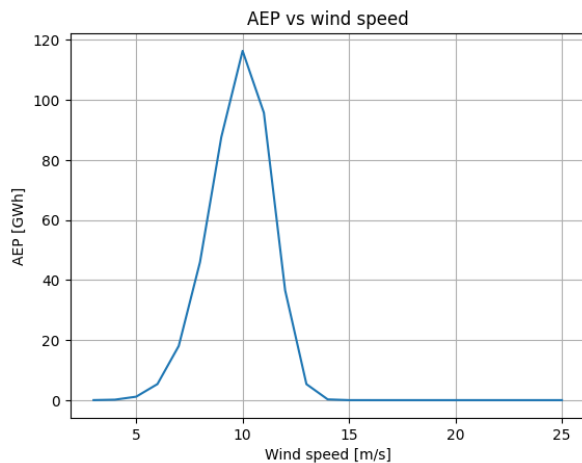


Figura 26: AEP en función de la velocidad del viento.

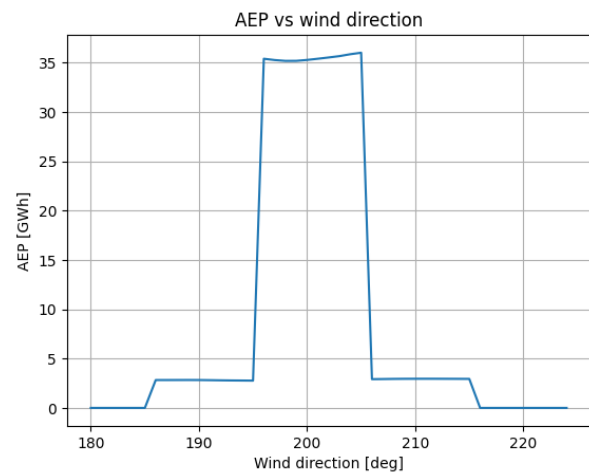


Figura 27: AEP en función de la dirección del viento.

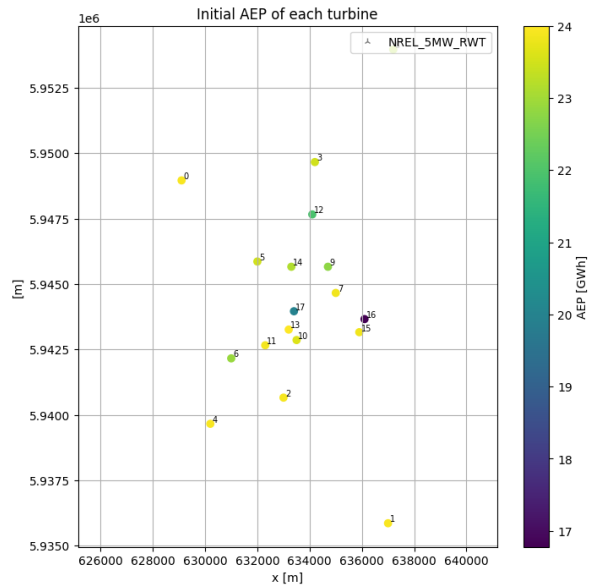


Figura 28: AEP de cada turbina, direcciones en el rango [180, 225].

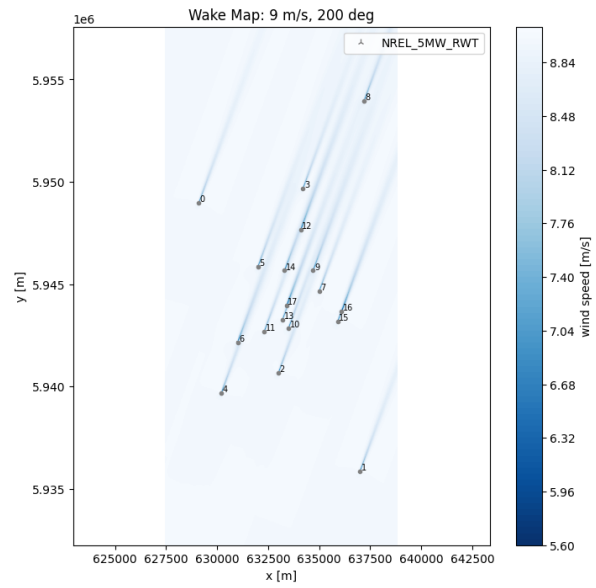


Figura 29: Mapa de flujo para visualizar la estela, con 10m/s y dirección de 200 grados

b) Optimización con TOPFARM

Luego de aplicar la optimización el AEP vale 431.106538 [GWh], aumentando su valor un 4.52 % del inicial.

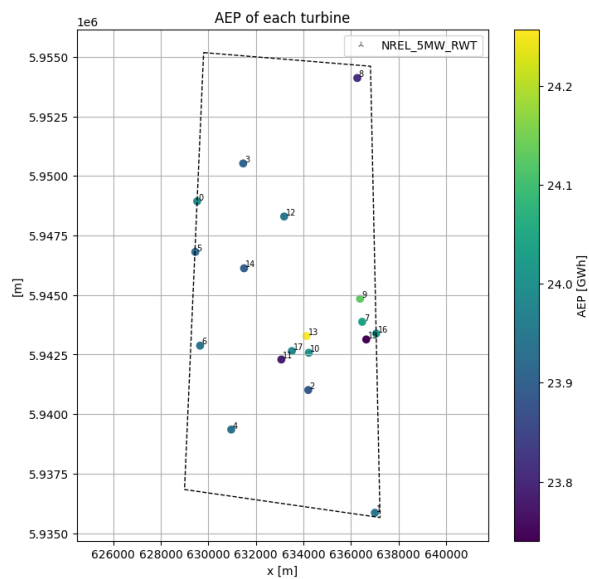


Figura 30: AEP de cada turbina con coordenas optimizadas.

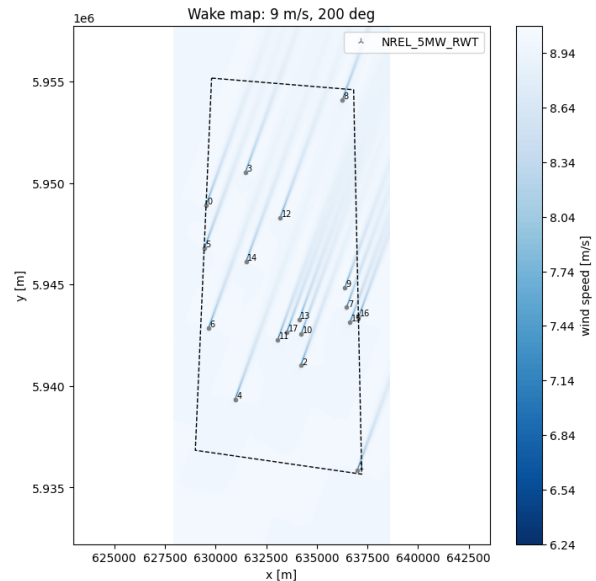


Figura 31: Mapa de flujo con coordenas optimizadas, con 9m/s y dirección del viento de 200 grados

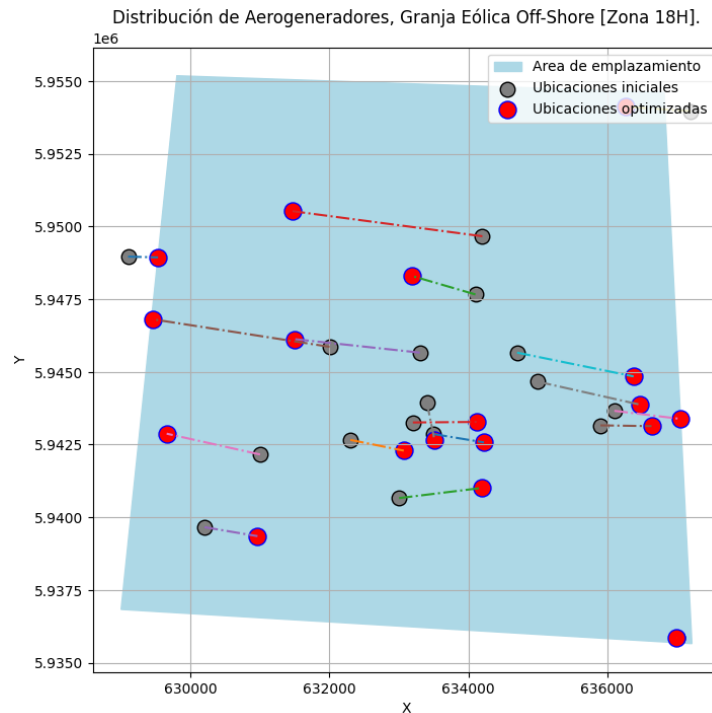


Figura 32: Distribución inicial y optimizada de los aerogeneradores de la granja eólica offshore, caso 2.

Discusión

De acuerdo a los resultados expuestos de AEPs iniciales y optimizados se observa que los porcentajes de aumentos de AEP que van entre el 1-5 % son valores bajos, de esta “situación” surgió la necesidad de corroborar si la turbina, Site y WindFarmModel estaban correctamente definidos implementándolos en otros ejemplos de los sitios oficiales de las librerías con resultados favorecedores de los objetos. Por lo que, en segunda instancia se modificaron parámetros de TopFarmProblem aumentando ligeramente los porcentajes de mejora de AEP, sin embargo mientras más precisos más tiempo de espera requerían y el aumento no era tan significativo. Finalmente se aumentó el número de aerogeneradores vislumbrando que el área de emplazamiento muy grande para el número inicial de aerogeneradores, por lo que estos al estar tan separados no interferían de mayor manera entre sí por ello las mejoras eran muy pequeñas. Al aumentar los aerogeneradores de 9 a 18 unidades se los porcentajes variaron desde los 0.5 al 5 %.

También el añadir un rango de direcciones en la instancia de optimización puede presentar mejores porcentajes de mejora, pues el posicionamiento de los aerogeneradores se centra en aprovechar aquellas direcciones de viento.

El hecho de implementar dos casos tiene el propósito de comparar las optimizaciones según los datos de viento usados debido a que al ser promedios mensuales los valores extremos de viento pueden disminuir en gran medida la magnitud de velocidad del viento, por ello se utiliza la totalidad y el mejor 40 % de ellos. Un futuro estudio podría estar enfocado en utilizar datos horarios, según los momentos con mayores velocidades de viento.

Por último, cabe mencionar que resultados más realistas pueden ser obtenidos aplicando más restricciones (separación entre turbinas, profundidades máximas de batimetría, entre otros.) y mejorando la definición del objeto Site, ya sea utilizando datos de un archivo en conjunto con la clase XRSite que es la más personalizable o añadiendo información de la instalación offshore (profundidad, lejaía a la costa, etc.).

Conclusión

Según los resultados y metodología utilizada se logró solucionar el problema de optimización de los aerogeneradores en la granja eólica de forma precisa, pero siguen existiendo diversas mejoras que podrían aplicarse para obtener porcentajes de aumento mayores. Aún así, los diferentes objetivos planteados fueron cumplidos logrando una familiarización con el tipo de energía, reforzando el lenguaje de programación Python y consiguiendo definir los objetos y parámetros necesarios resolviendo un problema de media complejidad.

Recalcando por último algunas ventajas de estas librerías se debe hacer mención a la gran cantidad de información y ejemplos en sus sitios oficiales que permiten un aprendizaje profundo de las funciones y módulos, también la gran utilidad de las restricciones que se pueden añadir haciendo más realista el problema y la versatilidad de los módulos pudiendo seleccionar opciones que acomoden al usuario. Siendo librerías de código libre y en continuo desarrollo aportan valiosas herramientas para la simulación y estudio de zonas para incorporar granjas eólicas.

Referencias

- [1] Carta González, J. A., Calero Pérez, R., Colmenar Santos, A., & Castro Gil, M.A. (2009). Centrales de energías renovables: generación eléctrica con energías renovables. Pearson Educación, S.A.
- [2] Tampier Brockhaus, G., Cifuentes Salazar, C., Aziares Aguayo, N., Cárdenas Cartes, R., Álvarez Castillo, K., & Pérez Pavez, A. (Eds.). (2024). Tecnologías de Energía Eólica Offshore para Chile: Perspectivas y Desafíos. Centro MERIC y Universidad Austral de Chile. Página oficial: <https://www.meric.cl>
- [3] Ferrari, R. (2023). Optimization of the spatial disposition of an array of floating offshore wind turbines using a Python genetic algorithm (Tesis de maestría). Politecnico di Torino.
- [4] Meteodyn. (s.f.). Meteodyn WFO. Meteodyn Universe. Recuperado el 05, enero 2024 de <https://meteodyn.com/business-sectors/renewable-energy/meteodyn-universe/meteodyn-wfo/>
- [5] National Renewable Energy Laboratory. (s.f.). SOWFA: Simulator for Offshore Wind Farm Applications. Recuperado el 05, enero 2024 de <https://www.nrel.gov/wind/nwtc/sowfa.html>
- [6] National Renewable Energy Laboratory. (2015). NREL Software Models Performance of Wind Plants: Highlights in Research and Development. Simulator for Wind Farm Applications helps optimize layouts and controls of wind plant arrays. Recuperado el 05, enero 2024 de <https://www.nrel.gov/docs/fy15osti/63378.pdf>
- [7] Mads M. Pedersen, Alexander Meyer Forsting, Paul van der Laan, Riccardo Riva, Leonardo A. Alcayaga Román, Javier Criado Risco, Mikkel Friis-Møller, Julian Quick, Jens Peter Schøler Christiansen, Rafael Valotta Rodrigues, Bjarke Tobias Olsen and Pierre-Elouan Réthoré. (2023, February). PyWake 2.5.0: An open-source wind farm simulation tool. DTU Wind, Technical University of Denmark. Recuperado el 05, enero 2024 de <https://topfarm.pages.windenergy.dtu.dk/PyWake/index.html>
- [8] Riccardo Riva, Jaime Yikon Liew, Mikkel Friis-Møller, Nikolay Krasimirov Dimitrov, Emre Barlas, Pierre-Elouan Réthoré, Mads Mølgaard Pedersen, “Welcome to TOPFARM”. Recuperado el 08 enero, 2024 de <https://topfarm.pages.windenergy.dtu.dk/TopFarm2/index.html>
- [9] Kunakote, T., Sabangban, N., Kumar, S., Tejani, G. G., Panagant, N., Pholdee, N., S.Bureerat & Yildiz, A. R. (2021), omparative Performance of Twelve Metaheuristics for Wind Farm Layout Optimisation, Archives of Computational Methods in Engineering, 1-14. Recupero el 08 enero, 2024 de <https://la.mathworks.com/matlabcentral/fileexchange/93315-wflo-code>
- [10] Gatscha S (2023). windfarmGA: Genetic Algorithm for Wind Farm Layout Optimization. R package version 3.1.0. Recuperado el 08 enero, 2024 de <https://ysosirius.github.io/windfarmGA/index.html>.
- [11] Link de acceso del perfil de GitHub: <https://github.com/TamiLatorre/Practica-Optimizacion-de-ubicaciones-de-aerogeneradores-en-una-Granja-Eolica-Offshore>
- [12] Aziares Aguayo, N., & Cárdenas Cartes, R. (2022). Sitios de interés para estudio de ciclo de vida con AdaptORE: Breve análisis de sitios de interés para proyectos ORE. Informe de Energía Marina SpA.
- [13] Google LLC. (2005). Google Earth. Fecha de acceso 18 enero, 2024 en <https://earth.google.com/>
- [14] SIRGAS Chile. (s.f.). Convertidor de coordenadas. Fecha de acceso 24, enero 2024. Enlace: <https://www.sirgaschile.cl/ConversionSC/Convertidor.php>

- [15] National Renewable Energy Laboratory. (s.f.). NREL_5MW_126_RWT.csv. Fecha de acceso 24, enero 2024. Enlace: https://github.com/NREL/turbine-models/blob/master/Offshore/NREL_5MW_126_RWT.csv
- [16] Copernicus Climate Change Service. (s.f.). Climate Data Store. Recuperado el 19 enero, 2024 de <https://cds.climate.copernicus.eu/>
- [17] MathWorks. wblfit. Recuperado de <https://la.mathworks.com/help/stats/wblfit.html>