

Universidad Internacional de La Rioja

Escuela Superior de Ingeniería y Tecnología

Universidad de Alcalá

Escuela de Posgrado

Universidad Rey Juan Carlos

Escuela de Másteres oficiales

Máster Universitario en Inteligencia Artificial para el Sector de
la Energía y las Infraestructuras

Diseño e implementación de sustratos en algoritmo CRO-SL para la distribución de aerogeneradores

Trabajo fin de estudio presentado por:	Romero Barrera, Antonio José
Tipo de trabajo:	Desarrollo software
Directora:	Cornejo Bueno, Laura María
Codirector	Casillas Pérez, David
Fecha:	13/07/2023

Esta obra está bajo una licencia Creative Commons “Atribución-
NoComercial-SinDerivadas 4.0 Internacional”.



Resumen

Los algoritmos metaheurísticos se han erigido como una forma eficaz para abordar problemas complejos en distintos campos. En este trabajo se pretende encontrar la distribución de turbinas eólicas que maximice la energía generada anualmente dentro de un parque eólico. Haciendo uso del algoritmo poblacional *Coral Reef Optimization with Substates Layer* (CRO-SL) e incorporando 3 sustratos adicionales basados en el operador 1Px y los algoritmos *Particle Swarm Optimization* (PSO) y *Whale Optimization* (WOA), se consigue una mejora en la tendencia de crecimiento de las soluciones y la calidad de los resultados obtenidos al distribuir aerogeneradores sobre un terreno discretizado de proporciones 30x30 de tipología *offshore*. Se han utilizado un total de 3 modelos de estela (Jensen, Larsen y Ainslie) que permiten estimar el déficit de velocidad que afecta a los aerogeneradores y calcular la energía generada a través de una curva de potencia de un aerogenerador AWEC-60.

Palabras clave: CRO-SL, metaheurísticas, optimización, granja eólica, algoritmos.

Abstract

Metaheuristic algorithms have emerged as an effective approach to address complex problems in various fields. This study aims to find the distribution of wind turbines that maximizes the annually generated energy within a wind farm. By using the population-based algorithm Coral Reef Optimization with Substates Layer (CRO-SL) and incorporating 3 additional substrates based on the 1Px operator and the Particle Swarm Optimization (PSO) and Whale Optimization (WOA) algorithms, an improvement in the growth trend of solutions and the quality of the obtained results is achieved when distributing wind turbines over a discretized offshore terrain of size 30x30. A total of 3 wake models (Jensen, Larsen, and Ainslie) have been used to estimate the velocity deficit affecting the wind turbines and calculate the generated energy through a power curve of an AWEC-60 wind turbine.

Keywords: CRO-SL, metaheuristics, optimization, wind farm, algorithm

Índice de contenidos

1. Introducción	15
1.1. Motivación	15
1.2. Planteamiento del trabajo	18
1.3. Planificación del trabajo	19
1.4. Estructura de la memoria	21
2. Contexto y estado del arte	23
2.1. Análisis de la literatura del algoritmo CRO	24
2.1.1. Justificación de las Preguntas de Investigación (RQ)	27
2.1.2. Resultados del proceso SLR	28
2.1.3. Respuesta a las Preguntas de Investigación (RQ) y discusión	29
2.2. Marco teórico	36
2.2.1. Aerogeneradores: propiedades y funcionamiento	36
2.2.2. Modelos de estela	40
2.2.3. Algoritmos metaheurísticos	47
3. Objetivos y metodología de trabajo.....	57
3.1. Objetivo general.....	57
3.2. Objetivos específicos	57
3.3. Metodología de trabajo	58
4. Identificación de requisitos	61
4.1. MATLAB: Sistema de cómputo numérico y entorno de programación.....	61
4.2. Datos climatológicos y curva de potencia	64
4.3. Función de <i>fitness</i> y modelos de estela de Jensen y Larsen.....	66
4.4. Programa de Algoritmo Genético Básico (BGA)	68

4.5.	Programa de <i>Coral Reef Optimization with Substrates Layer</i> (CRO-SL)	69
5.	Descripción de la herramienta software desarrollada.....	71
5.1.	Implementación del modelo de estela Ainslie	71
5.2.	Implementación de 1Px mutativo como sustrato	72
5.3.	Implementación de <i>Particle Swarm Optimization</i> (PSO) como sustrato.....	73
5.3.1.	Ajuste de hiperparámetros.....	73
5.3.2.	Adaptación del PSO como sustrato	75
5.4.	Implementación de <i>Whale Optimization Algorithm</i> (WOA) como sustrato	76
5.5.	Función generadora de terreno con desniveles y rugosidades superficiales.....	77
6.	Evaluación	79
6.1.	Comparativa de resultados del proceso de optimización	79
6.2.	Comparativa de la participación e importancia de cada sustrato.....	85
6.3.	Producto cartesiano de soluciones de CRO-SL de 7 sustratos	87
6.4.	Análisis de la afectación del relieve y los cambios de rugosidad	89
6.4.1.	Influencia de los desniveles del terreno en la energía generada.....	90
6.4.2.	Influencia de los cambios de rugosidad superficial en la energía generada.....	92
6.5.	Análisis del modelo de estela de Ainslie.....	94
7.	Conclusiones y trabajo futuro	97
7.1.	Conclusiones	97
7.2.	Consecución de objetivos	98
7.3.	Impacto social, económico y medioambiental del estudio	99
7.4.	Líneas de trabajo futuro	101
	Referencias bibliográficas.....	103
Anexo I.	Glosario de acrónimos	107
Anexo II.	Glosario de variables	109

Anexo III.	Listados de código	111
Anexo IV.	Valores de rugosidad superficial	121
Anexo V.	Valores de referencia de $\sigma\theta$	123
Anexo VI.	Tabla ponderativa entre MATLAB y Python	125

Índice de figuras

Figura 1.1. <i>Número de aerogeneradores operativos y energía eólica producida en España (2011-2021). Datos extraídos de (Statista Search Department, 2022).</i>	16
Figura 1.2. <i>Diagrama ejemplificativo del cono de turbulencia de los aerogeneradores.</i>	17
Figura 1.3. <i>Cálculo del espacio de búsqueda y metodologías descartadas.</i>	18
Figura 1.4. <i>Diagrama de Gantt del desarrollo del TFM.</i>	20
Figura 2.1. <i>Flujo de trabajo para el estudio de la literatura original.</i>	23
Figura 2.2. <i>Vertientes de investigación interrelacionadas encontradas en el estado del arte.</i>	24
Figura 2.3. <i>Diagrama circular de los campos de investigación que se usan el algoritmo CRO.</i>	29
Figura 2.4. <i>Publicaciones sobre el CRO por desde el año 2013 hasta junio del 2023.</i>	33
Figura 2.5. <i>Relación nodal entre autores de las publicaciones analizadas.</i>	35
Figura 2.6. <i>Valores del cT para distintos diseños y valores de TSR. Adaptada de (Hau, 2006).</i>	38
Figura 2.7. <i>Esquema de los componentes de una turbina eólica de eje horizontal.</i>	40
Figura 2.8. <i>Esquema de afectación de la velocidad dentro del cono de estela.</i>	41
Figura 2.9. <i>Clasificación de los algoritmos metaheurísticos. Adaptado de (Mirjalili et al., 2016).</i>	48
Figura 2.10. <i>Diagrama de funcionamiento de algoritmo CRO.</i>	49
Figura 2.11. <i>Esquema de la división de las larvas en capas de sustratos en algoritmo CRO-SL.</i>	50
Figura 2.12. <i>Diagrama de funcionamiento de BGA.</i>	52
Figura 4.1. <i>Curva de potencia original y aproximada de turbina eólica AWEC-60.</i>	65
Figura 4.2. <i>Representación de la metodología del operador 1Px.</i>	69
Figura 5.1. <i>Diagrama de funcionamiento del PSO como sustrato dentro del CRO-SL.</i>	75
Figura 5.2. <i>Diagrama de funcionamiento del PSO como sustrato dentro del CRO-SL.</i>	77

Figura 6.1. <i>Gráficas de evolución de la función de fitness en BGA.</i>	82
Figura 6.2. <i>Gráficas de evolución de la función de fitness en algoritmo CRO-SL (4 sustratos).</i>	83
Figura 6.3. <i>Gráficas de evolución de la función de fitness en algoritmo CRO-SL (7 sustratos).</i>	83
Figura 6.4. <i>Evolución de participación de los sustratos del CRO-SL (4 sustratos).</i>	86
Figura 6.5. <i>Evolución de participación de los sustratos del CRO-SL (7 sustratos).</i>	87
Figura 6.6. <i>Diagrama del Producto cartesiano de conjuntos $\{WM\}$ y $\{S\}$.</i>	88
Figura 6.7. <i>Representación del mapa de relieve utilizado.</i>	90
Figura 6.8. <i>Representación del mapa de rugosidad superficial utilizado.</i>	93

Índice de tablas

Tabla 1.1. <i>Tareas y fechas desglosadas de la planificación del TFM.</i>	21
Tabla 1.2. <i>Estructura del contenido principal de la memoria.</i>	22
Tabla 2.1. <i>String booleano de búsqueda.</i>	25
Tabla 2.2. <i>Número de artículos encontrados por plataforma de consulta.</i>	25
Tabla 2.3. <i>Protocolo SLR para el filtrado de artículos.</i>	26
Tabla 2.4. <i>Justificación de las preguntas de investigación (RQ).</i>	27
Tabla 2.5. <i>Variación del conjunto de publicaciones tras la aplicación de los criterios de exclusión y de inclusión.</i>	28
Tabla 2.6. <i>Número de publicaciones relacionadas con el CRO por campo de investigación.</i>	30
Tabla 2.7. <i>Afiliación institucional a universidades de los autores principales.</i>	34
Tabla 2.8. <i>Características de las turbinas eólicas de eje horizontal (HAWT) y vertical (VAWT).</i>	37
Tabla 2.9. <i>Componentes principales de una HAWT.</i>	39
Tabla 2.10. <i>Valores de los coeficientes de ajuste del modelo de estela de Larsen.</i>	45
Tabla 2.11. <i>Hiperparámetros característicos del WOA.</i>	54
Tabla 3.1. <i>Desglose de los objetivos específicos (OE).</i>	58
Tabla 3.2. <i>Comparativa de características entre el CFD y los modelos de estela.</i>	59
Tabla 4.1. <i>Comparativa de características entre MATLAB con Octave.</i>	62
Tabla 4.2. <i>Funciones y parámetros utilizados en la función de fitness.</i>	67
Tabla 5.1. <i>Rango de valores para el ajuste de hiperparámetros de movimiento del PSO.</i>	74
Tabla 5.2. <i>Valores de hiperparámetros del WOA.</i>	76
Tabla 6.1. <i>Características geométricas de los aerogeneradores y del entorno.</i>	79
Tabla 6.2. <i>Producción de energía máxima y mínima por modelo de estela.</i>	80
Tabla 6.3. <i>Valores de energía producida y rendimiento de las soluciones.</i>	81

Tabla 6.4. Distribuciones optimizadas de aerogeneradores por los algoritmos metaheurísticos.	84
Tabla 6.5. <i>Número de generaciones en las que cada sustrato ha generado la mejor larva. ...</i>	85
Tabla 6.6. <i>Participación de sustratos en la generación en la que se alcanzó el fitness máximo.</i>	86
Tabla 6.7. <i>Producto cruzado de las soluciones con los modelos de estela (fitness, kWh).</i>	88
Tabla 6.8. <i>Producto cruzado de las soluciones con los modelos de estela (rendimiento).</i>	89
Tabla 6.9. <i>Impacto del relieve en la energía producida por las distribuciones (modelo de Ainslie).</i>	91
Tabla 6.10. <i>Impacto de la longitud de rugosidad en la energía producida (modelo de Jensen).</i>	93
Tabla 6.11. <i>Tiempos de ejecución por modelo de estela en configuración de 30 aerogeneradores.</i>	95
Tabla 7.1. <i>ODS y metas específicas vinculadas al estudio realizado.</i>	101
Tabla 7.2. <i>Valores de rugosidad para distintas tipologías de terreno. Información obtenida de (Kumar, 2000).</i>	121
Tabla 7.3. <i>Valores de referencia de la fluctuación de la dirección del viento.</i>	123
Tabla 7.4. <i>Comparativa de características entre MATLAB y Python (NumPy).</i>	125

1. Introducción

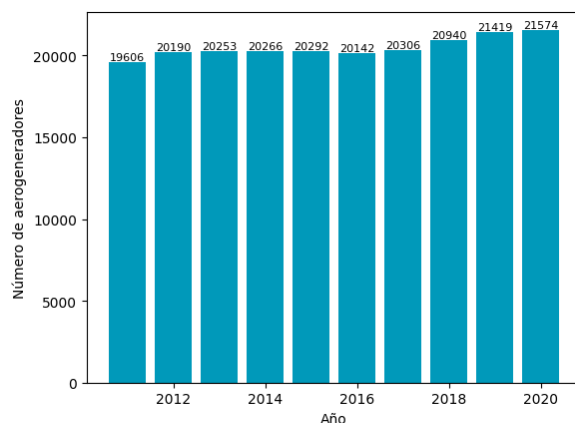
En este capítulo se introduce la temática del estudio que se va a realizar y las distintas problemáticas a las que se van a hacer frente durante el desarrollo de este trabajo. En primera instancia, en la Sección 1.1, se explica el estado actual del sector eólico y las acciones que están llevando a cabo varias instituciones para frenar la problemática de la contaminación. A continuación de ésta viene la Sección 1.2, donde se procede a explicar la metodología de trabajo a llevar a cabo y se argumentan ciertas decisiones tomadas para la elaboración de los análisis recogidos en el Capítulo 6. En la Sección 1.3 se elabora un diagrama de Gantt para organizar los plazos de las distintas tareas que se buscan ejecutar para realizar los distintos experimentos y redactar la memoria. Finalmente, la Sección 1.4 recoge una breve explicación de cada uno de los distintos capítulos que componen la memoria y otorga al lector una visión global de la estructura del documento.

1.1. Motivación

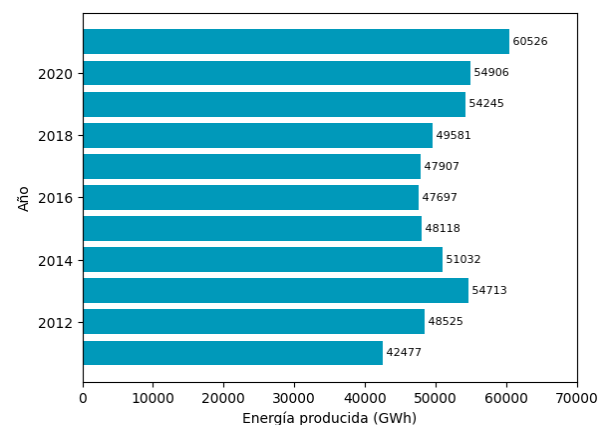
Nuestro mundo está en constante crecimiento, tanto a nivel tecnológico como a nivel poblacional, por lo que es necesario hacer un uso adecuado de las nuevas herramientas que surjan para mejorar la calidad de vida de los ciudadanos. En los últimos años, el planeta ha sufrido graves consecuencias debido a la acción humana como puede ser la contaminación, el calentamiento global derivado, la deforestación de varias zonas, la superpoblación o la pérdida de biodiversidad (Farfán Aguilar, 2009). Por estos motivos, instituciones como la Unión Europea (UE) o la Organización de Naciones Unidas (ONU) pretenden paliar estos efectos negativos con iniciativas como el Horizonte 2030 (del Río et al., 2017), la Coalición por la Neutralidad del Carbono (Naciones Unidas, n.d.) o los Objetivos de Desarrollo Sostenible (ODS). Las energías renovables y la concienciación de la población a la hora de gestionar las materias primas del planeta juegan un papel vital para conseguir un futuro adecuado para poder vivir.

La población actual del planeta se sitúa en torno a los ocho mil millones de personas, las cuales necesitan hacer uso de una serie de recursos para poder subsistir; y la extracción a gran escala de dichos recursos necesita grandes cantidades de energía. Conforme se vayan

agotando los recursos naturales fósiles debido a la extracción no controlada que se lleva a cabo a día de hoy, la energía generada por estos medios se encarecerá perjudicando a las compañías y sus consumidores. Como alternativa tenemos el uso de energías renovables, entre ellas la energía eólica que cuenta con una participación mayoritaria del 22% dentro del mercado de energías renovables, seguida de otras como las centrales de ciclo combinado con un 20% o la energía hidráulica con un 18%. España cuenta con más de 21500 turbinas eólicas que llegan a generar una energía anual superior a los 60 TWh, siendo regiones como Castilla y León (13793 GWh), Aragón (10200 GWh) y Galicia (9720 GWh) sus principales focos de producción según datos del 2022 consultados en (Orús, 2023). En la Figura 1.1 se pueden ver dos gráficos que ejemplifican la importancia que tiene la energía eólica en nuestro país y la gran inversión que realiza la industria eléctrica en ella, siendo la capacidad de potencia eólica instalada en España (29308 MW) la segunda más alta de Europa por detrás de Alemania (66315 MW), superando a otros países como Italia (11276 MW), Francia (21120 MW) o Reino Unido (28537 MW) según datos del año 2022 extraídos de (IRENA, 2023).



(a) Número de aerogeneradores



(b) Producción eólica anual

Figura 1.1. *Número de aerogeneradores operativos y energía eólica producida en España (2011-2021). Datos extraídos de (Statista Search Department, 2022).*

La instalación de turbinas eólicas para su uso en el sector energético conlleva una gran inversión de capital debido a todo el proceso de adquisición del aerogenerador, la logística del transporte y el ensamblaje en la zona donde se va a posicionar. Las pérdidas de energía generadas por una mala distribución de los aerogeneradores pueden afectar notablemente a la producción de energía de los parques eólicos, reduciendo la energía mecánica extraída del

aire y, por ende, la cantidad de electricidad generada para su posterior uso industrial o doméstico. Esta disminución en la producción puede suponer, con el paso de los años, pérdidas de ingresos bastante cuantitativas para las empresas y un retorno de la inversión inicial menor.

Por tanto, es importante, a la hora de optimizar el funcionamiento de los aerogeneradores de una zona, determinar correctamente la posición de los aerogeneradores en relación con el resto, de forma que se minimicen los efectos negativos producidos por las estelas generadas por el giro de los rotores de las turbinas eólicas de eje horizontal. Cuando el aire atraviesa el rotor de un aerogenerador se produce una alteración del flujo que perturba la estabilidad y dirección del viento, produciendo una turbulencia que desacelera la corriente aguas abajo. Este fenómeno turbulento queda contenido en una estela de geometría cónica, la cual delimita el efecto deficitario producido a dicho volumen de control. En la Figura 1.2 se puede observar una representación esquematizada del cono de estela generado por una turbina eólica y las modificaciones que sufre el aire contenido dentro de la estela.



Figura 1.2. Diagrama ejemplificativo del cono de turbulencia de los aerogeneradores.
Edición y fotografía propias. Parque eólico del municipio de Tébar, Cuenca.

Esta problemática ya fue tratada en el Trabajo Fin de Grado (Romero Barrera, 2023), sin embargo, al final del estudio realizado se establecieron una serie de líneas futuras de investigación, entre las cuales se han seleccionado 3 de ellas para establecer la temática de este Trabajo Fin de Máster (TFM), y poder así, continuar con una línea de investigación sobre la que ya se contaba con experiencia y conocimiento, pudiendo mejorar las tecnologías

desarrolladas previamente con nuevas funcionalidades y ampliar las conclusiones extraídas, a través de los nuevos resultados obtenidos, buscando intentar reforzarlas. El TFG previamente mencionado fue defendido el 22 de mayo de 2023 en la Universidad Rey Juan Carlos (URJC) y se calificó con una nota de sobresaliente, siendo propuesto para la asignación de mención honorífica del Grado de Ingeniería Aeroespacial en Vehículos Aeroespaciales.

1.2. Planteamiento del trabajo

La colocación de aerogeneradores sobre el terreno es un problema muy complejo y con una gran cantidad de combinaciones posibles, lo que hace que el espacio de búsqueda sea muy amplio y se dificulte trabajar con ciertas técnicas de optimización lineal, planificadores o fuerza bruta para obtener la mejor de las soluciones. Una buena alternativa para solucionar el problema es el uso de algoritmos metaheurísticos basados en poblaciones, que a pesar de que pueden alcanzar la mejor solución posible, sí consiguen soluciones óptimas muy cercanas a ella, lo que lo convierte en una opción muy a tener en cuenta para este tipo de problemas.

En la Figura 1.3 se puede observar una imagen representativa del proceso de cálculo del espacio de búsqueda del problema a través del cálculo combinatorio de posibles soluciones que puede tener el problema. Tras obtener una magnitud tan elevada de distribuciones de 30 turbinas sobre terrenos discretos con 900 posibles puntos de colocación, se descartaron posibles metodologías como el uso de planificadores, optimización lineal o fuerza bruta.

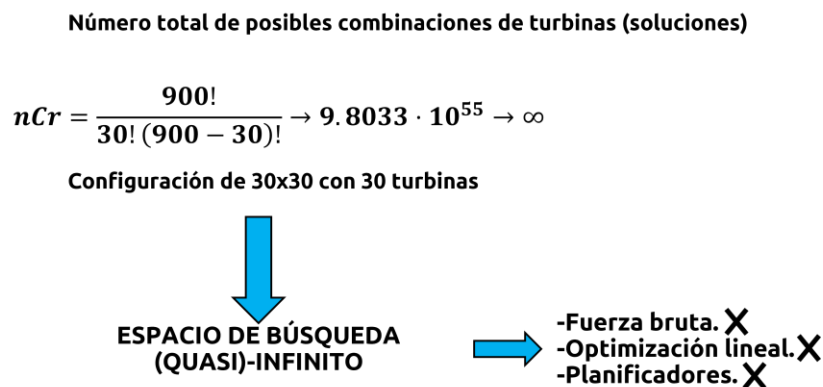


Figura 1.3. Cálculo del espacio de búsqueda y metodologías descartadas.

El algoritmo CRO-SL, al igual que otros algoritmos metaheurísticos basados en poblaciones, se utiliza para obtener soluciones óptimas a una gran variedad de problemas propuestos de diversa índole. Esto los convierte en una herramienta muy versátil y eficaz a la hora de plantear su uso en problemas con un gran espacio de búsqueda.

A pesar de su buen funcionamiento a la hora de encontrar soluciones óptimas, su tiempo de ejecución es notablemente alto y sus resultados no distan mucho de los obtenidos con otros algoritmos metaheurísticos empleados previamente (Romero Barrera, 2023). En este trabajo se quiere introducir tres sustratos adicionales que funcionen de manera más rápida, reduciendo así tiempos de cálculo y de ejecución que ralentizan el proceso. De igual forma, también se incorporará a la función de *fitness* del algoritmo la posibilidad de realizar predicciones de energía generada en parques eólicos de tipo terrestre (*onshore*) aplicando la capacidad de establecer desniveles y mapas de relieve a través de una función adicional.

Se estudiarán configuraciones con proporciones de terreno de 30x30 con 30 turbinas eólicas y vientos generados a partir de los vectores viento contenidos en un *dataset*, cuyos valores numéricos fueron recogidos de la provincia de Badajoz, datos con los que ya se contaba de otros análisis realizados previamente y que también fue utilizado en (Pérez-Aracil et al., 2022). Además, se utilizará un algoritmo genético básico (BGA, acrónimo de sus siglas en inglés) y el CRO-SL inicial para realizar una comparativa de funcionamiento con el CRO-SL modificado, y ver cuál de ellos domina en una configuración de aerogeneradores y proporciones de terreno intermedia entre las usadas en (Romero Barrera, 2023), donde se analizaron terrenos de proporciones de 20x20 con 20 turbinas y de 50x50 con 50 turbinas.

1.3. Planificación del trabajo

Se ha elaborado un diagrama de Gantt con el fin de organizar las tareas establecidas en la Sección 1.2 y los plazos correspondientes para su cumplimiento. La planificación que contiene el gráfico de la Figura 1.4, comprende desde el 25 de mayo de 2023 que se elaboró la hoja de ruta, hasta el 13 de julio de 2023, fecha en la cual se pretende finalizar la memoria, con el fin de realizar el predepósito del TFM en la convocatoria ordinaria.

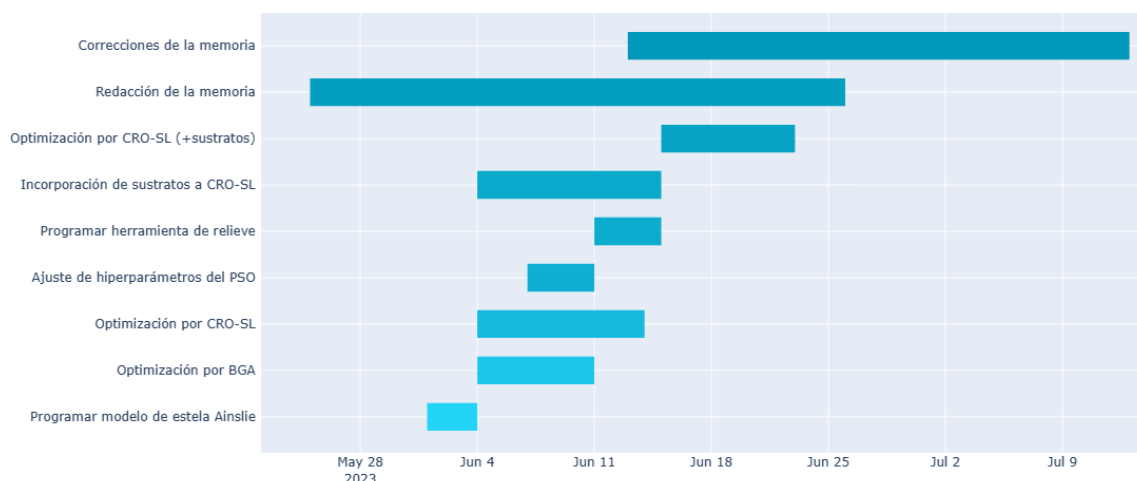


Figura 1.4. Diagrama de Gantt del desarrollo del TFM.

El contenido del diagrama de Gantt elaborado queda desglosado en la Tabla 1.1, donde se puede revisar las distintas fechas de inicio y finalización de las actividades propuestas para el cumplimiento de los objetivos establecidos en el Capítulo 1.

Se inicia la redacción de los capítulos introductorios de la memoria el 25 de mayo, lo cual fue 3 días después de la defensa del TFG que precede a este estudio. Tras ésta, se emplaza una reunión el 1 de junio, cuyo objetivo era establecer la hoja de ruta y las fechas claves para la correcta elaboración del trabajo.

En primera instancia, se procede al proceso de estudio del funcionamiento del nuevo modelo estela a incorporar, el modelo de Ainslie (descrito en la Sección 2.2.2.3 del Capítulo 2), y su posterior codificación, en forma de función, para ser llamada por los algoritmos al ejecutar la función de *fitness* para el cálculo del déficit de velocidades y la consecuente pérdida de energía producida. Una vez obtenido, se procederá entre los días 4 de junio hasta el 11 de junio (BGA) y 14 de junio (CRO-SL), a ejecutar la optimización de distribución de turbinas con este nuevo modelo (Ainslie) y otros dos modelos implementados en trabajos anteriores a modo comparativo (Jensen y Larsen).

Una vez finalizados algunos análisis de la primera tanda, del 11 al 15 de junio se elabora un pequeño script que permite generar una matriz de altitudes que simulen los desniveles del terreno y la adaptación de la función de *fitness* para que los modelos de estela hagan uso de ella para sus cálculos. En paralelo, entre el 4 y el 15 de junio, se realiza la adaptación e incorporación del algoritmo PSO como sustrato, de tal forma que se pueda ejecutar adecuadamente con la estructura definida por el algoritmo CRO-SL. A razón de los resultados

obtenidos en (Romero Barrera, 2023) por el PSO a la hora de optimizar las distribuciones, es necesario realizar un proceso de ajuste de hiperparámetros (*tuning*) para el problema planteado, por este hecho, entre el 7 y 11 de junio se elabora un *script* que permite la ejecución controlada del algoritmo PSO, modificando los parámetros a ajustar pasados un número establecido de iteraciones.

A partir del día 15 de junio se empieza con la última tanda de análisis para los tres modelos de estela escogidos, esta vez elaborados con el CRO-SL ya con los 2 sustratos nuevos implantados. Se estima que estos análisis finalicen en torno al 23 de junio, tiempo suficiente para incorporar las gráficas y valores conseguidos al informe, para ya dar finalización a la parte de redacción tras añadir las conclusiones tras el análisis y la comparativa de resultados. Una vez realizado la entrega del borrador final, se realizarán una serie de correcciones menores hasta el día del predepósito el 13 de julio.

Tabla 1.1. *Tareas y fechas desglosadas de la planificación del TFM.*

ACTIVIDAD	FECHA DE INICIO	FECHA DE FINALIZACIÓN
Redacción de la memoria.	25/05/2023	26/06/2023
Programar modelo de estela Ainslie.	01/06/2023	04/06/2023
Optimización por BGA (Jensen, Larsen, Ainslie).	04/06/2023	11/06/2023
Optimización por CRO-SL (Jensen, Larsen, Ainslie).	04/06/2023	14/06/2023
Incorporación de sustratos a CRO-SL (1Px, PSO y WOA).	04/06/2023	15/06/2024
Ajuste de hiperparámetros del PSO.	07/06/2023	11/06/2023
Programar herramienta de relieve.	11/06/2023	15/06/2023
Optimización por CRO-SL (nuevos sustratos).	15/06/2023	23/06/2023
Correcciones de la memoria.	13/06/2023	13/07/2023

1.4. Estructura de la memoria

En esta sección, se define la estructura de la memoria y se describe brevemente la temática de los distintos capítulos que la componen. La memoria de este TFM comprende un total de 7 capítulos que forman parte del contenido principal desarrollado, la bibliografía donde se recogen las distintas fuentes de información consultadas, 6 anexos que contienen datos que complementan el estudio desarrollado y 2 resúmenes del trabajo en español e inglés, expuestos en formato de artículo de investigación a doble columna. En la Tabla 1.2 se

desglosan las distintas partes que componen la memoria junto con una breve descripción de su temática.

Tabla 1.2. *Estructura del contenido principal de la memoria.*

CAPÍTULO Y TÍTULO	DESCRIPCIÓN
Capítulo 1: Introducción	En este capítulo se introduce el marco teórico del trabajo, el cual reúne los conceptos básicos de los aerogeneradores, del algoritmo CRO y de los modelos de estela. También se define la motivación, el planteamiento, la planificación y la estructura del trabajo.
Capítulo 2: Contexto y estado del arte	Se definen brevemente el conocimiento conseguido sobre la temática en estudios y trabajos previos, y posteriormente se amplía esa base de conocimiento con un estudio más específico de la literatura sobre el algoritmo CRO a través de una metodología SLR.
Capítulo 3: Objetivos y metodología de trabajo	Se plasman los objetivos generales y específicos con propiedades SMART y se explica el flujo de trabajo necesario para poder cumplirlos satisfactoriamente.
Capítulo 4: Identificación de requisitos	Análisis sobre los requerimientos necesarios para poder llevar a cabo el plan de trabajo establecido y desarrollar las nuevas herramientas ideadas.
Capítulo 5: Descripción de la herramienta software utilizada	Exposición detallada de la herramienta desarrollada y las novedades que aporta para resolver la problemática de la distribución de aerogeneradores sobre el terreno.
Capítulo 6: Evaluación	Debate y comparativa de los resultados de distribuciones de turbinas obtenidos tras la ejecución del algoritmo y análisis de las métricas de funcionamiento establecidas para medir la ratio de mejora en el algoritmo CRO-SL.
Capítulo 7: Conclusiones y trabajo futuro	Conclusiones extraídas tras el análisis de los datos y las vías de investigación que se puede llevar a cabo posteriormente relacionadas con la temática del estudio, con el fin de ampliar el número de estudios sobre esta temática y aportar valor al campo de investigación.
Bibliografía	Contiene todas las referencias y fuentes de consulta mencionadas en la memoria.
Anexos	Conjunto de anexos que cuentan con los glosarios de acrónimos y variables, la tabla con los valores de referencia de la longitud de rugosidad superficial y los fragmentos de código desarrollados para elaborar la memoria.
Artículos de investigación	Resumen de la memoria en forma de artículo de investigación.

2. Contexto y estado del arte

Los algoritmos metaheurísticos son ampliamente utilizados en la resolución de problemas complejos, y en particular, en el campo de la optimización de la colocación de aerogeneradores sobre el terreno correspondiente a un parque eólico.

En trabajos previos (Romero Barrera, 2023), se realizó un análisis de la literatura exhaustivo sobre el uso de los algoritmos metaheurísticos empleados en la optimización de los procesos relacionados con la energía eólica. En dicho estudio, se partió del artículo (Pérez-Aracil et al., 2022), se ejecutó un proceso de *snowballing* de una capa *forward* y *backward*, en el que se recogieron las *keywords* de los artículos que fueron citados en el artículo de referencia y los que lo citaron. Con estas *keywords* se diseñó un *string* booleano de búsqueda que fue introducido en las opciones avanzadas de búsqueda de Google Scholar y Scopus. Finalmente, se encontraron un total de 1112 artículos (885 en Google Scholar y 227 en Scopus), los cuales fueron filtrados y seleccionados a través de un protocolo de *Systematic Literature Review* (SLR), siguiendo el flujo representado en el diagrama mostrado en la Figura 2.1.

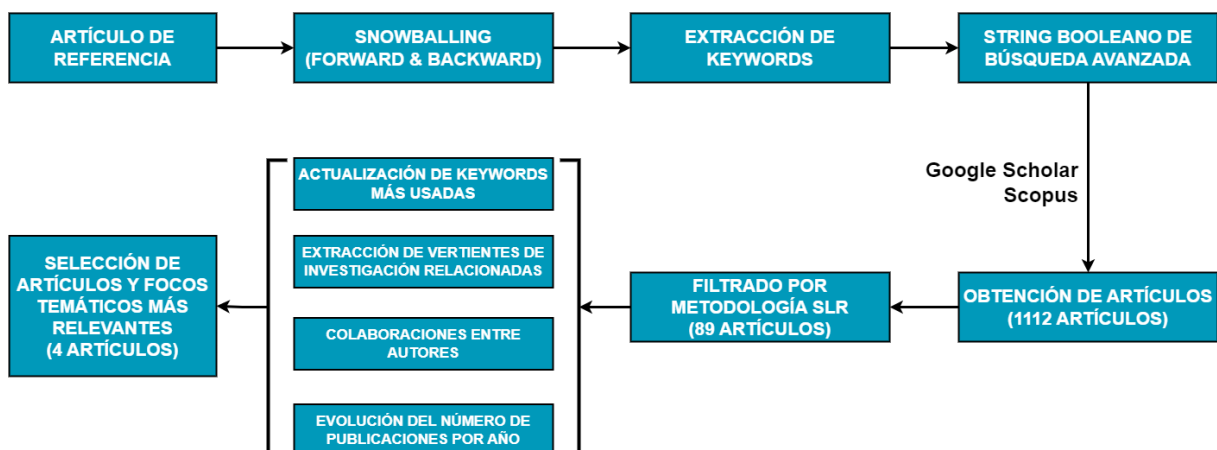


Figura 2.1. Flujo de trabajo para el estudio de la literatura original.

A partir de los datos obtenidos de los artículos y el posterior filtrado de éstos por la metodología SLR, se sacaron una serie de conclusiones en relación con las vertientes de investigación relacionadas con la aplicación de herramientas de IA en el estudio de las turbinas eólicas, las distintas colaboraciones entre autores, la tendencia de crecimiento del número de artículos publicados por año desde 2013 hasta 2022 y las *keywords* más utilizadas entre los artículos filtrados.

Este TFM se centra en la última de las vertientes detectadas, la cual se encarga de estudiar la distribución de los aerogeneradores sobre terrenos ya delimitados, que representan el parque eólico de interés. Otras vertientes encontradas fueron el uso de herramientas de IA como el *Machine Learning* (ML), distintos tipos de redes neuronales (RN) o algoritmos metaheurísticos, a la hora de seleccionar en grandes extensiones de terrenos la mejor zona para la construcción del parque eólico en función de las condiciones de relieve y atmosféricas, o la optimización geométrica de los perfiles alares de las palas del rotor para aumentar su eficiencia y, por ende, la energía producida por cada aerogenerador.

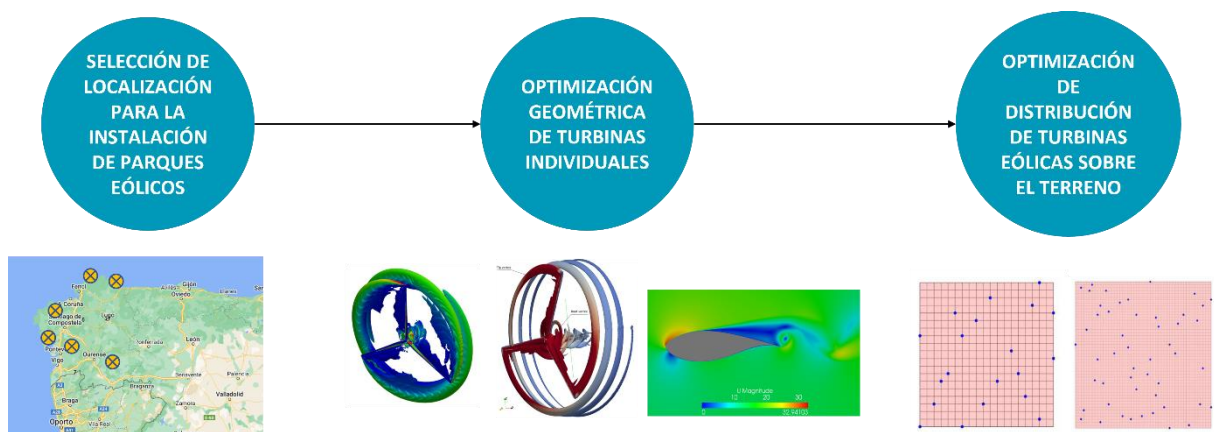


Figura 2.2. Vertientes de investigación interrelacionadas encontradas en el estado del arte.

2.1. Análisis de la literatura del algoritmo CRO

Partiendo del estudio mencionado a modo introductorio en el Capítulo 2, se procede a estudiar más en detalle la literatura científica relacionada con el algoritmo CRO-SL específicamente. Debido a que el algoritmo CRO es de creación bastante reciente, en la actualidad no existe una gran cantidad de artículos, por lo que la búsqueda se ha realizado manualmente, sin necesidad de realizar un parseo, a través de la base de datos de artículos científicos de Google Scholar, Scopus, IEEE Explorer y Web of Science. Se aplicará un protocolo *Systematic Literature Review* (SLR) para llevar a cabo un análisis de la literatura sobre el estado actual de las investigaciones efectuadas en torno al algoritmo CRO y su capacidad a la hora de obtener soluciones óptimas para los distintos problemas planteados.

Para iniciar la búsqueda, y dado que queremos obtener una visión global de los usos del algoritmo, se ha elaborado un *string* booleano de búsqueda con *keywords* generales sobre él,

así como ciertas *keywords* que describen el objetivo de uso o la familia de algoritmos a la que pertenece. Dado que los algoritmos metaheurísticos se basan en procesos observados en la naturaleza y muchas veces se les atribuye el nombre con relación a ello, estas últimas *keywords* han servido para eliminar de la búsqueda algunos artículos relacionados con los arrecifes de coral en el campo de la biología.

Tabla 2.1. *String booleano de búsqueda.*

("coral reef optimization" OR "CRO") AND ("optimization" OR "CRO-SL" OR "metaheuristics" OR "ensemble methods" OR "algorithm").

Tras elaborar la búsqueda, se han obtenido un total de 1389 artículos, cuya distribución cuantitativa de origen de las distintas plataformas queda recogida en la Tabla 2.2. Se puede observar que las bases de datos de Scopus (39.3%) y Web of Science (43%) albergan la mayor parte de los artículos reunidos, mientras que Google Scholar (8%) o IEEE Explorer (9.7%) representan la minoría de ellos. Hay que mencionar que Google Scholar no permite la descarga directa de un fichero CSV con los datos de la búsqueda sin previamente guardarlos en una lista de lectura, y para evitar realizar un parseo web, se ha optado por la selección manual de artículos resultantes de la búsqueda, lo que ha reducido bastante la cantidad de artículos final.

Tabla 2.2. *Número de artículos encontrados por plataforma de consulta.*

BASE DE DATOS	CANTIDAD DE ARTÍCULOS
Google Scholar	111
Scopus	546
IEEE Explorer	135
Web of Science	597
TOTAL	1389

Todos estos resultados se han unificado en un solo fichero de formato *spreadsheet*, con el fin de facilitar su análisis, y posteriormente, ejecutar un protocolo SLR que permite la selección de artículos de manera adecuada. En la Tabla 2.3 se recogen los criterios de inclusión, exclusión y de calidad seleccionados, así como las preguntas que se busca resolver tras el análisis de los distintos artículos filtrados.

Tabla 2.3. *Protocolo SLR para el filtrado de artículos.*

Preguntas de investigación (RQ)	<p>RQ1. ¿En qué ámbitos se ha aplicado el algoritmo CRO o alguna de sus modificaciones para la obtención de soluciones?</p> <p>RQ2. ¿Qué modificaciones han surgido del algoritmo base?</p> <p>RQ3. ¿Cómo es la tendencia de crecimiento del número de publicaciones por año relacionadas con el CRO?</p> <p>RQ4. ¿Qué investigadores han colaborado en el desarrollo de artículos relacionados con el CRO y con quiénes han colaborado?</p>
String de búsqueda	'coral reef optimization' AND ('CRO-SL' OR 'CRO' OR 'metaheuristics' OR 'optimization' OR 'algorithm').
Metadatos de la búsqueda	Título, <i>Abstract</i> , <i>Keywords</i> .
Bases de datos consultadas	Google Scholar, Scopus, IEEE Explorer, Web of Science.
Criterios de inclusión (IC)	<p>IC1. Documentos que hacen uso del algoritmo CRO o alguna de sus modificaciones.</p> <p>IC2. Documentos que hagan referencia al CRO, aunque no sea la herramienta que se emplea principalmente para el estudio.</p>
Criterios de exclusión (EC)	<p>EC1. Documentos no escritos en inglés.</p> <p>EC2. Documentos de menos de 4 páginas.</p> <p>EC3. Documentos duplicados en las búsquedas.</p> <p>EC4. Publicaciones ajenas al algoritmo <i>Coral Reef Optimization</i>.</p> <p>EC5. Publicaciones previas al año 2013.</p>
Criterios de calidad (QC)	<p>QC1. Grupo de clasificación de la revista donde se realiza la publicación.</p> <p>QC2. Relacionado con el sector de la energía o las infraestructuras.</p> <p>QC3. Número de colaboraciones, publicaciones y citas de los autores.</p>

2.1.1. Justificación de las Preguntas de Investigación (RQ)

Con el fin de demostrar la validez y utilidad de cada una de las preguntas de investigación (RQ, acrónimo de sus siglas en inglés) que fueron elaboradas, se ha procedido a la justificación y argumentación de cada una de ellas en la Tabla 2.4. Posteriormente, en la Sección 2.1.3, se dan respuesta a cada una de las preguntas, y se exponen algunas conclusiones extraídas tras la realización del proceso SLR planteado.

Tabla 2.4. *Justificación de las preguntas de investigación (RQ).*

Justificación RQ1: ¿En qué ámbitos se ha aplicado el algoritmo CRO o alguna de sus modificaciones para la obtención de soluciones?

En función del problema a tratar y de la forma que tengan las soluciones, será más efectivo el uso de uno u otro algoritmo metaheurístico debido a la forma de abordar la búsqueda de resultados. Es importante conocer el tipo de problemas en los que se ha utilizado el algoritmo CRO, con el fin de conocer en qué ámbitos funciona mejor y la forma en la que se analizan los individuos de la población.

Justificación RQ2: ¿Qué modificaciones han surgido del algoritmo base?

Desde su creación en el año 2013, el algoritmo CRO ha sufrido importantes modificaciones que mejoraban su funcionamiento y rendimiento, así como adaptaciones que permitieran operar con él ante problemas de distinta índole. Se busca conocer las vertientes más relevantes del CRO surgidas a lo largo de los años.

Justificación RQ3: ¿Cómo es la tendencia de crecimiento del número de publicaciones por año relacionadas con el CRO?

Esta información permite conocer el nivel de interés de la comunidad científica en relación con el CRO, así como el grado de uso de éste con respecto a otros algoritmos metaheurísticos poblacionales empleados en la problemática planteada.

Justificación RQ4: ¿Qué investigadores han elaborado artículos relacionados con el CRO y con quiénes han colaborado?

Conocer a los autores de una temática permite identificar los focos de investigación más importantes y participativos dentro de la investigación y el desarrollo del tema a tratar. En este caso se busca conocer qué autores e instituciones han aportado al desarrollo tecnológico del CRO.

2.1.2. Resultados del proceso SLR

El cómputo total de artículos conseguidos tras la búsqueda es de 1389 publicaciones entre las 4 bases de datos consultadas. Debido a la magnitud del conjunto de resultados conseguidos, es necesario aplicar una serie de filtros regidos por los criterios establecidos en la Tabla 2.3 para menguar la cantidad de publicaciones a analizar.

Se comienza aplicando los tres primeros criterios de exclusión a la vez (EC1, EC2 y EC3), cuya implementación es relativamente sencilla y automática, para eliminar la mayor cantidad de publicaciones posibles antes de establecer los siguientes criterios, pasando de 1389 resultados de búsqueda a tan sólo 887. Posteriormente, tras la aplicación del EC5, se detectan 163 artículos que se han publicado en años anteriores al 2013, dejando el total de publicaciones en 724. A continuación, se procede a comprobar manualmente la temática de los artículos y las herramientas que utilizan, descartando aquellos que no hagan uso del algoritmo CRO para sus experimentos, lo cual llevó a la eliminación de 609 resultados, resultando en un total de 115 publicaciones. Finalmente, bajo el amparo de los criterios IC1 e IC2, se optó por añadir manualmente la publicación propia (Romero Barrera, 2023) al conjunto de resultados de búsqueda, haciendo que el conjunto final sea de 116 publicaciones.

Tabla 2.5. *Variación del conjunto de publicaciones tras la aplicación de los criterios de exclusión y de inclusión.*

Nº. PUBLICACIONES	EC1 + EC2 + EC3	EC5	EC4	IC1 + IC2
Número inicial	1389	887	724	115
Modificados (+/-)	-502	-163	-609	+1
Número final	887	724	115	116

Cabe destacar que durante la aplicación del criterio EC4 se detectaron ciertos fallos en el *string* booleano de búsqueda, que provocó la aparición de artículos relacionados con otros algoritmos o métodos de optimización, cuyos acrónimos coincidían con el del CRO, tales como *Chemical Reaction Optimization*, *Coot Remora Optimization*, *Conversion Rate Optimization* o

Cognitive Routing Optimization. Este hecho amplió en exceso los resultados de búsqueda iniciales y ralentizó el análisis de publicaciones final.

2.1.3. Respuesta a las Preguntas de Investigación (RQ) y discusión

En esta sección se procede a responder a las distintas RQ planteadas en la Tabla 2.3 de la Sección 2.1, con el fin de aportar una visión global del estado de las distintas investigaciones llevadas a cabo haciendo uso del algoritmo CRO o derivados de él. En las respuestas se aportan una serie de gráficas y tablas de interés que apoyan a los datos y las conclusiones extraídas del análisis de las publicaciones.

RQ1: ¿En qué ámbitos se ha aplicado el algoritmo CRO o alguna de sus modificaciones para la obtención de soluciones?

Conocer los campos de aplicación del algoritmo CRO puede ayudar a conocer cómo de apto es su funcionamiento a la hora de afrontar determinadas problemáticas. Una vez obtenidos los 116 artículos finales, se han extraído de ellos las temáticas generales de uso a las que pertenecían, las cuales quedan registradas de forma porcentual en la Figura 2.3.

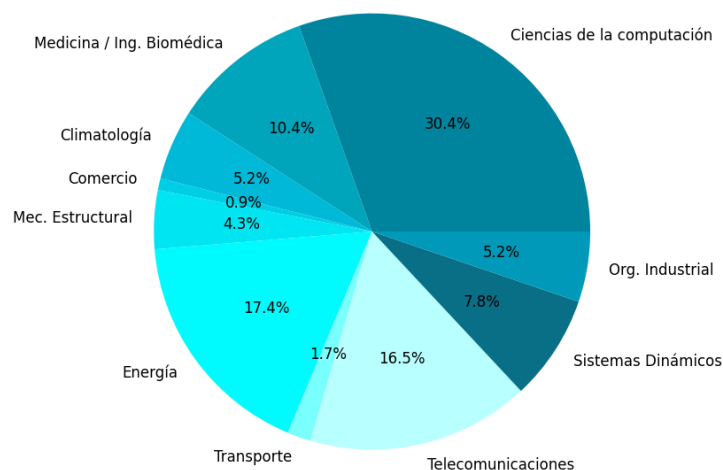


Figura 2.3. Diagrama circular de los campos de investigación que se usan el algoritmo CRO.

Los valores exactos de los distintos campos quedan recogidos en la Tabla 2.6. Los tres sectores que más relevancia cobran son el de Ciencias de la computación (32), las Telecomunicaciones (19) y la Energía (20), siendo este último en el que se encuadraría este TFM. Hay que destacar

también el sector Biomédico, del cual se han encontrado 12 publicaciones, sobre todo orientadas al tratamiento de imágenes médicas como radiografías (Gómez et al., 2020) y análisis de *datasets* compuestos de parámetros biomédicos (Yan et al., 2019), lo cual demuestra una gran versatilidad del algoritmo debido a la complejidad de la temática en comparación a las actividades observadas en el resto de los sectores.

Tabla 2.6. *Número de publicaciones relacionadas con el CRO por campo de investigación.*

CAMPO DE INVESTIGACIÓN	Nº. PUBLICACIONES
Medicina / Ing. Biomédica	12
Ciencias de la computación	35
Climatología	6
Comercio	1
Energía	20
Mecánica estructural	5
Transporte	3
Telecomunicaciones	19
Sistemas dinámicos	9
Organización industrial	6

En (Salcedo-Sanz, 2016) se realiza una revisión del estado del arte del algoritmo en lo referente a ámbitos de aplicación del CRO, en el cual introduce el funcionamiento del propio algoritmo y analiza ciertas actividades en las que se utiliza que están relacionadas con los sectores de la energía (colocación de aerogeneradores sobre el terreno y predicción de vientos), las telecomunicaciones (optimización del coste de redes móviles) o las ciencias de la computación (*clustering de datasets*).

En lo relativo al sector energético, y más específicamente en el campo de distribución de las turbinas eólicas, temática en la que se basa este trabajo, merece la pena mencionar al artículo (Salcedo-Sanz et al., 2014). En este artículo se realiza un análisis del rendimiento del CRO, aplicado al problema de diseño de parques eólicos *offshore*. Se presentan las características

principales del algoritmo y se detallan aspectos relevantes de su implementación. Además, se lleva a cabo una comparación entre el rendimiento del algoritmo CRO propuesto y diversas alternativas de metaheurísticas, como otros algoritmos evolutivos, la Evolución Diferencial (ED) y la Harmony Search (HS). Los resultados obtenidos demuestran que el algoritmo CRO supera a las alternativas mencionadas, proporcionando una solución más precisa y efectiva para la optimización en problemas relacionados con la energía eólica.

Las energías renovables dependen de diversos factores climatológicos, en los cuales también se ha aplicado el algoritmo CRO. En (Salcedo-Sanz S. et al., 2015) se detallan las principales características del algoritmo híbrido CRO-HS, el cual mezcla las metodologías del CRO con las de la HS con el fin de mejorar su funcionamiento, evaluando así su rendimiento al predecir la velocidad del viento utilizando datos de dos torres meteorológicas ubicadas en Estados Unidos y España. Los resultados obtenidos en los experimentos superan a los obtenidos por las versiones básicas de los algoritmos CRO y HS, lo que abre la posibilidad de utilizar este algoritmo híbrido en otros problemas de optimización de diversa índole.

RQ2: ¿Qué modificaciones han surgido del algoritmo base?

En la respuesta a la RQ1 se explicó ya una de las versiones del CRO, basada en la mezcla del HS con el CRO, la cual se denomina CRO-HS. Enfocándose ya en otras vertientes, se encuentra una de las modificaciones más relevantes, y en la que se ha enfocado este trabajo es la implementación de capas de sustratos al algoritmo, la cual le da unas propiedades tipo *ensemble* que permiten una mejor adaptación a varias tipologías de problemas, así como la posibilidad de romper periodos de convergencia de soluciones de forma más eficaz que con la versión original del algoritmo. Las distintas larvas que componen el coral se dividen de forma probabilística en capas, a las cuales se les aplica un operador distinto conocido como sustrato. Estos sustratos tendrán una forma de ejecutarse distinta, lo cual dará lugar a una diversidad amplia de nuevos individuos y a un mayor factor de exploración. Se puede ver una explicación más detallada del CRO-SL en la Sección 2.2.3.1.

También se ha detectado el uso del denominado algoritmo CRO con especies (Vermeij et al., 2017), el cual puede aplicarse en problemas de longitud de codificación variables. Esta modificación implica incorporar diferentes especies de corales al arrecife. El propósito de esta

modificación es que cada especie de coral represente un modelo distinto de entre una variedad de modelos posibles definidos. En este contexto, los modelos pueden abarcar desde una codificación distinta para el problema hasta un enfoque diferente para calcular la función de *fitness*. El CRO con especies constituye una metodología novedosa y eficaz para abordar problemas de optimización con codificaciones variables. En este enfoque, cada especie representa una longitud de codificación diferente, y la premisa es que solo los corales de una misma especie pueden reproducirse en el operador de desove masivo. Es relevante destacar que todos los modelos compiten en conjunto durante el proceso de establecimiento de las larvas, dado que la función de *fitness* debe ser la misma para todas las especies.

Por último, se ha detectado una versión multiobjetivo (Salcedo-Sanz et al., 2013) del propio algoritmo CRO (MO-CRO) que busca la solución óptima ante un problema dependiente de diversas variables, y que puede aplicarse sobre el algoritmo CRO básico aplicando tan solo unas pequeñas modificaciones. cuando una larva intenta ocupar una posición ya ocupada por un coral en el arrecife, se produce una lucha por el espacio. La dominancia de las soluciones determina quién gana, si la larva es dominante, ocupará el lugar del coral existente; de lo contrario, el coral existente se mantiene y la larva buscará otro lugar o morirá. Para garantizar diversidad en el arrecife, se permite un número fijo de corales con el mismo valor en todas las funciones objetivo. Después de establecer las larvas, si hay más corales con ese valor que el límite permitido, se eliminan aleatoriamente los excedentes. Este proceso recibe el nombre de Operador de Depredación Extrema (EDO).

RQ3: ¿Cómo es la tendencia de crecimiento del número de publicaciones por año relacionadas con el CRO?

Haciendo uso de los datos conseguidos tras la búsqueda y la aplicación del protocolo SLR, se ha elaborado una gráfica con el número de publicaciones por año desde el 2013 hasta junio de 2023. La tendencia mostrada en la Figura 2.4 es bastante irregular, alcanzando su máximo en el año 2018 y mostrando ciertas oscilaciones tanto en los años previos como en los posteriores, aunque los números más altos de publicaciones se dan a partir de dicha fecha.

El número de publicaciones es relativamente bajo al observado con otros tipos de algoritmos metaheurísticos (genéticos, *Chemical Reaction Optimization*, PSO, *Ant Colony Optimization*...)

o metodologías orientadas al uso de redes neuronales. Esto puede ser debido a que el algoritmo se presentó en el 2013 y tiene menos recorrido que los otros algoritmos mencionados, algunos de los cuales se presentaron el siglo pasado.

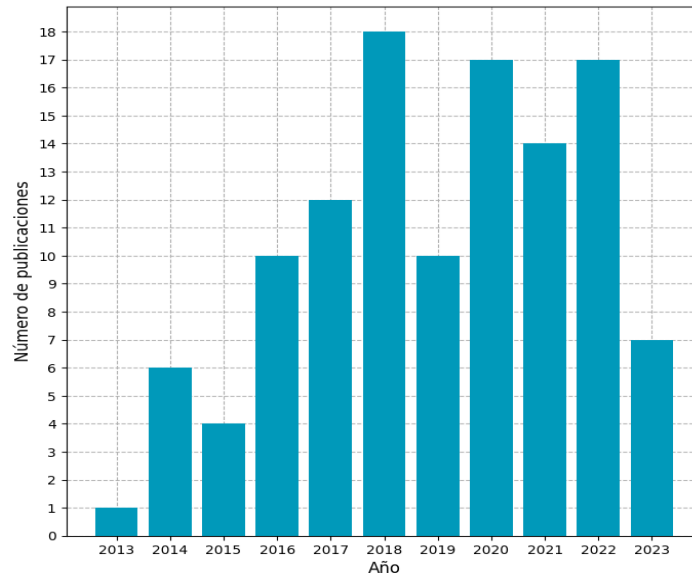


Figura 2.4. *Publicaciones sobre el CRO por desde el año 2013 hasta junio del 2023.*

Justificación RQ4: ¿Qué investigadores han elaborado artículos relacionados con el CRO y con quiénes han colaborado?

Uno de los autores más activos en el uso e investigación sobre el CRO es su propio creador, Sancho Salcedo Sanz, catedrático perteneciente al departamento de la Teoría de la Señal y la Comunicación de la Universidad de Alcalá (UAH). Entre las 116 publicaciones analizadas, este investigador está presente como uno de los autores en 50 de ellas, lo que le deja con un porcentaje de participación en las publicaciones relacionadas con el CRO cercano al 43%.

En la Tabla 2.7 se muestra el número de veces que una institución ha contribuido a las publicaciones del CRO a través de los autores vinculados a dichos organismos. La institución con más involucración en las publicaciones es la UAH (50) a la que pertenece Salcedo Sanz y que es el principal foco de investigación de este algoritmo. La Universidad de Córdoba (18), Granada (14) y la Politécnica de Madrid (15) han mostrado también cierto interés por el algoritmo y su desarrollo, a través de realizar ciertas colaboraciones con la UAH.

Tabla 2.7. *Afiliación institucional a universidades de los autores principales.*

AFILIACIÓN DE LOS AUTORES	Nº. REPETICIONES
Universidad de Alcalá	50
Universidad de Córdoba	18
Universidad Politécnica de Madrid	15
Universidad de Granada	14
Universidad de Río Grande do Norte	9
National Sun Yat Sen University	4
National Cheng Kung University	3
Universidad Complutense de Madrid	3
Universidad de la Laguna	2
Universidad Rey Juan Carlos	1
Universidad Católica de Murcia	1

En la Figura 2.5 se ha elaborado un gráfico de relación nodal entre autores de las publicaciones resultantes del proceso SLR llevado a cabo en la Sección 2.1.2. En él se pueden observar las distintas colaboraciones entre autores, entre los que destacan el ya mencionado S. Salcedo Sanz y otros investigadores como C. Camacho Gómez y J. Pérez Aracil, de cuyos nodos surgen un número relevante de arcos de conexión. También es de interés mencionar la presencia de varios autores asiáticos y latinoamericanos, demostrando la internacionalización que ha adquirido el algoritmo con el paso de los años, aumentando así su relevancia y difusión en varios países.

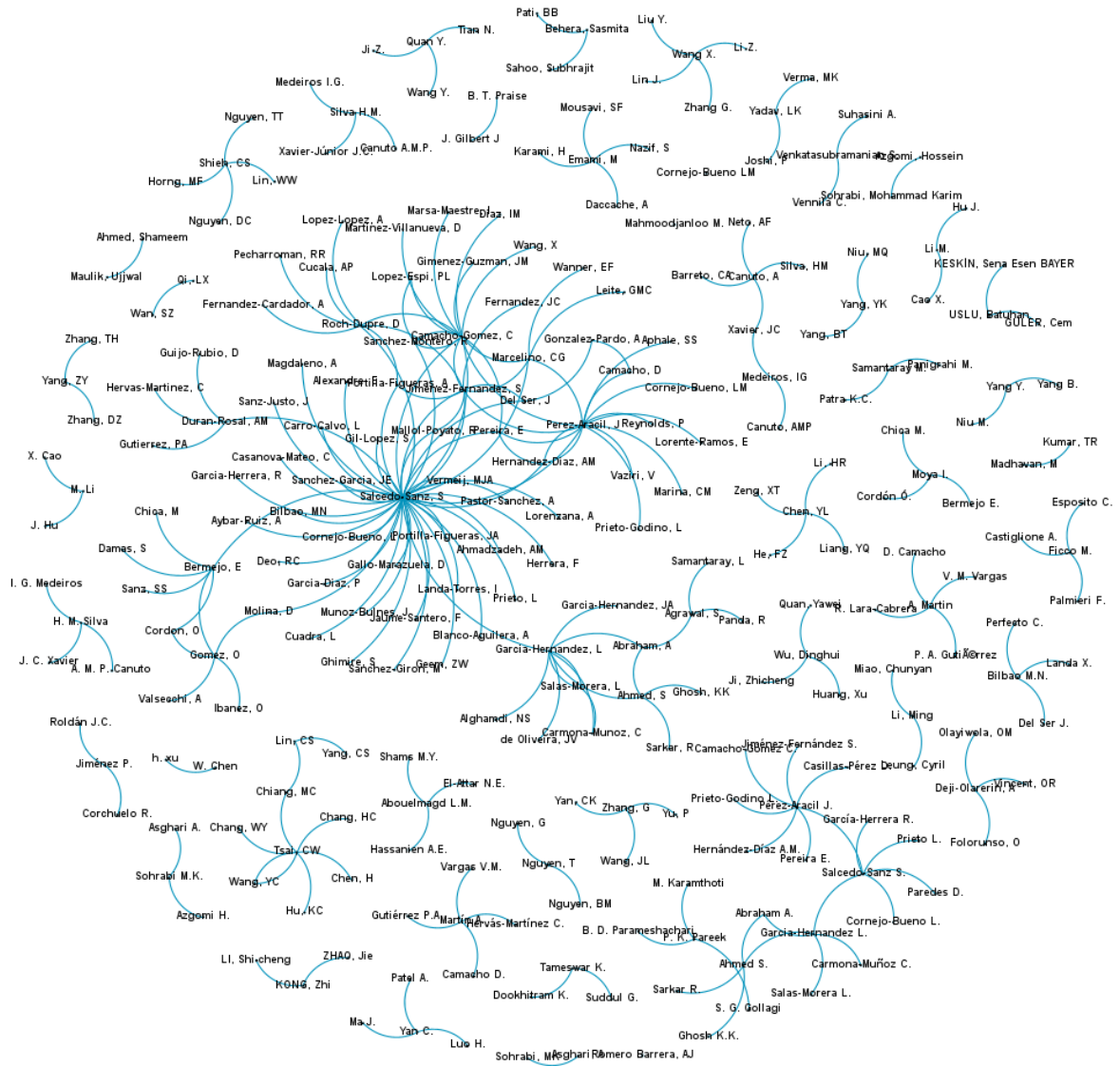


Figura 2.5. Relación nodal entre autores de las publicaciones analizadas.

2.2. Marco teórico

En esta sección, se especifica el contexto teórico y los conocimientos necesarios básicos para la comprensión del estudio realizado descrito en los capítulos posteriores. Al comienzo, en la Sección 2.2.1 se realiza una pequeña introducción de los tipos de aerogeneradores existentes y algunas de sus propiedades. Posteriormente, en la Sección 2.2.2 se detalla el funcionamiento y los conceptos clave de los modelos de estela utilizados para el cálculo de la pérdida de velocidad por la presencia de estelas generadas por los aerogeneradores, y finalmente, en la Sección 2.2.3 se describen 4 algoritmos metaheurísticos que han sido utilizados para obtener resultados o adaptados para ser incorporados como sustratos en el algoritmo CRO-SL.

2.2.1. Aerogeneradores: propiedades y funcionamiento

Los aerogeneradores son la vía más utilizada para la extracción de energía cinética del viento para su posterior conversión en electricidad. Actualmente existen dos tipos de aerogeneradores en función del eje rotacional de las palas del rotor:

- **Aerogeneradores de eje horizontal (HAWT):** son aerogeneradores que tienen su eje de rotación horizontal (paralelo a la superficie) y cuyo rotor está compuesto por palas contenidas en el plano perpendicular. Suelen ser estructuras de gran tamaño y es el tipo de turbina eólica más utilizada en los parques eólicos para la obtención de energía eléctrica debido a sus alto coeficientes de potencia (c_{pt}).
- **Aerogeneradores de eje vertical (VAWT):** cuentan con su eje de rotación en el plano vertical a la superficie y suelen tener un tamaño inferior a las HAWT. Dentro de esta clasificación de turbinas podemos distinguir dos grandes tipos, Darrieus y Savonius. Las turbinas de tipo Darrieus tienen palas curvas que aprovechan la diferencia de presión dinámica generada por el flujo de aire, siendo más eficiente a velocidades de viento bajas y en diferentes direcciones, mientras que, las turbinas eólicas de tipo Savonius tienen palas semicirculares y hacen uso de la fuerza del arrastre al moverse contra el viento, aumentando su eficiencia a velocidades altas y en una dirección constante.

Se ha elaborado la Tabla 2.8 que agrupa una serie de características con el fin de que permita al lector identificar de mejor manera las diferencias existentes entre ambos tipos de turbinas eólicas.

Tabla 2.8. Características de las turbinas eólicas de eje horizontal (HAWT) y vertical (VAWT).

CARACTERÍSTICAS	HAWT	VAWT
Longitud de pala	Elevada	Moderada
Esfuerzos principales sobre la estructura	Esfuerzos de cizalla	Esfuerzos de torsión
Rango de velocidades medias de rotación	10-35 RPM (comerciales)	50-100 RPM
Dirección incidente del viento	Los perfiles del rotor suelen ver un ángulo de resbalamiento incidente que reduce el ángulo de ataque efectivo. La góndola tiene cierto margen de giro que permite ajustar las palas del rotor a la corriente incidente.	La rotación de las palas sobre el eje de la turbina hace que se modifique el ángulo de ataque en función del ángulo de giro del rotor, provocando efectos como entradas en pérdida dinámicas por los elevados ángulos de ataque que puede llegar a adquirir el perfil aerodinámico de la pala.
Energía generada	Elevada, debido a que el rotor se encuentra a gran altura respecto a la superficie	Moderada, debido a la poca altitud a la que se encuentran
Estudio de distribución	Complejo, estelas muy turbulentas que afectan al resto de turbinas eólicas y generan un déficit en la velocidad incidente.	Sencillo, estelas débiles y con mejor distribución en sentido longitudinal a la corriente incidente que no afectan notoriamente a la velocidad del viento incidente del resto de turbinas eólicas.
Superficie barrida	Perpendicular a la corriente incidente. El ángulo de ataque de las palas del rotor se mantiene prácticamente constante con el giro del rotor.	Radial, alternando posiciones de barlovento y sotovento debido al eje de rotación. Es necesario un estudio del cambio del ángulo de ataque de la pala conforme se modifica el ángulo de giro del rotor.
Zonas de instalación	Superficies amplias y con pocos obstáculos. Pueden instalarse en el lecho marítimo (parques eólicos <i>offshore</i>).	Se suelen instalar cerca de zonas urbanas.
Mantenimiento	Complejo, debido a la gran altitud a la que se encuentra la góndola que aloja la maquinaria	Sencillo, ya que son turbinas de menor tamaño y su maquinaria es encontrada instalada a la altura del terreno.

El desarrollo de este trabajo se ha centrado en los HAWT y el estudio de cómo afecta su estela a las distintas turbinas eólicas de cara a su posterior optimización con algoritmos metaheurísticos. Los HAWT pueden tener distintas configuraciones de rotores en función del número de palas de las que dispongan, ya que, en función de esto, tendrán una serie de ventajas o desventajas a la hora de extraer energía o dar estabilidad estructural al aerogenerador. En la actualidad, la configuración más utilizada es la del modelo danés de 3 palas (Hau, 2006), ya que un número mayor de palas exigiría una reducción de la cuerda de los perfiles aerodinámicos de las palas y la necesidad de materiales más livianos y resistentes que suelen tener un coste mayor, mientras que el uso de menos palas afectaría negativamente a la estabilidad de la estructura por falta de simetría en el modelo de 1 pala o por vibraciones en el modelo de 2 palas. En la Figura 2.6 se puede observar la relación entre el coeficiente de tracción de los rotores (c_T) y el *tip speed ratio* ($TSR = \omega_r \cdot r / U_\infty$), el cual define la relación entre la velocidad lineal de la pala durante su rotación y la velocidad incidente del viento.

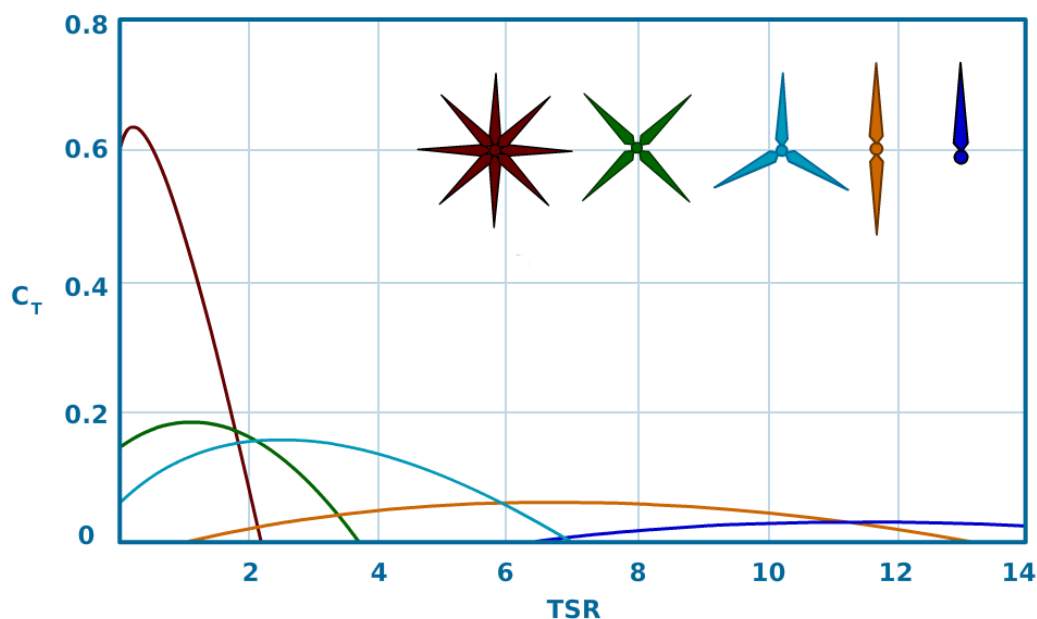


Figura 2.6. Valores del c_T para distintos diseños y valores de TSR. Adaptada de (Hau, 2006).

Las HAWT son las turbinas eólicas más utilizadas a nivel comercial debido a su capacidad de maximizar la energía cinética del propio viento para su posterior transformación en energía eléctrica. Para este hecho, este tipo de aerogeneradores dispone de una serie de maquinaria interna, cuyos componentes quedan señalizados en la Figura 2.7 y se describen en la Tabla 2.9.

Tabla 2.9. *Componentes principales de una HAWT.*

COMPONENTE	DESCRIPCIÓN
Torre	Estructura vertical que sostiene el aerogenerador y lo eleva sobre el suelo para aprovechar los vientos más fuertes y su energía cinética.
Góndola	Protección que alberga los componentes mecánicos y electrónicos del aerogenerador encargados de la transformación de energía y las mediciones del viento.
Rotor	Parte giratoria de la turbina eólica que aloja las palas y su eje de giro. Su rotación genera energía mecánica que se transmite al generador a través del eje. Está compuesto por el buje y las palas.
Palas	Elementos aerodinámicos encargados de capturar la energía cinética del viento y convertirla en energía mecánica mediante su rotación.
Buje	Componente central del rotor que conecta las palas al eje de transmisión, que a su vez hace girar el generador eléctrico.
Generador	Dispositivo electromagnético que transforma la energía mecánica en energía eléctrica. Convierte la energía mecánica generada por el movimiento de las palas del rotor en energía eléctrica mediante el principio de inducción electromagnética. Cuando el eje del rotor gira a través de un campo magnético, se genera una corriente eléctrica en el cableado del generador, lo que produce energía eléctrica.
Sistema de control	Conjunto de dispositivos electrónicos que supervisan y regulan el funcionamiento completo del aerogenerador, ajustando su posición relativa para maximizar la producción de energía y garantizar la seguridad y estabilidad del sistema.
Sistemas de medición	Elementos encargados de medir propiedades del viento, como anemómetros y veletas, para posteriormente transmitir esta información al sistema de control.
Caja de engranajes	Sistema que adapta la velocidad de rotación de las palas al generador, asegurando un funcionamiento eficiente y seguro de la turbina eólica. Esto se logra mediante la combinación interna de engranajes que aumentan la velocidad de rotación de las palas y las ajustan a la velocidad de rotación establecida del generador, protegiendo al sistema de sobrecargas.

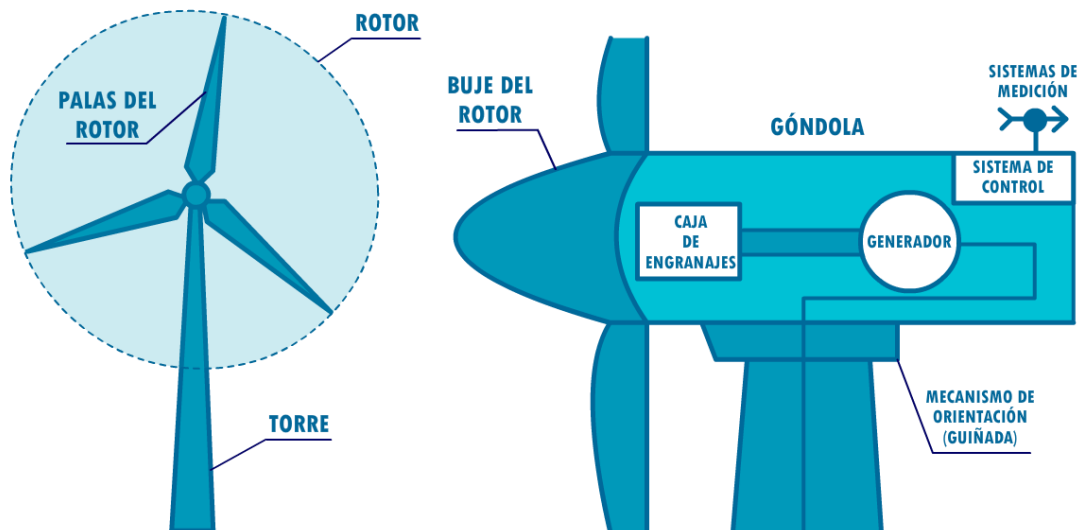


Figura 2.7. Esquema de los componentes de una turbina eólica de eje horizontal.

2.2.2. Modelos de estela

Los modelos de estela son aproximaciones numéricas basadas en modelos de ajuste experimentales, que permiten predecir de manera rápida y eficaz, el déficit de velocidad que percibe una turbina que se ve afectada por el cono de estela generado por otra turbina eólica cercana. Cuando la corriente sin perturbar aguas arriba incide sobre el rotor de una turbina eólica, el giro provocado por sus palas genera una corriente turbulenta, contenida en una geometría prácticamente cónica, que provoca que los módulos de las velocidades internas del volumen descrito se vean menguados, y, por tanto, que se produzca una deceleración del flujo de aire. En el caso de este TFM se ha decidido realizar las distintas pruebas de los algoritmos metaheurísticos con 3 modelos de estela distintos (Jensen, Larsen y Ainslie), a fin de comparar la influencia de éstos en las distintas soluciones.

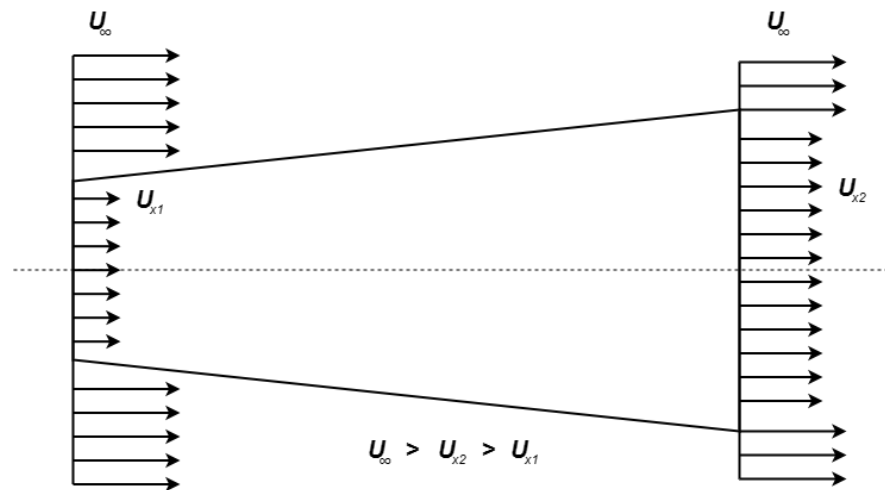


Figura 2.8. Esquema de afectación de la velocidad dentro del cono de estela.

En la Figura 2.8 se ha realizado una representación esquemática del fenómeno del cono de estela y la reducción de velocidad que sufre el viento al atravesar las palas de una turbina eólica de eje horizontal. Conforme se avanza en la dirección longitudinal del cono, el déficit de velocidad generado se minimiza y la corriente va recuperando su velocidad inicial. Existen otros modelos de carácter gaussiano que tienen en cuenta la distancia radial para definir la pérdida de velocidad, lo que hace que conforme te alejas a lo largo del eje radial, acercándote a la periferia del cono, el déficit producido tiende a disminuir de forma similar a como sucedía con el desplazamiento a lo largo del eje longitudinal.

2.2.2.1. Modelo de estela de Jensen

El modelo de estela ideado por N.O. Jensen fue expuesto en (Jensen, 1983), siendo uno de los modelos pioneros en la simulación del flujo de estelas para predecir la disminución de la velocidad incidente debido a las corrientes turbulentas generadas por el paso del aire a través de las turbinas eólicas de eje horizontal, y uno de los más populares y usados hoy en día. Jensen definió el perfil del déficit de velocidad producido en el interior del cono de la estela de forma constante para una distancia axial dada, de forma similar a una función escalón activada en la periferia de la estela, al que se denominó perfil *top hat*. El perfil descrito por la velocidad deficitaria interior del volumen de control cónico va adecuándose nuevamente al módulo de la corriente sin perturbar, exterior al cono de estela, conforme la corriente se aleja axialmente del rotor que produjo las turbulencias y la alteración del flujo de aire.

La Ecuación (1) es la definida por Jensen para llevar a cabo la predicción del decremento de velocidad sufrido por la corriente. Ésta depende de una serie de parámetros geométricos del aerogenerador como el coeficiente de tracción (c_T) o el diámetro del rotor (D_r), y otros externos como la pendiente del cono de la estela (k) o la distancia axial hasta el punto de interés donde se busca hallar la velocidad (x).

$$U_x(x) = U_\infty \cdot \left(1 - \frac{1 - \sqrt{1 - c_T}}{\left(1 + \frac{2 k \cdot x}{D_r} \right)^2} \right) \quad (1)$$

En caso de querer adaptar dicha ecuación a la actuación de varias estelas en un mismo punto para hallar el déficit producido por el solapamiento y superposición de éstas, es necesario acudir a la Ecuación (2), donde se aplica una superposición cuadrática de tipo *Axial Velocity Deficit Rankine Source Surface* (AVDRSS), cuya metodología de aplicación se puede consultar en (Zhang et al., 2022).

$$U_x(x) = U_\infty \cdot \left(1 - \frac{1 - \sqrt{1 - c_T}}{\sqrt{\sum \left(1 + \frac{2 k \cdot x}{D_r} \right)^4}} \right) \quad (2)$$

Para hacer uso de estas dos ecuaciones, es necesario hallar la pendiente del cono de estela, la cual viene dada por la Ecuación (3), que hace uso de dos parámetros para el cálculo, la altura de la turbina (z) y la longitud de rugosidad superficial del terreno (z_0).

$$k = \frac{1}{2 \ln \left(\frac{z}{z_0} \right)} \quad (3)$$

En la Tabla 7.2 del Anexo IV se pueden ver algunos valores de referencia para el valor de z_0 en función de la tipología de terreno en la que se encuentre focalizado el estudio. También, se puede hacer uso de la expresión descrita por Lettau (Gaona et al., 2015), la cual describe la longitud de rugosidad del terreno en función de una serie de características del terreno y los

elementos que lo conforman, tal y como se muestra en la Ecuación (4), donde se tiene en cuenta la altura del elemento rugoso (z_e), la superficie de la sección vertical del elemento (S) y el área horizontal media correspondiente a los distintos elementos (A_h).

$$z_0 = \frac{1}{2} \cdot \frac{z_e \cdot S}{A_h} \quad (4)$$

2.2.2.2. Modelo de estela de Larsen

Este modelo de estela fue presentado por G.C. Larsen en (Larsen, 1988), para posteriormente, en 1999 ser recomendado como modelo de cálculo para el déficit de velocidad en aerogeneradores en el informe (Pierik, et al., 1999) elaborado por la *Netherlands Energy Research Foundation* (NERF). El modelo de estela descrito por Larsen es uno de los más complejos debido a su gran cantidad de parámetros, y, además, está adaptado para realizar un cálculo aproximado en el entorno tridimensional, al hacer uso de las coordenadas longitudinales y radiales del cono de estela para el cálculo del déficit de la velocidad incidente en unas coordenadas determinadas internas al volumen de control analizado.

Este modelo describe la capa límite a través de las ecuaciones de Prandtl y utiliza la forma simplificada de las ecuaciones de Navier-Stokes realizadas por Euler para describir los fenómenos de turbulencia del flujo comprendido dentro de los límites del cono de estela. Entre las hipótesis declaradas para definir el correcto funcionamiento del modelo encontramos:

- Flujo de aire turbulento del cono de estela incompresible ($Mach < 0.3$) y estacionario.
- Simetría de estela turbulenta en el eje axial.
- Viscosidad despreciable.

En la Ecuación (5) se recoge la expresión desarrollada por Larsen para realizar el cálculo de la pérdida de velocidad sufrida por una turbina eólica que se ve afectada por las estelas de otras.

$$U_x(x, r) = U_\infty \cdot \left(1 - \frac{(c_T \cdot A_r \cdot x^{-2})^{\frac{1}{3}}}{9} \cdot \left(r^{\frac{3}{2}} \cdot (3 c_1^2 \cdot c_T \cdot A_r \cdot x)^{-\frac{1}{2}} - \left(\frac{35}{2\pi} \right)^{\frac{3}{10}} \cdot (3 c_1^2)^{-\frac{1}{5}} \right)^2 \right) \quad (5)$$

A continuación, se exponen las definiciones ordenadas de los parámetros necesarios para poder hacer uso de la Ecuación (5):

$$c_1 = \left(\frac{k \cdot D_r}{2} \right)^{\frac{5}{2}} \cdot \left(\frac{105}{2\pi} \right)^{-\frac{1}{2}} \cdot (c_T \cdot A_r \cdot x_0)^{-\frac{5}{6}} \quad (6)$$

Para obtener el valor de c_1 , es necesario desglosar dos parámetros adicionales debido a su dependencia directa con las variables x_0 y k . Para obtener el valor de k , representado en la Ecuación (7), se requiere utilizar otro parámetro adicional, m , que va ligado al coeficiente de tracción del rotor (c_T), cuya definición se recoge en la Ecuación (8).

$$k = \sqrt{\frac{m + 1}{2}} \quad (7)$$

Tal y como se puede observar en la Ecuación (8), el valor de m es directamente proporcional al valor del coeficiente de tracción (c_T).

$$m = \frac{1}{\sqrt{1 - c_T}} \quad (8)$$

El parámetro x_0 hace uso de ciertos valores geométricos de la propia turbina eólica como el diámetro del rotor (D_r) y otras variables como k y $R_{9,6}$ que muestran relación directa con el coeficiente de tracción de las palas del rotor (c_T). En la Ecuación (9) se muestra la expresión que define el valor de x_0 en función de los parámetros ya mencionados.

$$x_0 = \frac{9.6 D_r}{\left(\frac{2 R_{9,6}}{k \cdot D_r} \right)^3 - 1} \quad (9)$$

El valor de $R_{9,6}$ representa el radio del cono de estela a una distancia de 9.6 veces el diámetro del rotor de la turbina eólica, y es dependiente del coeficiente de tracción del rotor (c_T), la intensidad de turbulencia ambiental adicional (I_a) y una serie de coeficientes obtenidos a raíz de varias pruebas experimentales, tal y como se puede ver en la Ecuación (10).

$$R_{9,6} = a_1 \cdot \exp (a_2 \cdot c_T^2 + a_3 \cdot c_T + a_4) \cdot (b_1 \cdot I_a) \cdot D_r \quad (10)$$

En la Tabla 2.10 se recogen los distintos valores asignados a los 5 coeficientes empíricos que aparecen en la Ecuación (10), utilizados para el cálculo del radio de la estela situada a una distancia de la turbina eólica que la genera, equivalente a 9.6 veces el diámetro de su rotor ($R_{9,6}$).

Tabla 2.10. *Valores de los coeficientes de ajuste del modelo de estela de Larsen.*

COEFICIENTE	VALOR
a_1	0.435449861
a_2	0.797853685
a_3	- 0.124807893
a_4	0.136821858
b_1	15.62980000

2.2.2.3. Modelo de estela de Ainslie

Este modelo de estela fue presentado por J. F. Ainslie en el año 1988, tras la publicación en 1985 y 1986 de dos estudios previos, (Ainslie J. F., 1985) y (Ainslie J. F., 1986), los cuales ayudaron al desarrollo y definición de las ecuaciones que rigen el modelo descrito en (Ainslie J. F., 1988). Este modelo fue concebido a través de la comparativa de las ecuaciones de *Reynolds average Navier-Stokes* (RANS) con datos obtenidos experimentalmente de un túnel de viento, aunque debido a la complejidad de resolución de este tipo de ecuaciones del flujo, se tomaron una serie de hipótesis simplificativas que se detallan a continuación:

- Flujo de aire en el cono de estela turbulento, estacionario y axisimétrico.
- Se desprecian las velocidades circunferenciales de la estela y los gradientes de presiones del flujo de la periferia externa de ésta.

El déficit asumido por el modelo de Ainslie tiene carácter Gaussiano, lo que provoca que la velocidad incidente final sobre una turbina eólica afectada por una estela quede definida por la siguiente expresión de la Ecuación (11).

$$U_x(r) = U_\infty - U_{def} = U_\infty \cdot \left(1 - \hat{U}_M \cdot e^{-3.56 \left(\frac{r}{b \cdot D_r} \right)^2} \right) \quad (11)$$

Para poder hacer uso de la Ecuación (11) que describe la pérdida de velocidad conforme aumenta la distancia radial interna del cono de la estela ($r = \sqrt{y^2 + z^2}$, en coordenadas locales), es necesario obtener una serie de parámetros previos que son usados para realizar las correcciones correspondientes conforme se aleja la estela del rotor que genera la turbulencia.

El parámetro 'b' utilizado hace referencia a la anchura adimensional de la estela a una distancia adimensional de 2 veces el diámetro del rotor (D_r).

$$b = \sqrt{\frac{3.56 c_T}{8 D_M \cdot (1 - 0.5 D_M)}} \quad (12)$$

En torno a 5 metros de la estela, el perfil de velocidades interno del modelo es altamente gaussiano, aunque conforme se aleja del rotor generador de la misma, este perfil de velocidades tan acentuado va menguando y adecuándose a la corriente no perturbada de la periferia de la estela. Para hallar el parámetro \hat{U}_M , es necesario aplicar un factor de corrección al módulo de la velocidad deficitaria en el centro de la estela a 2 diámetros de distancia, \hat{U}_m , el cual irá disminuyendo conforme avance la estela en el eje longitudinal del cono de estela.

$$\hat{U}_M(x) = \hat{U}_m \cdot \left(1 + 7.12 \left(\sigma_\theta \cdot \frac{x}{b \cdot D_r} \right)^2 \right) \quad (13)$$

En la Tabla 7.3 del Anexo V se pueden consultar una serie de valores de referencia de la desviación típica de la dirección del viento (σ_θ) en función de la magnitud que adquiera la longitud de rugosidad superficial del terreno (z_0).

El módulo adimensional del déficit de velocidad en la línea longitudinal central del cono de estela, \hat{U}_m , marca la cota máxima de déficit producida por la estela a lo largo de todo su recorrido y viene formulada por la siguiente expresión:

$$\hat{U}_m = c_T - 0.05 - (16 c_T - 0.5) \cdot \frac{I_0}{1000} \quad (14)$$

Donde I_0 corresponde al porcentaje de intensidad de turbulencia ambiental existente en el aire que baña la zona y c_T es el parámetro geométrico que hace referencia al coeficiente tracción del rotor (c_T).

2.2.3. Algoritmos metaheurísticos

Los algoritmos metaheurísticos se basan en procesos observados en la naturaleza para llevar a cabo sus distintos procedimientos a la hora de optimizar las soluciones del problema con las que tratan, obteniendo así soluciones óptimas cercanas a la ideal. Existen dos tipos de algoritmos metaheurísticos, los basados en trayectorias que optimizan una única solución y los basados en poblaciones que trabajan sobre un conjunto amplio de soluciones que van modificando y mejorando a lo largo de las distintas iteraciones. En la Figura 2.9 se pueden observar las distintas clasificaciones de los algoritmos metaheurísticos en función de la temática en la que se inspiren a la hora de establecer las pautas seguidas durante el proceso de optimización, así como las restricciones que implementen para controlar los procesos de sustitución y generación de nuevas soluciones.

Este tipo de algoritmos, al contrario que los algoritmos basados en trayectorias, trabajan sobre un conjunto de múltiples soluciones al problema que se busque resolver, las cuales quedan recogidas en un grupo denominado población. La aptitud de cada una de las soluciones contenidas en la población viene dada por la función de *fitness* específica para el problema a tratar, que en función de si el algoritmo actúa sobre un problema de maximización o

minimización, la población tenderá a mantener a las soluciones con mayores o menores valores de *fitness* respectivamente.

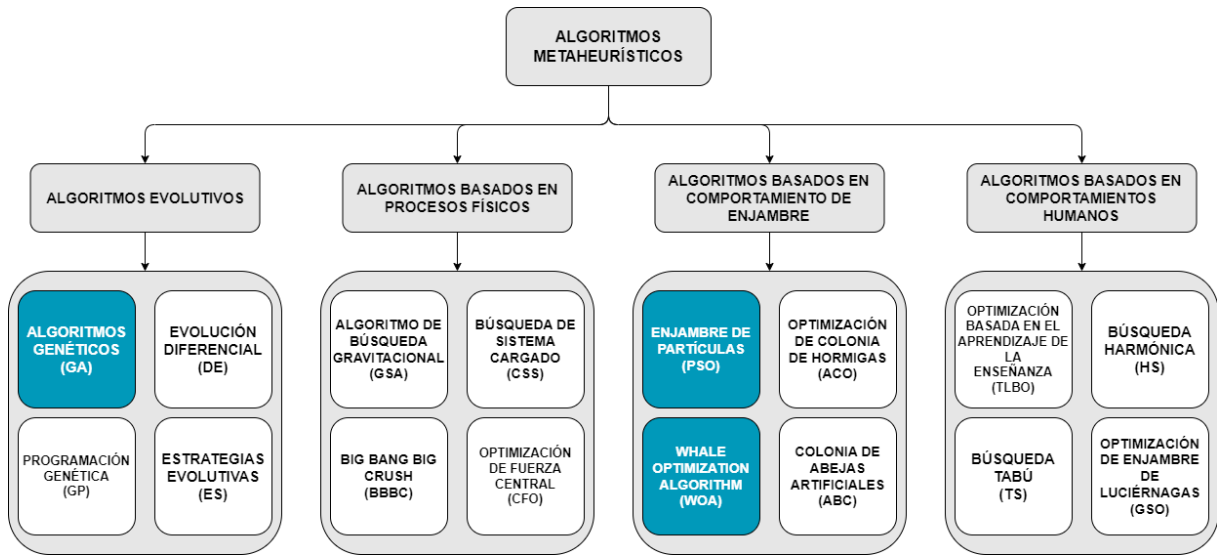


Figura 2.9. Clasificación de los algoritmos metaheurísticos. Adaptado de (Mirjalili et al., 2016).

Debido a su proceso iterativo, es necesario optimizar la función de *fitness* para que se ejecute de forma rápida y precisa, ya que esto influirá de forma notoria en la velocidad de ejecución del algoritmo metaheurístico utilizado. Es necesario realizar el cálculo del *fitness* de los nuevos individuos generados tras cada iteración, con el fin de determinar si se incorporan en la población sustituyendo a otro individuo o, si, por el contrario, son descartados y eliminados.

2.2.3.1. Algoritmo *Coral Reef Optimization with Substrates Layer* (CRO-SL)

Entre los algoritmos bio-inspirados basados en comportamientos coloniales, tenemos el algoritmo *Coral Reef Optimization* (CRO), presentado oficialmente en (Salcedo-Sanz et al., 2014). Este algoritmo simula el comportamiento de las larvas en los arrecifes de coral, usando operadores característicos de los algoritmos evolutivos, como pueden ser distintos tipos de cruce, y combinándolos con otros propios como la fragmentación o la depredación característicos de dicha especie.

Este algoritmo se clasifica dentro de las metaheurísticas evolutivas, en las cuales, los individuos (en este caso denominado larvas) pertenecientes a una población generan descendencia que tiende a sustituirles a lo largo de las sucesivas generaciones. En la Figura 2.10 se ha representado un diagrama del funcionamiento del algoritmo CRO-SL y los distintos

procesos a los que se ven sujetos las larvas de la población a lo largo de las sucesivas generaciones. Los procesos establecidos dentro del algoritmo son los siguientes:

- **Reproducción sexual externa (*broadcast spawning*):** haciendo uso de las larvas asentadas en el coral, se generan nuevas larvas a través de una serie de operadores seleccionados que modifiquen los genes de los progenitores dando lugar a nuevas larvas descendientes que intentarán establecerse en el coral.
- **Reproducción sexual interna (*brooding*):** se modifican los genes de un individuo a través de un operador de mutación seleccionado dando lugar a una larva con genes nuevos.
- **Asentamiento de larvas:** proceso por el cual las nuevas larvas generadas de los operadores reproductivos intentan establecerse dentro del coral, ya sea por huecos libres o por sustitución de otras larvas previamente asentadas. El número de intentos de asentamiento (saltos) está limitado y si no se consigue establecer en dicho periodo, la larva desaparece.
- **Reproducción asexual (*budding*):** tras producirse el asentamiento de nuevas larvas, se selecciona un porcentaje del coral para sufrir mutaciones que permitan romper periodos de convergencia.
- **Fase de depredación:** tras cada iteración, un número determinado de larvas establecidas en el coral tienen una probabilidad asociada a su defunción según su valor de *fitness*, consiguiendo que se generen nuevos huecos en el coral eliminando a las peores larvas y favoreciendo el asentamiento de descendientes futuros.

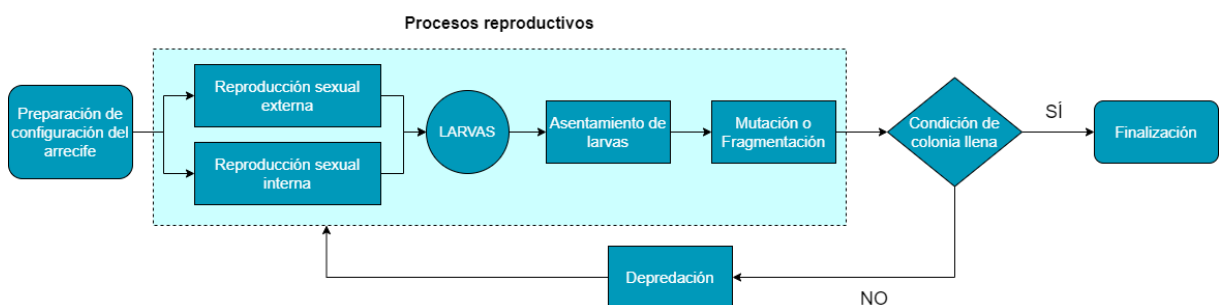


Figura 2.10. Diagrama de funcionamiento de algoritmo CRO.

Con el paso de los años, como se ha investigado en el Capítulo 2, este algoritmo ha sufrido distintas modificaciones que permiten su adaptación y la mejora de su funcionamiento ante problemas de distinta índole. Una de las vertientes generadas a raíz de este algoritmo es su

versión de métodos *ensemble*, que permiten que la reproducción sexual de las larvas se produzca por operadores distintos en función del sustrato al que pertenezca cada una de las larvas originales del coral. Esta alteración del algoritmo original recibe el nombre de *Coral Reef Optimization with Substrate Layers* (CRO-SL) y en la Figura 2.11 se realiza una representación del reparto de las larvas pertenecientes a la población del coral al principio de cada generación, entre los distintos sustratos que se hayan establecido para los procesos de generación de nuevas larvas que intenten sustituir a sus predecesoras.

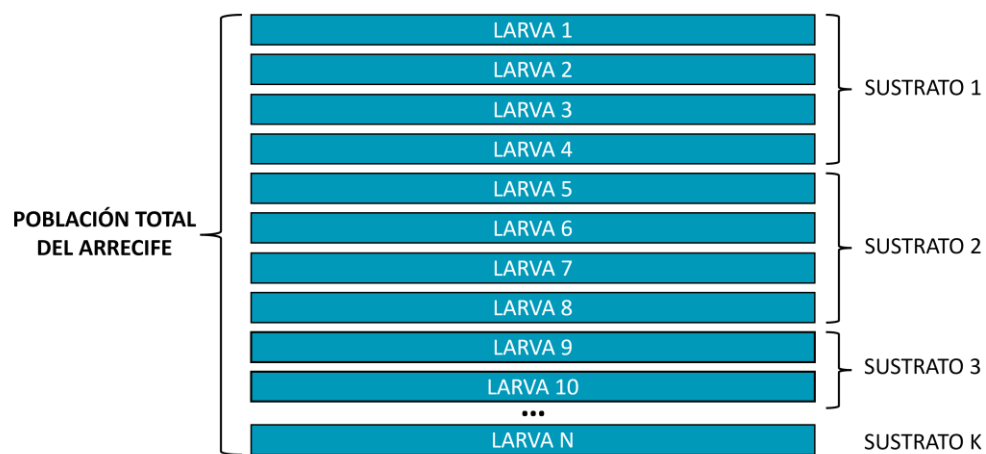


Figura 2.11. Esquema de la división de las larvas en capas de sustratos en algoritmo CRO-SL.

En este caso, los sustratos pasan a estar dentro del proceso de reproducción sexual y son repartidos entre el conjunto de larvas de la población en función de las probabilidades asociadas a cada uno de ellos. Cada uno de los sustratos irá adquiriendo mayor importancia y participación en la generación de nuevas larvas a lo largo de las iteraciones, permitiendo detectar qué operador muestra mejor funcionamiento a la hora de resolver la problemática planteada y cuáles tienen una labor más residual.

2.2.3.2. Algoritmo *Particle Swarm Optimization* (PSO)

El algoritmo *Particle Swarm Optimization* (PSO) fue presentado en el año 1995 por J. Kennedy y R. Eberhart en (Kennedy & Eberhart, 1995). Este algoritmo está inspirado en el comportamiento de enjambre de ciertas especies animales, como bancos de peces o bandadas de pájaros, imitando así sus patrones de búsqueda orientados al seguimiento del

líder del colectivo para alcanzar una finalidad determinada. Además de la posición de la partícula líder, el resto de las partículas también tienen en cuenta otros parámetros como las experiencias personales previas o la inercia de su propio movimiento.

Entre todos los hiperparámetros del PSO, hay una serie de ellos destinados a controlar el desplazamiento de las distintas partículas, los cuales se detallan a continuación:

- **Coeficiente cognitivo / aceleración personal (c_1):** controla la influencia de la mejor posición personal de cada partícula en su movimiento.
- **Coeficiente social / aceleración social (c_2):** controla la influencia de la mejor posición global encontrada por el enjambre de partículas en su conjunto a lo largo de las distintas iteraciones.
- **Factor de inercia (ω):** determina la relevancia de la velocidad actual de cada partícula en su movimiento. Un valor alto de ω ($\omega > 1$) hará que las partículas tengan un movimiento más exploratorio, acelerando y buscando nuevas soluciones sin converger en un óptimo temporal. Por otro lado, un valor bajo de ω ($\omega < 1$) generará un movimiento más explotativo, desacelerando las partículas y tendiendo a estabilizarse. Los valores típicos para ω suelen estar en el rango de 0.8 a 1.2.

Haciendo uso de la Ecuación (15) se puede obtener la velocidad de desplazamiento de la partícula (\vec{v}_{i+1}) en función de la velocidad en el instante anterior (v_i), los pesos asignados a sus referencias personales y grupales (c_1 y c_2) y una serie de valores aleatorios (r_1 y r_2) que otorgan un factor exploratorio al vector velocidad.

$$\vec{v}_{i+1} = \omega \cdot \vec{v}_i + c_1 \cdot r_1 \cdot (\vec{x}_p^{Best} - \vec{x}_i) + c_2 \cdot r_2 \cdot (\vec{x}_g^{Best} - \vec{x}_i) \quad (15)$$

En la Ecuación (16) se actualiza la posición de la partícula (\vec{x}_{i+1}) sumando a su posición en el instante previo (\vec{x}_i) la velocidad de la partícula en el instante actual calculada previamente (\vec{v}_{i+1}).

$$\vec{x}_{i+1} = \vec{x}_i + \vec{v}_{i+1} \quad (16)$$

Las partículas se encuentran en constante movimiento, por lo que el valor de *fitness* medio poblacional tenderá a describir un carácter oscilatorio, aunque el *fitness* máximo de la

población se mantendrá constante ya que la partícula que ostenta dicho valor permanecerá estática hasta que otra partícula supere su valor de *fitness*.

2.2.3.3. Algoritmo Genético Básico (BGA)

El algoritmo genético es una de las metodologías metaheurísticas basadas en poblaciones más utilizadas y conocidas, debido a su buena capacidad a la hora de adaptarse a los distintos problemas y la facilidad de comprensión de su funcionamiento en relación con otros algoritmos más complejos. La idea de este algoritmo fue presentada por J. H. Holland en los años 70, donde definió un modelo de optimización basado en la Teoría Evolutiva de Charles Darwin, descrita en (Darwin, 1859), asemejando la búsqueda de soluciones a un problema de selección natural, en los que los nuevos individuos, generados por procesos de recombinación de genes de sus progenitores, intentan sustituir a los individuos asentados previamente en la población.

En la Figura 2.12 se representa el funcionamiento del BGA y las distintas fases que pasan los individuos de la población según avanzan las generaciones, con el fin de mejorar los valores de *fitness* de la población y descartar a los peores individuos, prevaleciendo aquellos que se muestren más aptos para el problema planteado.

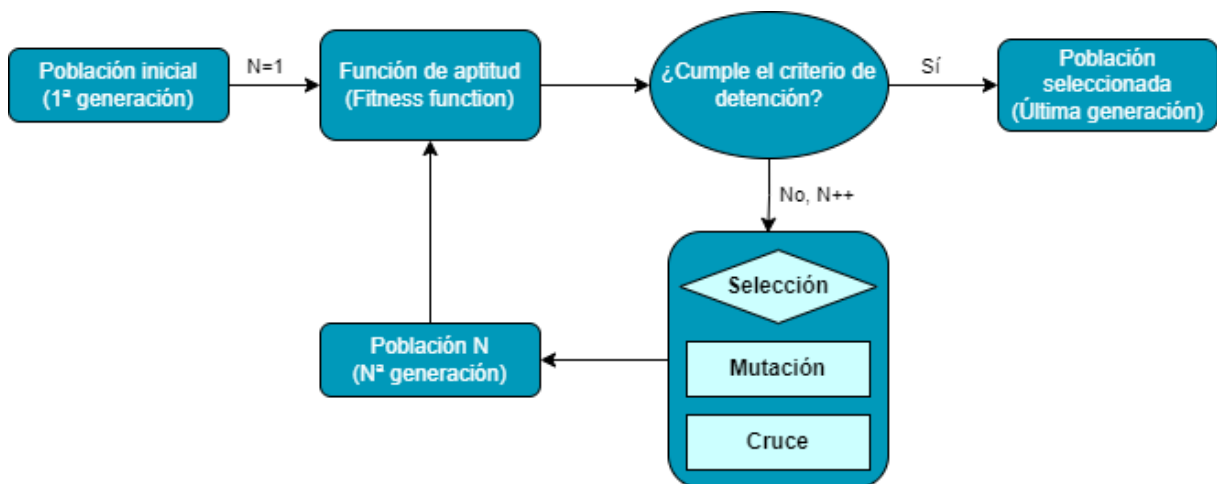


Figura 2.12. Diagrama de funcionamiento de BGA.

El BGA cuenta con tres operadores principales en su estructura, encargados de hacer evolucionar la población de una generación a otra. Los operadores de la estructura básica del algoritmo genético son:

- **Cruce/Recombinación:** encargado de seleccionar una serie de progenitores, de entre toda la población actual de individuos, y combinarlos entre sí para dar lugar a varios sucesores que cuentan con una mezcla de genes de ellos. Hay distintos métodos utilizados para realizar el cruce de individuos, entre ellos, los más conocidos son el 1Px (cruce de 1 punto), 2Px (cruce de 2 puntos) y MPx (cruce multipunto), que especifican la cantidad de cortes en la cadena de genes de los progenitores para la recombinación.
- **Mutación:** proceso que altera la cadena de genes de una serie de individuos aleatorios con el fin de explorar una mayor cantidad de soluciones al problema planteado.
- **Selección:** una vez ejecutados los operadores de cruce y mutación, y obtenidos los nuevos individuos, este operador se encarga de escoger de entre todos los individuos existentes, aquellos que se mantienen dentro de la población en la siguiente generación y cuáles son descartados en función del criterio establecido (habitualmente aquellos que cuentan con un mejor valor de *fitness*).

Cada uno de estos operadores puede sufrir variaciones y ser ajustados al problema que se busque resolver, dando lugar a una gran variedad de algoritmos genéticos alternativos, diferenciados por los tipos de operadores escogidos.

2.2.3.4. Algoritmo *Whale Optimization* (WOA)

La metodología que propone el algoritmo bioinspirado *Whale Optimization* se basa en el comportamiento social de las ballenas jorobadas y sus métodos de comunicación submarinos. Fue propuesto por S. Mirjalili y A. Lewis en el año 2016, en su publicación (Mirjalili et al., 2016).

El funcionamiento de este algoritmo es muy similar al del PSO descrito en la Sección 2.2.3.2, donde una partícula (en este caso una ballena) tiende a desplazarse en función de una posición de referencia, habitualmente dada por el líder de la población. El WOA cuenta con una serie de hiperparámetros propios de este algoritmo que sirven para actualizar la posición de las ballenas, los cuales se han recogido en la Tabla 2.11.

Tabla 2.11. *Hiperparámetros característicos del WOA.*

HIPERPARÁMETRO	DESCRIPCIÓN	EXPRESIÓN
Amplitud (\vec{A})	Determina el rango de variación permitido en la actualización de las posiciones de las ballenas en cada una de las iteraciones.	$2 \vec{a} \cdot \vec{r} - \vec{a}$
Control de amplitud (\vec{a})	Se encarga de controlar el carácter exploratorio de la solución debido a un descenso lineal de su valor que afecta de forma directa a la amplitud de la llamada de las ballenas al movimiento.	$2 \rightarrow 0$
Contracción (\vec{C})	Variable que cambia el peso de la posición de referencia, dada por el mejor de los individuos, a la hora de ejecutar un desplazamiento.	$2 \vec{r}$
Distancia contraída (\vec{D})	Distancia ligeramente modificada, en función del valor de \vec{C} , entre el mejor individuo de la población y aquel que va a modificar su posición.	$ \vec{C} \cdot \vec{x}_{ref} - \vec{x}_i $
Distancia (\vec{D}')	Distancia real entre el mejor individuo de la población y aquel que va a modificar su posición.	$ \vec{x}_{ref} - \vec{x}_i $

En la fase de inicialización, se crea una población inicial con una serie de soluciones candidatas asociadas (ballenas). Posteriormente, en la fase de evaluación, se analiza el rendimiento de cada ballena según la función de *fitness* establecida que estima la calidad de cada una de las ballenas de la población. La mejor ballena se identifica y su posición se almacena a modo de punto de referencia como la mejor solución histórica de la población hasta el momento, para que, en la fase de actualización de posición y velocidad, todas las ballenas ajusten sus posiciones y velocidades en función de dicho punto de referencia de la mejor solución histórica. Algunas ballenas también pueden realizar una búsqueda local en su vecindario para mejorar sus soluciones y usar como punto de referencia las coordenadas de otra ballena distinta a la líder, a fin de dar a la búsqueda de nuevas soluciones una naturaleza más exploratoria. Se verifica que las posiciones de las ballenas estén dentro de los límites permitidos en cada iteración. Después de cada actualización, se evalúa nuevamente el *fitness*

de las ballenas de la población y, si se cumple algún criterio de terminación, se finaliza el algoritmo y se devuelve la mejor solución encontrada, mientras que, si no se cumple el criterio de terminación, se repiten las fases anteriores hasta que se cumpla.

El hiperparámetro de control de amplitud (\vec{A}) se actualiza de forma descendiente y lineal para equilibrar la exploración y explotación del algoritmo en diferentes etapas tras cada iteración. Esto hace que, en iteraciones iniciales, las ballenas tiendan a realizar una búsqueda más amplia, desplazándose distancias mayores (exploración), y en iteraciones finales se tienda a realizar movimientos más lentos, acercándose de mejor manera a los puntos de referencia (explotación).

Haciendo uso de los hiperparámetros contenidos en la Tabla 2.11, se podrá actualizar el valor de la posición de la partícula en función del valor de amplitud asociado a dicho movimiento en esa iteración. Existirán dos metodologías de movimiento con un 50% de probabilidades de ejecutarse. La primera de ellas describirá un movimiento envolvente alrededor del punto de referencia y utiliza la Ecuación (17) para actualizar la posición. Dentro de esta metodología de movimiento, cambiará el punto de referencia en función del valor de su amplitud:

- Si $\vec{A} < 1$, para modificar la posición de la ballena se tomará la posición de la ballena con mejor valor de *fitness* de la población para referenciar el rango de alcance el movimiento ($\vec{x}_{ref} = \vec{x}_g^{Best}$).
- Si $\vec{A} \geq 1$, la referencia de movimiento de la ballena para ejecutar su movimiento pasa a ser la posición de una ballena aleatoria de la población ($\vec{x}_{ref} = \vec{x}_{rand}$).

$$\vec{x}_{i+1} = \vec{x}_{ref} - \vec{A} \cdot \vec{D} \quad (17)$$

El segundo mecanismo de desplazamiento que puede utilizar la ballena es el movimiento en espiral, de forma que actualizará su posición siguiendo una espiral logarítmica descrita por la Ecuación y que es dependiente de un parámetro '*l*' cuyo valor corresponde a un número aleatorio contenido en el rango [-1, 1].

$$\vec{x}_{i+1} = \vec{D}' \cdot e^{b \cdot l} \cdot \cos(2 \pi l) + x_g^{Best} \quad (18)$$

3. Objetivos y metodología de trabajo

En este capítulo se marcan los objetivos iniciales planteados. Para la definición de objetivos elaborada en las Secciones 3.1 y 3.2, se ha seguido la metodología SMART ideada por George T. Doran en (Doran & others, 1981), que establece las propiedades con las que deben contar los distintos objetivos generales y específicos definidos para la ejecución del trabajo. Según el método SMART escogido, los objetivos deben ser específicos (S), medibles (M), alcanzables (A), relevantes (R) y con un tiempo de ejecución determinado (T). De igual forma, el objetivo general del trabajo pasará a ser denominado por el acrónimo 'OG', mientras que los objetivos específicos serán denominados por 'OE(X)', siendo '(X)' el número identificativo de dicho objetivo específico a fin de diferenciarlos.

3.1. Objetivo general

El objetivo general de este TFM es la mejora del funcionamiento y ejecución del algoritmo CRO-SL, a través de la incorporación de nuevos sustratos, que hagan que se obtengan mejores soluciones que el algoritmo CRO-SL inicial que cuenta con 4 sustratos activos por defecto, con sus correspondientes probabilidades de alojar individuos. Los sustratos implementados serán diseñados y ajustados por un proceso de *tuning* de hiperparámetros seleccionados, orientado a la optimización de posicionamiento de aerogeneradores sobre terrenos planos (parques eólicos *offshore*), y también desarrollando la funcionalidad de analizar las distribuciones de turbinas con distintas altitudes (parques eólicos *onshore*).

3.2. Objetivos específicos

En la Tabla 3.1 se detallan los objetivos específicos establecidos para el desarrollo de este TFM. Se han establecido un cómputo total de 6 objetivos específicos sobre distintas vertientes relacionadas con el problema de colocación de aerogeneradores y las nuevas implementaciones que se buscan añadir.

Tabla 3.1. *Desglose de los objetivos específicos (OE).*

OBJETIVO ESPECÍFICO	DESCRIPCIÓN
OE1: estudiar el aporte de la comunidad investigadora al desarrollo del CRO en los últimos años.	Conocer el estado de las investigaciones científicas en un campo permite marcar de forma más concisa los objetivos del estudio y agilizar el proceso de trabajo. Se busca conocer los problemas en los que se ha aplicado el CRO-SL y la calidad de las soluciones obtenidas respecto a otros métodos.
OE2: ampliar la librería de modelos de estela actual.	Incorporar a la librería de <i>scripts</i> desarrollados en el estudio previo, que cuenta en la actualidad con 8 modelos, un nuevo modelo de estela no utilizado anteriormente y que muestre unas predicciones adecuadas que verifiquen su correcto funcionamiento.
OE3: añadir nuevos sustratos al algoritmo CRO-SL que mejoren su funcionamiento.	Haciendo uso del código base del algoritmo CRO-SL, añadir nuevos sustratos basados en operadores no utilizados o algoritmos metaheurísticos.
OE4: analizar la influencia de cada uno de los sustratos en las soluciones obtenidas a raíz del análisis con el CRO-SL.	A modo de evaluación, comprobar el comportamiento de los sustratos a lo largo de las generaciones y ver la importancia de cada uno de ellos a la hora de optimizar el posicionamiento de turbinas eólicas haciendo uso del algoritmo CRO-SL, midiendo así su grado de mejora.
OE5: incorporar la posibilidad de generar terrenos con relieve y adaptar los modelos de estela para analizar las distribuciones con desniveles.	Crear una función que permita generar una matriz de desniveles que pueda ser utilizada para realizar una predicción tridimensional del déficit de velocidad producido por las estelas de las turbinas eólicas.
OE6: realizar una comparativa del CRO-SL modificado con el CRO-SL inicial y el BGA.	Esta comparativa entre algoritmos permite conocer el desempeño y la aptitud de cada uno de los algoritmos a la hora de abordar la resolución del problema de distribución de aerogeneradores.

3.3. Metodología de trabajo

Para iniciar los distintos análisis, se ha llevado a cabo una selección de metodología y herramientas a utilizar. Se han ponderado las opciones más eficaces a la hora de realizar los distintos cálculos, con el fin de agilizar los procesos de iteración correspondientes a las distintas generaciones de individuos de la población.

Una de las elecciones que era necesario hacer fue ponderar entre el uso de herramientas basadas en *Computer Fluid Dynamics* (CFD) o utilizar modelos de estela experimentales. En la Tabla 3.2 se han recogido algunos parámetros de interés, que inciden notablemente en la resolución del problema, y se ha descrito las características de cada uno de ellos con respecto a cada una de las propiedades especificadas.

Tabla 3.2. *Comparativa de características entre el CFD y los modelos de estela.*

PARÁMETROS	CFD	MODELOS DE ESTELA
Precisión	Alta.	Media.
Tiempo de ejecución	Muy elevado.	Bajo (<1 seg.).
Capacidad de automatización	Baja.	Alta.
Necesidad de preprocesamiento de datos	Diseño CAD, mallado, cálculos de y^+ ...	Coordenadas de turbinas sobre un plano.
Equipamiento informático	Componentes de alta calidad.	Estándar.
Simplicidad jerárquica de programas	Necesidad de sincronizar varias herramientas.	Se puede implementar en una única herramienta.

Para realizar la comparativa de resultados, se almacenarán los valores de la energía media producida por la población y la energía máxima generada por el mejor de los individuos a lo largo de las distintas generaciones. Esto permite la representación de la tendencia evolutiva de cada uno de los algoritmos comparados, así como la velocidad de crecimiento de los valores obtenidos por la función de *fitness*. Además, el hecho de contar con varios algoritmos (dos ya implantados previamente y uno nuevo modificado), hace que se pueda ver la efectividad de cada uno para este factor de ocupación de turbinas sobre el terreno seleccionado. Al también disponer de 3 modelos de estela distintos, se puede observar cuál de ellos genera una disminución de velocidad más acentuada y la afectación que tiene sobre la distribución de aerogeneradores, haciendo el producto cartesiano entre el conjunto de modelos de estela {WM} y cada una de las soluciones optimizadas obtenidas, {S}, gracias a los algoritmos metaheurísticos utilizados.

4. Identificación de requisitos

En este capítulo se debe indicar el trabajo previo realizado para guiar el desarrollo del software. Esto debería incluir la identificación adecuada del problema a tratar, así como del contexto habitual de uso o funcionamiento de la aplicación. Idealmente, la identificación de requisitos se debería hacer contando con expertos en la materia a tratar.

Se parte del programa desarrollado en (Romero Barrera, 2023). Para realizar este estudio, será necesario modificar varios parámetros del código original de la función de *fitness*, de forma que sea posible habilitar el uso del nuevo modelo de estela codificado desde allí. También habrá que modificar la función de *fitness* para hacer uso de la diferencia de alturas entre dos puntos del mapa del terreno y hacer uso de este desfase de altitud entre turbinas para el cálculo de pérdida de velocidad por los distintos modelos de estela.

4.1. MATLAB: Sistema de cómputo numérico y entorno de programación

MATLAB (MATrix LABoratory) es un lenguaje de cómputo numérico creado en 1984 por la empresa Mathworks, y que ofrece a su vez, un entorno de programación de pago que destaca por su eficiencia y buena optimización para las distribuciones de Windows, MacOS y Linux. Es uno de los lenguajes más utilizados en el sector científico y de la ingeniería, y que es impartido en muchas universidades con el reparto de licencias educativas a los estudiantes, que permiten su uso gratuito durante el periodo de matriculación de sus estudios.

En (Romero Barrera, 2023) se elaboró una tabla comparativa entre MATLAB y el uso de Python con librerías de cálculo como NumPy para el desarrollo del programa. Se puede consultar dicha información en la Tabla 7.4 del Anexo VI, la cual se ha recuperado del estudio previamente mencionado. Partiendo de la base de haber seleccionado ya MATLAB como lenguaje de programación, se procede a analizar los distintos entornos de programación con compilador existentes.

De igual forma, existe una alternativa a este entorno de programación, la cual es de código abierto, llamada Octave. En la Tabla 4.1 se ha realizado una comparativa para la selección de la plataforma a utilizar para ejecutar los algoritmos metaheurísticos diseñados, con el fin de poder ponderar más fácilmente cuál es la mejor alternativa entre las dos opciones indicadas.

Tabla 4.1. *Comparativa de características entre MATLAB con Octave.*

CARACTERÍSTICAS	MATLAB	OCTAVE
Licencia	Privativa (licencias individuales, académicas, empresariales o de prueba).	Código abierto.
Plataformas compatibles	Windows, Linux, Ubuntu.	Windows, macOS, Linux, BSD/OS.
Interfaz de usuario	Interfaz gráfica y de línea de comandos.	Interfaz gráfica mínima y línea de comandos.
Rendimiento	Plataforma muy optimizada y ampliamente centrada en cálculos ágiles con matrices.	Plataforma más tosca y con menores velocidades de ejecución de procesos de cálculo y reparto de recursos.
Compatibilidad de ficheros de datos y scripts	'.m', '.mat'	'.m', '.mat'
Lenguaje de programación	MATLAB.	MATLAB.
Librerías y paquetes	Amplia colección de paquetes y herramientas, con soporte oficial	Menor cantidad de librerías y paquetes, muchos de ellos con comandos que difieren de los de MATLAB.

Dado que se dispone de una cuenta con licencia educativa gratuita, la cual permite la descarga de todos los paquetes oficiales de MATLAB para su uso en la plataforma, el factor económico en el cual tenía ventaja Octave, deja de tener relevancia a la hora de escoger entre ellos.

El hecho de contar con una amplia gama de paquetes oficiales y con soporte, una interfaz gráfica de usuario (GUI) más amigable y personalizable, y tiempos de procesado y ejecución de scripts más cortos, han hecho que finalmente sea MATLAB la plataforma escogida para el desarrollo y la ejecución del código asociado a este TFM.

Para ejecutar los distintos *scripts* desarrollados, es necesario la instalación de una serie de paquetes adicionales de MATLAB, ya que se requieren ciertas funciones pertenecientes a éstos. Estos paquetes pueden ser instalados desde la ventana de *add-ons* del entorno de

MATLAB, donde se puede descargar el paquete y consultar cierta documentación relacionada.

Entre los paquetes requeridos destacan:

- ***Statistics and Machine Learning Toolbox***: proporciona funciones y herramientas para realizar análisis estadísticos y aplicar técnicas de aprendizaje automático. Incluye funciones para el análisis exploratorio de datos, modelado estadístico, pruebas de hipótesis, regresión, clasificación, agrupamiento, entre otros. Además, ofrece algoritmos populares de aprendizaje automático como máquinas de vectores de soporte (SVM), árboles de decisión, redes neuronales, entre otros.
- ***Optimization Toolbox***: centrado en la optimización numérica, que implica encontrar los mejores valores posibles para las variables de un problema dado, proporcionando para ello un conjunto de algoritmos y herramientas para resolver problemas de optimización lineal, no lineal, convexa y no convexa.
- ***Control System Toolbox***: diseñado para el análisis y diseño de sistemas de control, ofreciendo una amplia gama de funciones y herramientas para modelar sistemas dinámicos, realizar análisis de estabilidad, diseño de controladores clásicos y modernos, sintonización de controladores, simulación y verificación de sistemas de control.
- ***Image Processing Toolbox***: conjunto de funciones y herramientas orientados al procesamiento de imágenes que permiten realizar diversas operaciones de procesamiento de imágenes como la aplicación de filtros, la segmentación, la extracción de características o el reconocimiento de objetos y más. Incluye algoritmos de visión por computadora, transformadas de imágenes, manipulación de formatos de imágenes y visualización de resultados.
- ***Curve Fitting Toolbox***: proporciona herramientas y funciones para ajustar curvas y superficies a conjuntos de datos. Esta herramienta es útil cuando se desea encontrar una función matemática que se ajuste de manera óptima a los datos experimentales o teóricos.

4.2. Datos climatológicos y curva de potencia

Para el cálculo del valor de la función de *fitness*, la cual arroja un valor de *fitness* a cada individuo, equivalente a la energía producida a lo largo de un periodo de tiempo especificado, es necesario disponer de los vectores de vientos medios por hora de la zona de interés. Con estos datos, es posible elaborar una representación gráfica de las frecuencias, direcciones y magnitudes de la velocidad del viento a través de una rosa de los vientos, al igual que realizar el cálculo de la energía producida que indicará el nivel de aptitud de cada individuo de la población.

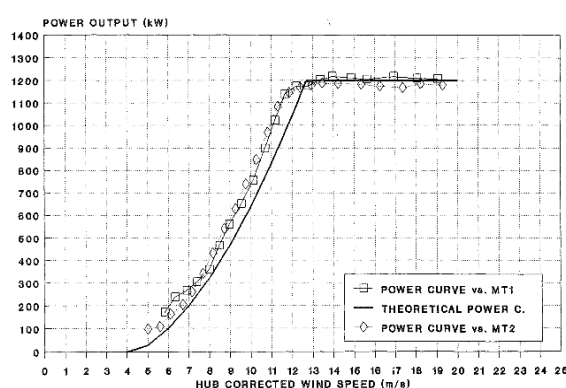
Se ha utilizado una matriz de vectores viento reales correspondientes a la provincia de Badajoz. Esta matriz cuenta con un total de 8760 vectores de 2 dimensiones, que hace referencia a los vectores de viento medios por hora a lo largo de 1 año.

Para obtener estos datos, se ha utilizado la base de datos de ERA5 que aloja los datos climáticos de distintas coordenadas terrestres desde 1940 hasta la actualidad, los cuales fueron recopilados por el Programa de Observación de La Tierra de la UE, denominado Copernicus. Gracias a ella, se pueden descargar *datasets* con diversos parámetros meteorológicos pertenecientes a unas coordenadas geográficas específicas para su posterior uso en investigaciones que lo requieran, como en este caso los valores de velocidad U (velocidad este-oeste) y V (velocidad norte-sur) a 10 m de altitud respecto a la superficie, lo cuales son usados para sacar los vectores viento utilizados.

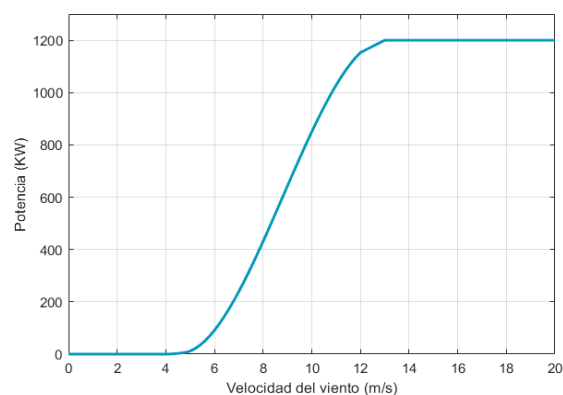
A una distancia de 10 m de altitud no existe el efecto de la capa límite que genera la superficie y, para cambios de altitud en torno a su misma magnitud, la densidad no sufre apenas modificaciones siguiendo las ecuaciones establecidas la normativa ISO 2533:1975 (ISO, 1975) de la *International Standard Atmosphere* (ISA) para altitudes menores a 11 km. Si la zona no tiene obstáculos de gran tamaño, puede considerarse el mismo módulo no perturbado para la altitud de análisis de 60 m utilizada en los experimentos.

En lo referente a la gráfica de la curva de potencia, se ha consultado el informe técnico (Avia, 1992) y se ha extraído de él una curva de potencia perteneciente a un modelo de turbina eólica AWEK-60. Utilizando la imagen presente en el documento, se procedió a transferir los datos a MATLAB mediante un ajuste mediante *splines* del gráfico original, con el propósito de utilizarlos posteriormente en la etapa final del cálculo de la función de *fitness*.

La turbina AWEC-60 seleccionada para la generación de nuestro gráfico de potencia tiene una altura al buje de 60 m y un diámetro de rotor de 60 m. Sin embargo, se decidió utilizar un diámetro de rotor de 80 m para mantener las mismas condiciones que se mencionan en la publicación (Zergane et al., 2018), uno de los artículos de referencia con el que se inició estas investigaciones en el pasado. La elección del diámetro del rotor es crucial para la delimitación de la estela turbulenta generada por las turbinas, lo cual afecta la interacción con un mayor o menor número de turbinas aguas abajo. Por lo tanto, se concluyó que sería beneficioso mantener dicho diámetro como referencia para ciertas magnitudes en las etapas de validación.



(a) Curva de potencia real de AWEC-60.



(b) Curva de potencia aproximada de AWEC-60.

Figura 4.1. Curva de potencia original y aproximada de turbina eólica AWEC-60.

En definitiva, el *dataset* de vientos tomado dispone de un total de unas dimensiones de 8760x2, siendo la primera dimensión el número de horas a lo largo de un año (8760 horas) y la segunda las dos componentes de la velocidad que se alojan por cada una de las horas. Estos datos son almacenados en un fichero '.mat' que es consultado por el *script* principal para obtener la matriz de vientos y realizar los cálculos necesarios durante el cálculo de la función de *fitness*. La curva de potencia perteneciente a una turbina eólica AWEC-60, extraída de (Avia, 1992), se ha adaptado a través de una *B-spline* que aproxima los valores de la gráfica experimental expuesta en el informe técnico de la turbina, y que permite que se puedan obtener valores continuos de la misma para cualquier velocidad incidente.

4.3. Función de *fitness* y modelos de estela de Jensen y Larsen

La función de *fitness* diseñada en (Romero Barrera, 2023) cuenta con dos partes diferenciadas. En la primera parte, se obtiene, en función del mapa de distribuciones de turbinas transmitido a modo de *input* de la función, la matriz de distancias de afectación, en las cuales se recogen las distancias entre el aerogenerador que se encuentra inmerso en la estela y el otro aerogenerador que la provoca. La matriz resultante tendrá una geometría cuadrada y las distancias a las turbinas que no se vean afectadas serán nulas. Cada una de las columnas representará al aerogenerador afectado, mientras que cada fila representa a las turbinas eólicas afectantes.

La función de aptitud de nuestro problema proporcionará la energía generada por todas las turbinas, considerando una distribución horaria de los vientos dentro de un marco temporal específico. Esta función toma como hiperparámetros la población total (donde cada individuo es un vector que representa las posiciones ocupadas por las turbinas en la matriz), el vector de vientos y la estructura de datos 'data', que almacena los parámetros del modelo de estela y las dimensiones del terreno.

En función del modelo de estela seleccionado, se generan las constantes correspondientes a dicho modelo dentro de la estructura 'data', que se pasa como *input* a la función de *fitness*, y dichos valores serán utilizados para representar los límites de la estela y calcular las velocidades incidentes en cada una de las turbinas. En la Tabla 4.2 se resume la estructura utilizada para el cálculo de *fitness* de los individuos.

La estructura del flujo de trabajo dispone de la capacidad para el cálculo del *fitness* con varios modelos de estela, sin necesidad de cambiar el *script* principal ni crear diferentes estructuras de código según el modelo. Se utiliza la función 'f_powerPlantsT_fast.m' para gestionar las dos fases del proceso, en las cuales se obtienen las distancias entre los aerogeneradores afectantes y afectados, para posteriormente, calcular la velocidad incidente sobre ellos.

Tabla 4.2. *Funciones y parámetros utilizados en la función de fitness.*

FUNCIÓN	INPUTS	OUTPUTS	DESCRIPCIÓN
f_powerPlants_T	vVec gr data	pwr_T gan_T cost_T obj_T dt	Función principal encargada de analizar los datos de la estructura 'data' y gestionar el uso del resto de funciones.
f_powerPlants_f1	vVec gr data ppPower	xUDef yUDef	Función encargada de detectar el modelo de estela y llamar a la función 'getDist' correspondiente.
f_powerPlants_f2	vVec gr data ppPower xUDef yUDef	pwr_t pwrGen Ux gan cost obj	Función que detecta el modelo de estela, llama a la función 'wm_' correspondiente para predecir el déficit de velocidad y calcula la energía generada por el parque eólico en la hora analizada.
getDistTrapz getDistSmooth	vDir gr data	xUDef yUDef	Calcula la distancia entre el aerogenerador que produce la estela y aquel que se encuentra afectado por ella.
wm_jensen wm_larsen wm_ainslie	m vMod xUDef yUDef	Ux	Calcula la velocidad incidente sobre cada una de las turbinas eólicas posicionadas en el terreno.

En primer lugar, es necesario determinar la interacción mutua entre las estelas de cada una de las turbinas eólicas para calcular el déficit de velocidad resultante. Las estelas cambiarán su dirección y geometría en función de la matriz de vientos ('vVec') utilizada y el modelo de estela seleccionado. Para determinar qué aerogeneradores se ven afectados por las estelas de otros, es necesario conocer el modelo de estela y estimar los límites del cono de estela. Mediante la función 'f_powerPlants_f1', se obtienen dos matrices de afectación que registran las distancias longitudinales ('xUDef') y transversales ('yUDef') desde la turbina eólica que genera la estela hasta la turbina que se ve afectada por ella.

En segundo lugar, utilizando las ecuaciones del modelo de estela seleccionado, invocadas desde la función 'f_powerPlants_f2', se calcula la pérdida de velocidad en cada una de las turbinas debido a las diferentes estelas que las afectan, utilizando las matrices 'xUDef' e 'yUDef' obtenidas en el paso anterior. Esta velocidad deficitaria permite calcular la velocidad final incidente en cada una de las turbinas eólicas para las diferentes horas analizadas, lo que a su vez permite estimar la energía generada por cada una de ellas en cada hora.

Finalmente, una vez que se recibe el módulo de velocidad que incide en cada una de las turbinas eólicas ubicadas en el terreno, el gestor principal ('f_powerPlantsT_fast') procede a calcular la energía generada a lo largo del período temporal establecido. Como *inputs* para el cálculo, se requiere la matriz 'gr' que contiene la distribución de turbinas en el terreno, la matriz 'vVec' que contiene los vectores de viento para cada hora, y la estructura 'data'. Como resultado, se obtiene la energía total generada 'pwr_T' y una estructura 'dt' que proporciona los datos desglosados de la energía total generada por cada hora correspondiente al vector 'vVec'.

4.4. Programa de Algoritmo Genético Básico (BGA)

Para comparar resultados de optimización se ha utilizado un BGA, previamente programado para la realización del estudio llevado a cabo en (Romero Barrera, 2023), el cual no ha sido modificado con respecto a la versión desarrollada en dicho trabajo. Los resultados obtenidos se utilizan a modo de validación y justificación de la implementación del sustrato 1Px, con mutación aleatoria asociada, dentro del grupo total de sustratos del CRO-SL. El programa previamente desarrollado hacía uso de 2 operadores, el 1Px y la mutación, la cual es la estructura más básica del propio algoritmo. En la Figura 4.2 se muestra una representación gráfica del funcionamiento del operador 1Px para la modificación de genes de los distintos individuos.

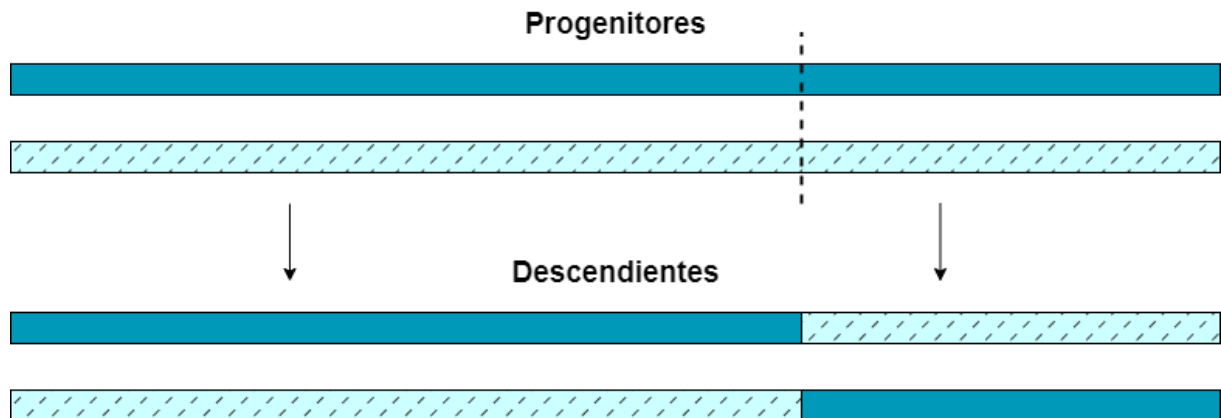


Figura 4.2. Representación de la metodología del operador 1Px.

En el caso del operador de 1Px, éste actuaba también a modo de operador de selección de los progenitores encargados de crear descendencia y posteriormente, se aplicaba una recombinación de genes aplicando un corte de 1 punto en la cadena de genes/cromosomas, que en este caso eran equivalentes por la codificación de individuos utilizada.

Posteriormente, la población de individuos generados era sometida a un proceso pseudoaleatorio de mutación, al cual, se le aplicaban dos probabilidades distintas: una correspondiente a la probabilidad de que mute uno de los nuevos individuos y otra la probabilidad de que mute cada gen del individuo.

Tras finalizar los dos procesos, se mezclan la población antigua con la nueva población de descendientes y se escogían los mejores del conjunto total según el valor obtenido de la función de *fitness*, descartando los peores resultados y manteniendo constante el tamaño de la población.

4.5. Programa de *Coral Reef Optimization with Substrates Layer* (CRO-SL)

El código base de este programa fue cedido por Jorge Pérez Aracil, profesor de la Universidad de Alcalá (UAH), para su modificación y adaptación al problema de colocación de turbinas eólicas en (Romero Barrera, 2023). Este algoritmo, contaba con una librería de 14 sustratos que podían actuar de para problemas discretos cuyos individuos cuenten con genes numéricos de tipo enteros (*int*) o reales (*float*).

Por defecto, el código del programa hacía uso de 4 sustratos distintos con sus probabilidades correspondientes a la incorporación de cada uno de los individuos a uno u otro sustrato. Entre

el grupo de sustratos seleccionado, 3 eran tipos de cruce ($BLX\alpha$, MPx y 2Px) con un 20% de probabilidad cada uno, y el último de los sustratos correspondía a la mutación gaussiana (GM) con un 40% de probabilidad asociado.

A pesar de esto, se ha de mencionar que el algoritmo CRO-SL tenía una serie de sustratos ya diseñados y que se podían incorporar al proceso de optimización tanto para operaciones con variables de tipo entero como real. En total, la librería contaba con 10 sustratos de los que venían integrados por defecto sólo 3 en la función principal de optimización del código: $BLX\alpha$, Evolución Diferencial (DE), algoritmo FireFly, Mutación Gaussiana (GM), algoritmo Greedy, Búsqueda Harmónica (HS), Búsqueda Local (LS), MPx, 2Px y Recocido Simulado (SA).

Durante cada iteración, se evalúan las larvas generadas a partir de los diferentes sustratos y se comparan con un número aleatorio de posiciones del coral, las cuales pueden estar vacías u ocupadas por otras larvas. Si la posición evaluada del coral está vacía, la larva se acopla directamente en ese lugar. En caso de que la posición esté ocupada, se realiza un proceso de disputa donde se compara el valor de *fitness* entre ambas larvas y se decide cuál es más adecuada para ocupar ese espacio. Si la nueva larva no logra encontrar un lugar para adherirse después de un número establecido de intentos, desaparece.

A medida que pasan las generaciones, el coral se llenará con las larvas que tengan un mejor valor de *fitness*, y los sustratos predominantes utilizarán los genes de esas larvas para generar nuevas larvas que puedan desafiar a las larvas ya posicionadas y sustituirlas. Debido a su estructura, este procedimiento es más complejo que el que lleva a cabo el BGA, requiriendo más recursos y tiempo para hallar las soluciones óptimas.

5. Descripción de la herramienta software desarrollada

Este capítulo se divide en cuatro secciones que describen cada una de las mejoras implantadas al algoritmo base del CRO-SL de partida o a la función *fitness* utilizada para el cálculo de energía o calcular las distancias entre los aerogeneradores que generan las estelas y los que se ven afectados por ella. En la Sección 5.1 se detalla cómo se ha añadido el nuevo modelo de estela para el cálculo del déficit de velocidad. Las Secciones 5.2, 5.3 y 5.4 se explica el funcionamiento de los 3 sustratos elaborados para su incorporación en el algoritmo CRO-SL.

Por último, en la Sección 5.5 se elabora una función que permite, dados una serie de parámetros, generar una matriz de altitudes para simular los relieves de una zona, y poder realizar los análisis de las distribuciones con desniveles, y ver cuánto afecta en relación con la magnitud de energía generada estimada por los distintos modelos con respecto a los parques eólicos en terreno llano.

5.1. Implementación del modelo de estela Ainslie

El modelo de estela de Ainslie pasa a ser el octavo modelo de estela que se ha programado, teniendo en cuenta los 7 utilizados en trabajos anteriores entre los que figuraban los modelos de estela de Jensen y Larsen, empleados a modo comparativo. Este modelo será incorporado a modo de función para que pueda hacer uso de ella la función de *fitness*, y, al igual que el resto de los modelos que ya se tenían implantados, hará uso de la estructura 'data' que genera y aloja los datos correspondientes al modelo tras inicializarse a través de la función 'genData()'.

Ainslie no definió los límites de su estela de forma directa, por lo que se optó por tomar los límites de estela (D_w) del modelo de Jensen, que provocaba que el déficit estimado por la ecuación gaussiana de Ainslie cerca de la periferia del cono fuese prácticamente nula, lo cual validaba su uso combinado.

Al igual que el resto de los modelos de estela, las ecuaciones que rigen las predicciones de velocidad incidente recogidas en la Sección 2.2.2.3 del Capítulo 2, se almacenan en una función denominada 'wm_ainslie()' cuyos parámetros de entrada serán la estructura 'data', el módulo de los vectores viento (vMod), el indicador de la turbina ('m') y las distancias

longitudinal y radial ('xUDef' e 'yUDef' respectivamente). Esta información es suficiente para llevar a cabo los cálculos de la velocidad deficitaria y estimar la velocidad final incidente sobre el aerogenerador inmerso en las estelas turbulentas.

En el Script 1 del Anexo III se muestra el código de la función del modelo de estela. En ella se puede observar la incorporación de la intensidad de turbulencia adicional de Larsen (I_a), en caso de que se quiera utilizar la intensidad de turbulencia de la estela (I_w) para el cálculo de las predicciones en lugar de la intensidad de estela ambiental (I_0). Para los experimentos se ha usado directamente la intensidad de estela ambiental, la cual recomendaba el autor del modelo en (Ainslie J. F., 1988), y a la que se le ha asignado un valor de 0.035.

5.2. Implementación de 1Px mutativo como sustrato

Debido al buen funcionamiento mostrado por el BGA en terrenos de alta densidad de ocupación (Romero Barrera, 2023), se ha optado por incorporar, a modo de sustrato, el operador de cruce por corte de 1 punto (1Px). El algoritmo CRO-SL inicial, únicamente contaba con los sustratos de cruce de BLX α , MPx y 2Px, lo que hace que se aumente el comportamiento exploratorio de las soluciones en comparación con el uso de un único punto de corte en la cadena de genes. Sin embargo, por la experiencia obtenida del estudio previo, se observó que cambios bruscos en la posición de las turbinas acentuaban el efecto deficitario generado a raíz de las estelas, disminuyendo así la energía obtenida, por lo que, con la implantación de este sustrato, de carácter más conservador, se busca mejorar el comportamiento explotativo del CRO-SL.

Al comienzo de la ejecución del sustrato, se selecciona a otra larva del coral para que se produzca el cruce con la larva perteneciente al sustrato. Para ello, se escoge aleatoriamente a un tercio de las larvas del coral y se les asigna una probabilidad de seleccionarse para el cruce en función del valor de la función de *fitness* asociado. Posteriormente se selecciona un punto aleatorio para realizar el corte en la cadena de genes de ambas larvas, y se recombinan dando lugar a dos nuevas larvas. Se comprueban que no hay genes repetidos en las larvas generadas y posteriormente se escoge a la larva con mejor valor de la función de *fitness*, a la cual se somete a un proceso de mutación con un 15% de probabilidades de llevarse a cabo. Si el proceso de mutación se activa, habrá un 30% de probabilidades de que cada uno de los

genes que componen la larva se modifique aleatoriamente. Finalmente, la larva resultante de todo el proceso pasará a la etapa de asentamiento en el coral posterior.

En el Anexo III se puede encontrar el Script 2 correspondiente al código realizado para la adaptación del operador 1Px del BGA al algoritmo CRO-SL, así como comentarios en las distintas líneas del código para facilitar la comprensión de este.

5.3. Implementación de *Particle Swarm Optimization* (PSO) como sustrato

El algoritmo PSO es una de las alternativas más usadas dentro del campo de los algoritmos metaheurísticos, su capacidad a la hora de tomar referencias del historial de movimiento de cada una de las partículas para definir las velocidades de desplazamiento y fijar una nueva posición, hacen de éste una opción con gran potencial de optimización a la hora de operar con matrices. Sin embargo, el hecho de que se ocupen posiciones discretizadas de terreno genera un entorno de posicionamiento descrito por números enteros, lo cual hace que se reduzca su capacidad de búsqueda de soluciones al no ser operar con coordenadas de terreno continuas.

5.3.1. Ajuste de hiperparámetros

Los algoritmos PSO requieren de un buen ajuste de hiperparámetros (*tuning*) para que muestren un comportamiento adecuado a la hora de optimizar las soluciones. En (Rehman, 2020) se realizó un estudio del efecto de los de los coeficientes de aceleración de las partículas en los resultados obtenidos por el PSO.

Los valores escogidos para ser asignados a los distintos hiperparámetros del algoritmo quedan recogidos en la Tabla 5.1. Se muestran en ella un vector de 3 valores, correspondiente al factor de inercia, y dos vectores de 8 valores perteneciente a las aceleraciones personal y social. En el conjunto del factor de inercia (ω), existe un salto entre valores de 0.2, mientras que en los conjuntos de las aceleraciones (c_1 y c_2) se muestra una diferencia entre valores de 0.1.

Tabla 5.1. Rango de valores para el ajuste de hiperparámetros de movimiento del PSO.

COEFICIENTES	CONJUNTOS DE VALORES
Factor de inercia (ω)	[0.8, 1, 1.2]
Aceleración personal (c_1)	[0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4, 1.5]
Aceleración social (c_2)	[0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4, 1.5]

El *tuning* se ha programado de forma manual sin necesidad de llamar a ninguna librería o función externa de MATLAB. Además, se han adaptado varios de los scripts que componían al código del PSO elaborado en (Romero Barrera, 2023), con el fin de que al ejecutar la nueva estructura de PSO, orientada a ajustar sus hiperparámetros, se vayan comprobando todas las combinaciones de valores de coeficientes especificados. Para ello, es necesario obtener una matriz tridimensional de valores, lo cual se consigue realizando el producto cartesiano de los conjuntos definidos en forma de vectores, a través de la función 'meshgrid()' de MATLAB.

El resultado genera una matriz que (denominada 'joint' en el *script*) de dimensiones 192x3, la cual es almacenada en la estructura 'data' junto con el resto de hiperparámetros del algoritmo y ciertas magnitudes de referencia para el cálculo de la función *fitness*. Posteriormente, es necesario fijar una serie de hiperparámetros del algoritmo para analizar ajustar los 3 coeficientes seleccionados (ω , c_1 , y c_2). Para agilizar el proceso y no demorarlo en exceso, se estable una población de 15 individuos y un número total de 40 iteraciones por cada uno de los 192 conjuntos de hiperparámetros a ajustar.

Tras realizar el proceso iterativo de ajuste, los datos arrojaron que los valores óptimos para los hiperparámetros, dado un terreno de proporciones de 30x30 con 30 turbinas, era el correspondiente a $[c_1, c_2, \omega] = [1.3, 1.0, 1.0]$, el cual arrojó en las 40 iteraciones del PSO, un valor de fitness máximo de $7.7774 \cdot 10^7$ kWh haciendo uso del modelo de estela de Jensen.

Los modelos de estela afectan levemente a la calidad de los resultados optimizados obtenidos, debido en gran parte por los límites de estela que establecen y la penalización en el módulo de velocidad incidente que sufren las turbinas en función de la distancia con el foco generador de la misma. Por esto, el uso de otro modelo de estela a la hora de ajustar los hiperparámetros del PSO puede dar lugar a una combinación de valores distinta que la aquí conseguida.

5.3.2. Adaptación del PSO como sustrato

En este caso, la implantación del PSO a modo de sustrato sigue un funcionamiento equivalente al realizado con el algoritmo PSO individual. Se agrupa cada uno de los genes de las larvas (equivalentes a las partículas en este diseño) en distintos grupos de partículas que seguirán a la partícula líder, la cual pertenecerá a la cadena genética de la larva con mejor *fitness*. Estas partículas tendrán una probabilidad (10%) de cambiar de grupo de partículas con otra de las partículas del mismo individuo, pasando a seguir a una nueva partícula de referencia líder desde la posición en la que se encontraban. Para realizar los cambios de grupo entre las partículas de un individuo, se permuta el orden completo de los genes de la larva, provocando que se altere la referencia de la partícula líder a la que seguían (x_g^{Best}) e intercambiando su mejor posición histórica por la de la otra partícula (x_p^{Best}), lo que provoca que se modifiquen sus patrones de movimiento, haciendo que se ejecuten desde una posición inicial nueva de cara a explorar nuevas soluciones y evitar convergencias.

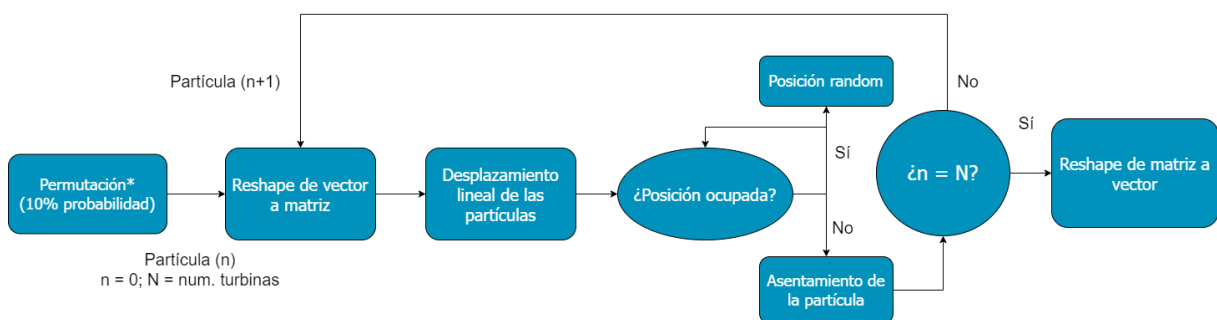


Figura 5.1. Diagrama de funcionamiento del PSO como sustrato dentro del CRO-SL.

En la Figura 5.1 se ha elaborado una representación del flujo de trabajo del sustrato en forma de diagrama. Al comienzo, se pondera la permutación de los genes de la larva cuya probabilidad de llevarse a cabo es de un 10%, a modo de dar un factor exploratorio al sustrato. Posteriormente, la cadena de partículas que componen la larva pasa de su forma de vector al plano matricial de posiciones y se aplican los correspondientes desplazamientos lineales en función del grupo al que pertenezca la partícula y la referencia de dicho grupo. Finalmente se comprueba si no se repiten posiciones dentro de las partículas de la larva y se realiza de nuevo un *reshape* para pasar de la forma de matriz de posiciones a la forma original de la larva. En el Script 3 del Anexo III se puede encontrar el código correspondiente al sustrato.

5.4. Implementación de *Whale Optimization Algorithm* (WOA) como sustrato

El uso de un sustrato basado en el WOA puede ayudar, al igual que el PSO, a romper la duplicidad de individuos en generaciones altas, la cual estará muy presente debido a la tendencia convergente de 4 de los 7 sustratos que utiliza el CRO-SL, basadas en operadores de cruce que recombinan las distintas variables que componen a cada larva del coral.

Para implementar este nuevo sustrato, se ha utilizado la base ya creada para el algoritmo PSO, modificando la parte correspondiente a las ecuaciones de movimiento y los hiperparámetros del propio sustrato. Para establecer el valor de los hiperparámetros se han tomado los valores recomendados en (Mirjalili et al., 2016) y que se pueden consultar en la Tabla 5.2.

Tabla 5.2. Valores de hiperparámetros del WOA.

HIPERPARÁMETRO	VALOR
Amplitud (\vec{A})	[0, 2]
Control de amplitud (\vec{a})	$2 \rightarrow 0$
Contracción (\vec{C})	[0, 2]
Distancia contraída (\vec{D})	[(0, 0), (29, 29)]
Distancia (\vec{D}')	[(0, 0), (59, 59)]

En la Figura 5.2 se representa en forma de diagrama el funcionamiento del sustrato del WOA implementado. Cuenta con un funcionamiento muy similar al visto en el sustrato del PSO, incorporando pequeños cambios en la etapa relativa al movimiento de las turbinas sobre el plano. En este caso, las ballenas de las que se compone la larva tienen dos formas de desplazamiento alternativas, el movimiento directo rectilíneo hacia la posición de referencia o el movimiento con trayectoria en espiral, ambos con un 50% de probabilidades de ejecutarse. También es importante mencionar que la posición de referencia que tomará la ballena variará en función del valor del hiperparámetro \vec{A} , tal y como se explicó en la Sección 2.2.3.4 del Capítulo 2. El código correspondiente al sustrato se puede consultar en el Script 4 del Anexo III.

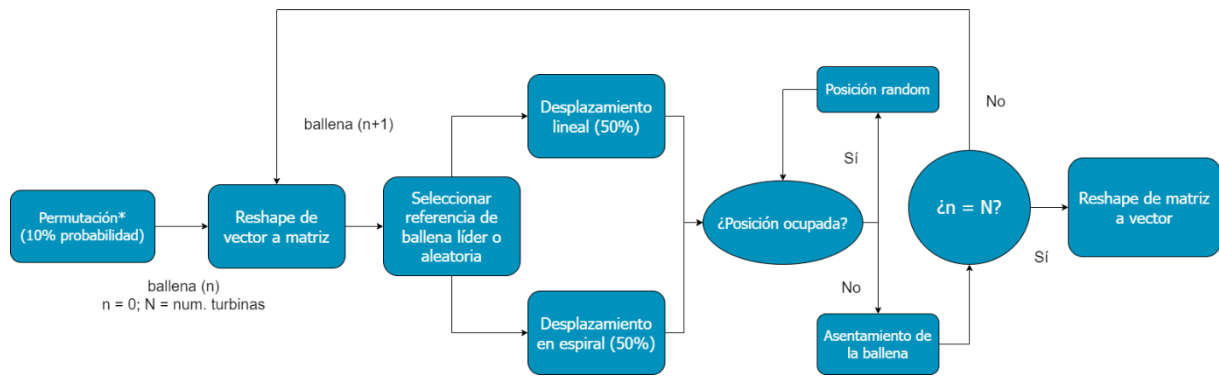


Figura 5.2. Diagrama de funcionamiento del PSO como sustrato dentro del CRO-SL.

5.5. Función generadora de terreno con desniveles y rugosidades superficiales

Esta función está diseñada para generar dos *outputs*, uno correspondiente a una matriz de desniveles que representa la coordenada sobre el eje \hat{z} del plano, modificando a su vez el valor de la longitud radial del cono (r), tomado en coordenadas cilíndricas. Esto provoca que la existencia de un cambio de altitud entre la turbina eólica que genera la estela y aquella que se ve afectada disminuya el déficit de velocidad que se produce, y, por tanto, termine incidiendo un módulo de velocidad del viento superior en las coordenadas especificadas.

Será necesario disponer de los paquetes de MATLAB *Curve Fitting Toolbox* e *Image Processing Toolbox* para poder ejecutar la función 'genRelief()' programada. Para generar la matriz de relieves, se aplica un filtro promediador con una serie de puntos dados por el usuario a modo de *input*, haciendo uso de la función 'fspecial()'.

A la hora de generar la matriz de rugosidad, el usuario deberá establecer una ley de distribución de rugosidad que dependerá del tipo de terreno. En este caso, se ha establecido por defecto la siguiente ley de rugosidad:

$$Z_0 = \frac{z_0}{2} \cdot \frac{Z}{z_{med}} \quad (19)$$

La Ecuación (19) otorga el valor de los componentes de la matriz de rugosidad (Z_0), la cual depende de la longitud de rugosidad de referencia (z_0), la matriz de relieve (Z) y la altitud

media (z_{med}) entre los componentes de 'Z'. Esta ecuación puede modificarse para elaborar otro tipo de distribuciones de rugosidad más complejas.

Posteriormente, se ha adaptado la función 'getDistTrapz()' para que tenga en cuenta el desnivel entre los dos puntos donde se sitúan la turbina eólica generadora de la estela y la que se ve afectada por ella. También, ha sido necesario modificar las ecuaciones que utilizan algunos modelos de estela, como el modelo de Jensen, para incorporar el uso de la rugosidad variable almacenada en la matriz de rugosidad y que recalculen los parámetros dependientes de ella en función del punto en el que se realice el cálculo.

6. Evaluación

En este capítulo se procede a comparar y evaluar los distintos resultados obtenidos tras la ejecución de los procesos de optimización ejecutados, así como el grado de aportación de los nuevos sustratos implementados, dejando el cómputo total de sustratos en 7, en comparación con los valores obtenidos del CRO-SL sin modificar, el cual, contaba únicamente con 4 sustratos. También se realizará un análisis de la función de generación de relieves y cómo afecta al déficit de velocidad predicho por cada uno de los modelos de estela aplicados en el estudio desarrollado.

Para llevar a cabo la evaluación de las herramientas implantadas, se han realizado una serie de experimentos tomando las siguientes hipótesis:

- Flujo subsónico incompresible ($M^a < 0.3$) y estacionario.
- Ángulo de resbalamiento de las palas del rotor nulo ($\beta = 0$) y la velocidad incidente será siempre perpendicular al plano que contiene al rotor.
- Todas las turbinas eólicas serán del modelo AWEC-60 y tendrán las mismas características geométricas y curva de potencia (vista en la Sección 4.2 del Capítulo 4).
- Atmósfera en calma y densidad prácticamente constante en todo el terreno analizado.

De igual forma, también es necesario contar con una serie de parámetros característicos de la propia turbina eólica y del terreno para proceder al cálculo correspondiente. En la Tabla 6.1 se recogen los valores asociados a los parámetros geométricos de las turbinas eólicas y las propiedades del entorno, estando algunos de estos datos extraídos de (Zergane et al., 2018).

Tabla 6.1. Características geométricas de los aerogeneradores y del entorno.

z (m)	D_r (m)	c_T	A_{terreno} (km ²)	d_{sec} (m)	z_0 (m)	I_0
60	80	0.88	144	400	0.3	0.035

6.1. Comparativa de resultados del proceso de optimización

Para realizar la comparativa, es necesario conocer la energía máxima que pueden generar 30 turbinas en la mejor disposición posible (sin solapamiento de estelas) y en la peor de ellas (solapamiento total alineado), de forma que se pueda obtener la energía máxima y mínima

obtenible por los dispositivos. Con esta información, es posible realizar el cálculo de rendimiento de las distintas soluciones en base a los valores máximos y mínimos de energía obtenibles por modelo a través de la Ecuación (20), la cual da una referencia porcentual de la aptitud de las distintas soluciones.

$$Ren(\%) = \frac{E_{gen} - E_{mín}}{E_{máx} - E_{mín}} \cdot 100$$

(20)

La energía mínima se calcula introduciendo como inputs, de forma manual, un valor fijo y constante del ángulo del vector dirección del viento manteniendo su módulo, y alineando las turbinas a lo largo del terreno de forma paralela a dicho vector. De forma más sencilla se puede calcular la energía máxima, ya que simplemente hay que realizar el cálculo de la energía producida directamente con los módulos de la velocidad del vector de vientos, sin descontarle los déficits producidos por las estelas, es decir, no actuarán los modelos y la energía máxima será universal independientemente del modelo de estela escogido para el cálculo. En la Tabla 6.2 se pueden consultar los valores obtenidos de energía máxima y mínima producidas para una configuración de proporciones de 30x30 con este vector de vientos establecido.

Tabla 6.2. *Producción de energía máxima y mínima por modelo de estela.*

Modelo	Energía mínima (kWh)	Energía máxima (kWh)
Jensen	$4.0803 \cdot 10^7$	$7.8064 \cdot 10^7$
Larsen	$9.5351 \cdot 10^6$	$7.8064 \cdot 10^7$
Ainslie	$7.1726 \cdot 10^7$	$7.8064 \cdot 10^7$

Tras los procesos de optimización con los distintos algoritmos probados (BGA, CRO-SL inicial de 4 sustratos y CRO-SL modificado de 7 sustratos), se han obtenido las distribuciones optimizadas a lo largo de 1000 generaciones de una población con un total de 100 individuos establecidos. El uso del BGA como del CRO-SL de 4 sustratos nos sirven, a modo comparativo, del grado de mejora que supone la inclusión de los nuevos sustratos elaborados durante este estudio, mientras que la comparación con el BGA, que mostró mejores resultados para tasas

altas de ocupación superiores al 2% (en este caso del 3.33%), se utiliza para comprobar una metodología de optimización ajena al CRO y comprobar el rendimiento individual del operador 1Px sin estar combinado con otros operadores a modo de sustrato.

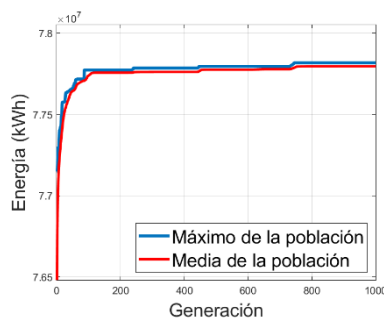
Observando los resultados de la Tabla 6.3, se puede dirimir cuál de los tres algoritmos funciona de manera más adecuada de cara a maximizar la eficiencia del parque eólico descrito, formado por 30 aerogeneradores, con unas proporciones de terreno de 30x30 y la matriz de vientos real utilizada para los cálculos.

El BGA sigue siendo el algoritmo que más rápido alcanza las 1000 generaciones, estando ambas versiones del CRO-SL lejos de sus tiempos de ejecución. En cuanto a los resultados de energía producida correspondientes al valor de la función de *fitness* de cada una de las mejores distribuciones halladas, se puede concluir que la implementación de los 3 nuevos sustratos en el CRO-SL ha sido satisfactoria, dominando entre el conjunto de soluciones obtenidas por cada algoritmo, en cada uno de los 3 modelos de estela analizados. Los valores de rendimiento obtenidos por esta versión modificada del CRO-SL mejoran notablemente los resultados, sobre todo con los modelos de Larsen y Ainslie, siendo más tenue la mejora con el modelo de Jensen.

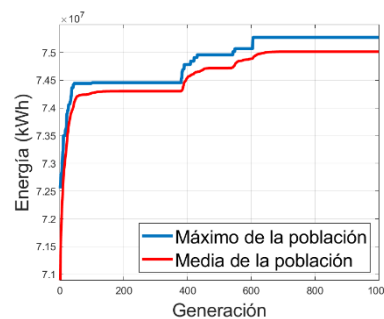
Tabla 6.3. *Valores de energía producida y rendimiento de las soluciones.*

Algoritmo	Modelo	Energía producida (kWh)	Rendimiento (%)
BGA	Jensen	$7.7818 \cdot 10^7$	99.34
	Larsen	$7.5273 \cdot 10^7$	95.93
	Ainslie	$7.7896 \cdot 10^7$	97.35
CRO-SL (4 sustratos)	Jensen	$7.7722 \cdot 10^7$	99.08
	Larsen	$7.4619 \cdot 10^7$	94.97
	Ainslie	$7.7882 \cdot 10^7$	97.13
CRO-SL (7 sustratos)	Jensen	$7.7856 \cdot 10^7$	99.44
	Larsen	$7.6144 \cdot 10^7$	97.20
	Ainslie	$7.8004 \cdot 10^7$	99.05

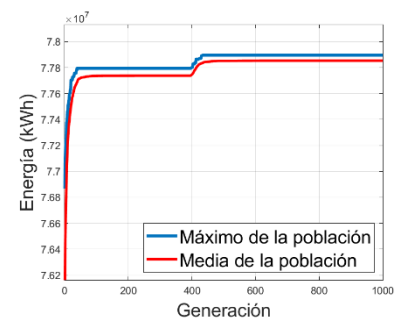
En la Figura 6.1 se representa la evolución de los valores de energía generada (valor de la función *fitness*) por la mejor de las distribuciones y la media de los valores conseguidos por todas las distribuciones en cada una de las generaciones, optimizadas mediante el uso del BGA. En las gráficas contenidas en la figura se puede ver un crecimiento rápido de los valores de energía, situándose la media poblacional siempre por debajo del valor de *fitness* del mejor individuo. Ambos valores de la función *fitness* representados mantienen un crecimiento suave y constante a lo largo de las generaciones consiguiendo siempre un valor igual o superior al de la generación anterior.



(a) Jensen



(b) Larsen



(c) Ainslie

Figura 6.1. Gráficas de evolución de la función de *fitness* en BGA.

La Figura 6.2 expone 3 gráficas que describen la evolución de la energía generada por las distintas soluciones obtenidas por el CRO-SL de 4 sustratos. Se puede observar una tendencia creciente de las soluciones máximas y una leve oscilación entre la media de energía generada entre todas las soluciones de una misma generación, esta tendencia debe a una serie de factores aleatorios como la defunción de larvas (soluciones) o al operador de torneo en el que dos larvas compiten por un mismo hueco del coral, existiendo una leve posibilidad de que se establezca la larva con peor *fitness* a fin de aumentar el comportamiento exploratorio del algoritmo.

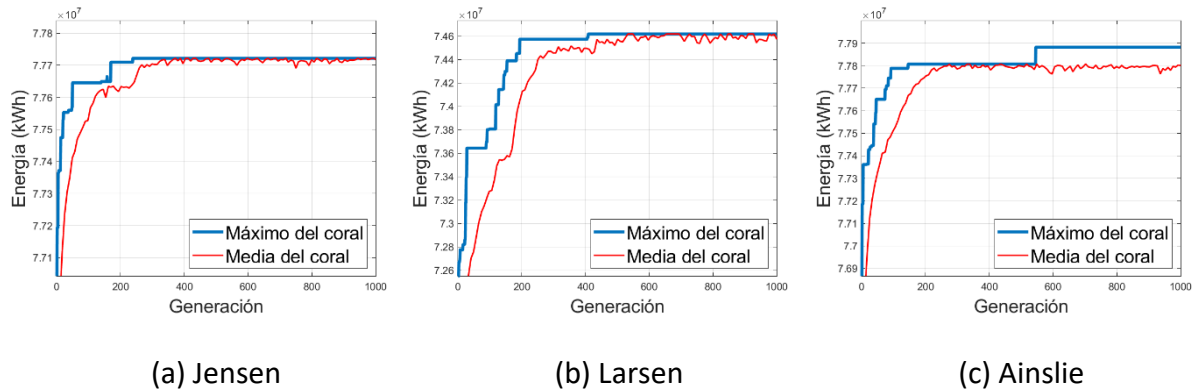


Figura 6.2. Gráficas de evolución de la función de fitness en algoritmo CRO-SL (4 sustratos).

En las gráficas recogidas en la Figura 6.3 se muestra la evolución de los valores de energía generada por la mejor de las soluciones y los valores medios de todo el conjunto analiza en cada una de las generaciones. Al igual que en las gráficas de la Figura 6.2, se muestra una tendencia ascendente de los valores dados por la mejor de las soluciones y oscilante por parte de la media del conjunto de soluciones.

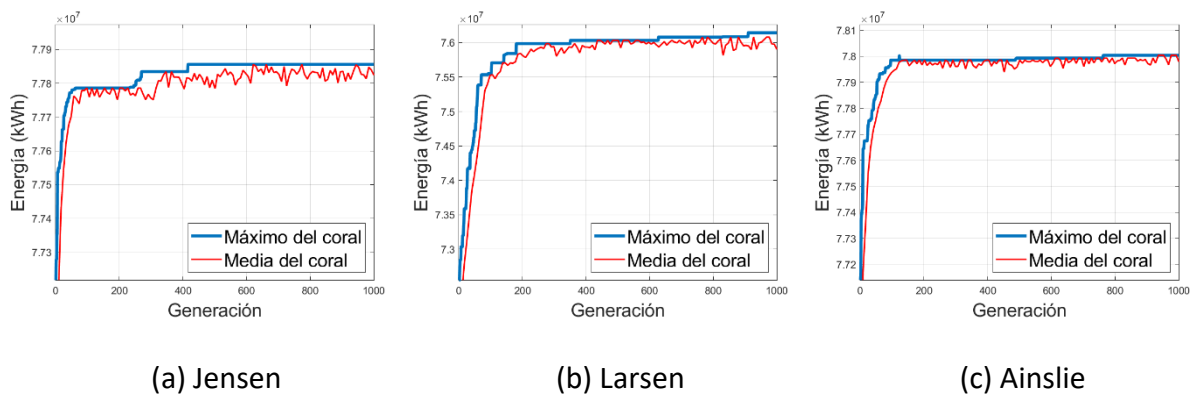
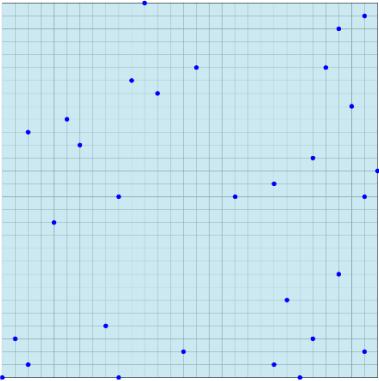
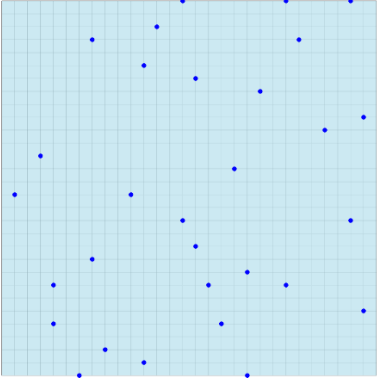
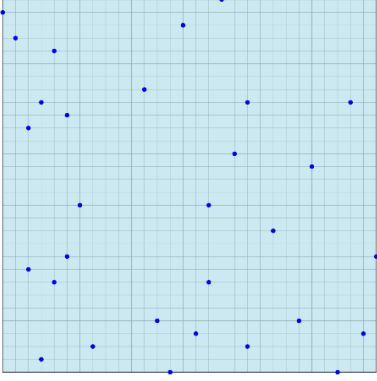
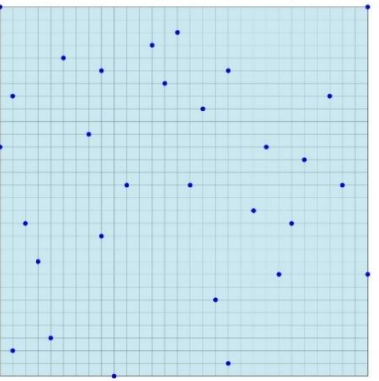
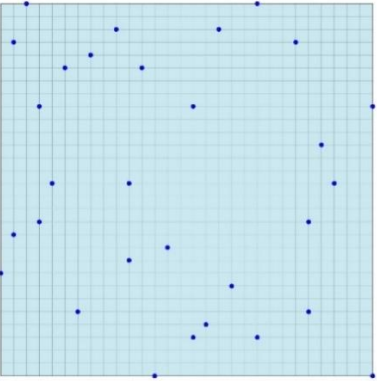
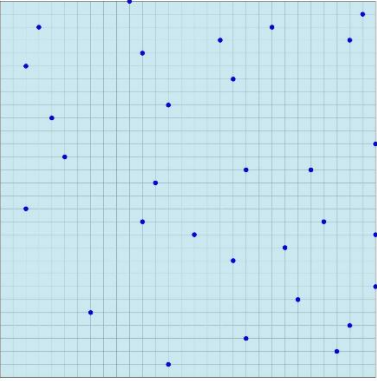
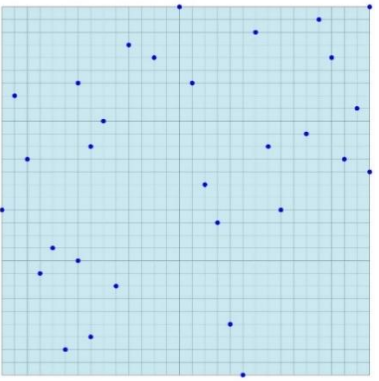
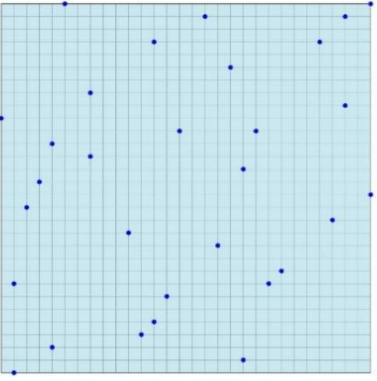
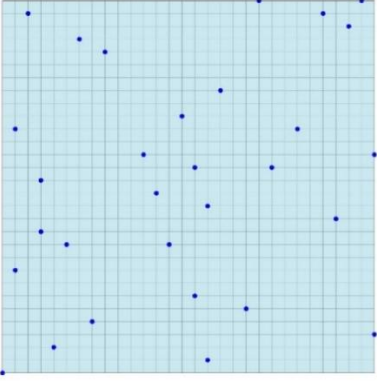


Figura 6.3. Gráficas de evolución de la función de fitness en algoritmo CRO-SL (7 sustratos).

Se puede destacar que la tendencia de las soluciones alcanzadas por el algoritmo CRO-SL de 7 sustratos muestra un crecimiento más acentuado y rápido que los observados en los obtenidos con 4 sustratos. Esto implica que los valores medios de las soluciones conseguidas con 7 sustratos se sitúan más cerca del valor máximo alcanzado desde generaciones tempranas, mientras que, con 4 sustratos, es necesario situarse en generaciones bastante avanzadas en comparación. Las distribuciones optimizadas de turbinas eólicas correspondientes a los valores de energía producida por cada uno de los algoritmos utilizados, de la Tabla 6.3 quedan recogidas en la Tabla 6.4.

Tabla 6.4. Distribuciones optimizadas de aerogeneradores por los algoritmos metaheurísticos.

BGA		
		
(a) Jensen	(b) Larsen	(c) Ainslie
CRO-SL (4 sustratos)		
		
(a) Jensen	(b) Larsen	(c) Ainslie
CRO-SL (7 sustratos)		
		
(a) Jensen	(b) Larsen	(c) Ainslie

6.2. Comparativa de la participación e importancia de cada sustrato

En la Tabla 6.5 se puede observar el número de generaciones que ha dominado cada uno de los sustratos, obteniendo la mejor de las larvas, tanto para el CRO-SL estándar con 4 sustratos (3 de cruce y 1 de mutación) como para el CRO-SL modificado con 7 sustratos (4 de cruce, 2 de comportamiento de enjambre y 1 de mutación). Entre los resultados del CRO-SL compuesto de 4 sustratos, nos encontramos una participación muy baja de la mutación gaussiana (GM) en comparación con los otros tres sustratos, siendo el cruce producido por el operador $BLX\alpha$ el sustrato que más veces ha generado la mejor de las larvas a lo largo de las 1000 generaciones establecidas.

Tabla 6.5. Número de generaciones en las que cada sustrato ha generado la mejor larva.

Algoritmo	Modelo	$BLX\alpha$	MPx	2Px	GM	1Px	PSO	WOA
CRO-SL (4 sustratos)	Jensen	385	265	305	55	-	-	-
	Larsen	354	290	316	40	-	-	-
	Ainslie	376	282	304	38	-	-	-
CRO-SL (7 sustratos)	Jensen	164	168	257	1	237	173	0
	Larsen	147	199	241	0	251	162	0
	Ainslie	188	219	223	1	195	172	2

Resulta interesante observar las generaciones en las que se alcanza el mejor valor de *fitness* de la población y la participación que tenían los sustratos por aquel entonces, los cuales, quedan recogidos en la Tabla 6.6. El modelo de estela de Jensen alcanzó el máximo en una generación más prematura que los otros dos modelos, pudiendo deberse a una actuación más eficaz del sustrato del PSO, cuyos hiperparámetros se optimizaron haciendo uso de este modelo. Tanto el modelo de Larsen como el de Ainslie obtienen sus mejores soluciones en el segundo tramo generacional, cuando ya las soluciones están más asentadas y muchos de los individuos son idénticos debido a la cantidad de recombinación de genes que ha sufrido la

población por los cuatro operadores de cruce que incorpora el algoritmo. Se observa una tendencia muy similar a la observada en la Tabla 6.5, situándose el sustrato correspondiente al PSO entre la cuarta y quinta posición, mientras que el 1Px se suele situar entre los tres sustratos con mayor participación.

Tabla 6.6. Participación de sustratos en la generación en la que se alcanzó el fitness máximo.

Modelo	Generación	BLX α	MPx	2Px	GM	1Px	PSO	WOA
Jensen	416	63	86	95	1	93	78	0
Larsen	910	136	179	216	0	228	151	0
Ainslie	778	137	172	183	1	156	127	2

Observando la participación de cada uno de los sustratos a lo largo de las distintas generaciones, se puede detectar que el impacto que tienen ciertos sustratos es bastante más notorio en las iteraciones iniciales, cuando el coral cuenta con valores medios de fitness relativamente bajos. Esto es debido al hecho de que las larvas que ocupan los huecos del coral son fácilmente sustituibles, y, por tanto, todos los sustratos tienden a generar soluciones suficientemente buenas como para que las nuevas larvas se asienten en el coral sustituyendo a las larvas establecidas u ocupen un hueco de libre.

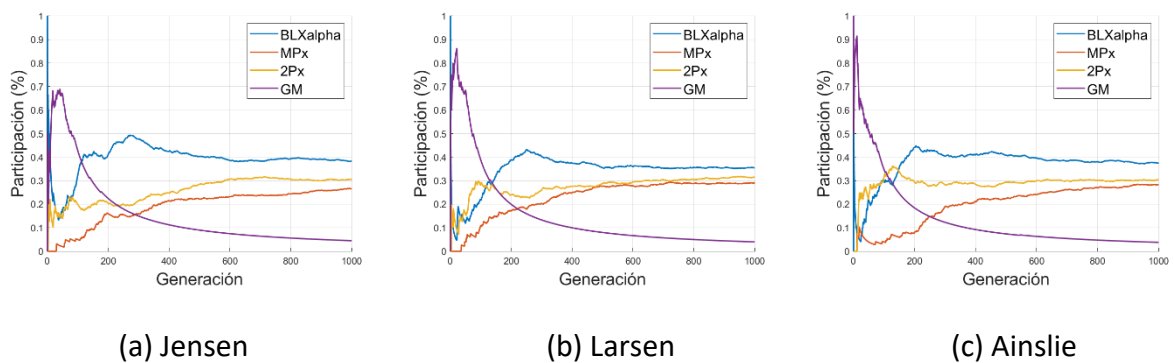


Figura 6.4. Evolución de participación de los sustratos del CRO-SL (4 sustratos).

Como se puede observar en las gráficas de la Figura 6.5 que muestran la participación generacional de cada uno de los sustratos en el CRO-SL modificado, en los modelos de Ainslie

y Larsen, el sustrato con la metodología de GM implantada muestra un cierto grado de participación en torno a las primeras 100 generaciones, que termina anulándose por completo posteriormente. En lo referente al sustrato WOA que se añadió, éste muestra una participación prácticamente nula en los modelos de Jensen y Larsen, mientras que en el de Ainslie describe un comportamiento muy similar al descrito para el sustrato de GM.

Tanto los sustratos incorporados del PSO como el 1Px muestran un comportamiento muy satisfactorio en el desarrollo de nuevas soluciones óptimas, siendo el sustrato 1Px uno de los sustratos más activos junto con el 2Px y el MPx en función del modelo de estela, mientras que el sustrato de PSO describe una tendencia de participación en la sustitución de larvas del coral muy similar a la demostrada por el sustrato de BLX α .

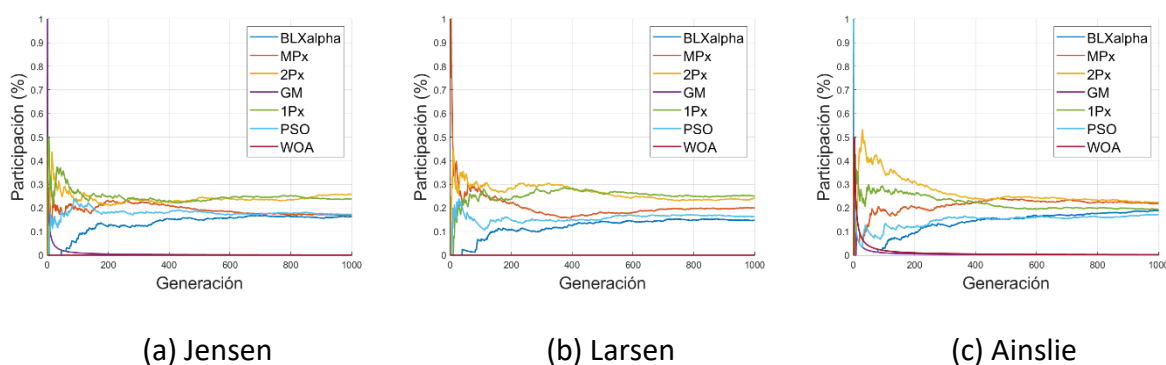


Figura 6.5. Evolución de participación de los sustratos del CRO-SL (7 sustratos).

La tendencia de los sustratos es a la estabilización de su grado de participación, quedando los sustratos menos efectivos anulados y mostrando un comportamiento más parejo y paralelo entre los porcentajes de participación del resto.

6.3. Producto cartesiano de soluciones de CRO-SL de 7 sustratos

Para hallar la mejor de todas las soluciones obtenidas, es necesario realizar el producto cartesiano de los conjuntos de los modelos de estela {WM} y las soluciones optimizadas {S}. Dado que los mejores valores de fitness se obtuvieron con el algoritmo CRO-SL de 7 sustratos, únicamente se tienen en cuenta las 3 soluciones conseguidas con él para conformar el conjunto {S}, haciendo un total de 9 combinaciones posibles resultantes del producto

cartesiano $\{WM\} \times \{S\}$. En la Figura 6.6 se representa un diagrama de la aplicación del producto cartesiano sobre los dos conjuntos mencionados.

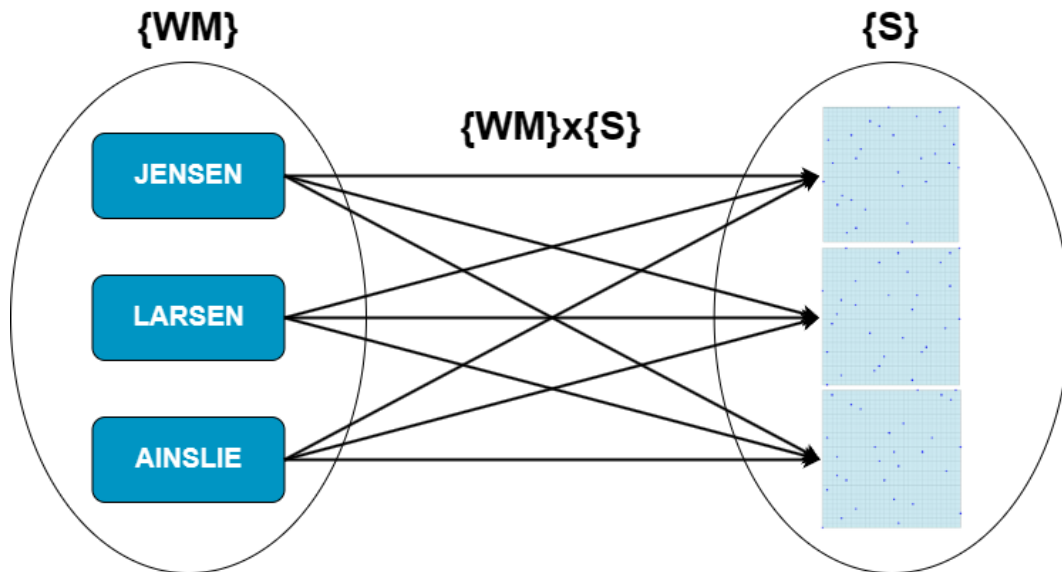


Figura 6.6. Diagrama del Producto cartesiano de conjuntos $\{WM\}$ y $\{S\}$.

Para estimar cuál de las 3 soluciones es la mejor, se realiza una comparativa del rendimiento promedio obtenido por las distribuciones de aerogeneradores para los tres modelos planteados. En la Tabla 6.7 se muestran los valores de energía producida estimados por cada uno de los tres modelos de estela para las tres soluciones optimizadas seleccionadas. Como se puede observar, la solución conseguida por el modelo de Larsen consigue los mejores valores medios de producción, en parte debido a que penaliza mucho a las otras dos distribuciones de aerogeneradores conseguidas por el resto de los modelos.

Tabla 6.7. Producto cruzado de las soluciones con los modelos de estela (fitness, kWh).

Modelo	Solución		
	JENSEN	LARSEN	AINSLIE
JENSEN	$7.7856 \cdot 10^7$	$7.6891 \cdot 10^7$	$7.5181 \cdot 10^7$
LARSEN	$7.2209 \cdot 10^7$	$7.6144 \cdot 10^7$	$7.2182 \cdot 10^7$
AINSLIE	$7.7828 \cdot 10^7$	$7.7240 \cdot 10^7$	$7.8004 \cdot 10^7$
FITNESS MEDIO DE LA SOLUCIÓN	$7.5964 \cdot 10^7$	$7.6758 \cdot 10^7$	$7.5122 \cdot 10^7$

Sin embargo, dado que cada modelo cuenta con unas características propias, el hecho de conocer la energía media estimada entre los 3 modelos para las soluciones no es indicativo de que sea buena solución. Para realmente estimar cómo de apta es una distribución, es necesario acudir al rendimiento medio de las mismas, tomado a través de la Ecuación con los datos de energía máxima y mínima estimada por los modelos de estela, recogidos en la Tabla 6.2 de la Sección 6.1. A través de los rendimientos mostrados en la Tabla 6.8, se ha estimado que la mejor distribución de turbinas eólicas fue hallada por el modelo de estela de Jensen haciendo uso del algoritmo CRO-SL de 7 sustratos.

Tabla 6.8. *Producto cruzado de las soluciones con los modelos de estela (rendimiento).*

Modelo	Solución		
	JENSEN	LARSEN	AINSLIE
JENSEN	99.44%	96.85%	92.26%
LARSEN	91.45%	97.20%	91.41%
AINSLIE	96.27%	86.99%	99.05%
RENDIMIENTO DE LA SOLUCIÓN	95.88%	93.68%	94.24%

6.4. Análisis de la afectación del relieve y los cambios de rugosidad

Gracias a la función desarrollada, se pueden generar matrices de altitudes y rugosidad del terreno que permitan realizar un cálculo más realista de la energía generada por las turbinas eólicas. Estas modificaciones pueden suponer cambios significativos en la cantidad de energía generada por una distribución de aerogeneradores dada, lo que puede llevar a resultados más próximos a los que se obtendrían de mediciones reales. Esta implementación supone un gran avance a la hora de realizar cálculos para la optimización de las distribuciones de aerogeneradores, ya que amplía el rango de terrenos que se pueden simular en futuros proyectos.

En la Sección 6.4.1 se realizan los experimentos relacionados con el mapa de relieve usando el modelo de estela de Ainslie, mientras que en la Sección 6.4.2 los experimentos realizados

estarán orientados a medir el impacto que genera la inclusión de un rango variable de rugosidades en los cálculos con el modelo de estela de Jensen. Se tomarán todas las soluciones optimizadas obtenidas para ambos casos.

6.4.1. Influencia de los desniveles del terreno en la energía generada

El hecho de incorporar desniveles conlleva un incremento en la distancia entre las turbinas eólicas y un aumento de la superficie de terreno vigente dentro de los márgenes establecidos para su proyección en planta, designado por un cuadrado de 12 km de lado tal y como se indica en la Tabla 6.1 mostrada en la introducción del Capítulo 6.

Como bien es sabido, tras la explicación de los distintos modelos de estela expuesta en la Sección 2.2.2 del Capítulo 2, un aumento de la distancia longitudinal y radial de los ejes locales del cono de la estela entre el aerogenerador que la produce y el que se ve envuelto en ella, produce una disminución del déficit de velocidad sufrido y, por tanto, incidirá sobre la turbina eólica afectada un módulo de velocidad el viento mayor.

En la Figura 6.7 se pueden apreciar las tres representaciones del mapa de relieve utilizado con el que se ha elaborado un experimento para comprobar la incidencia del relieve del terreno, analizando con el modelo de estela de Ainslie implementado, las tres distribuciones de aerogeneradores optimizadas por el algoritmo CRO-SL de 7 sustratos.

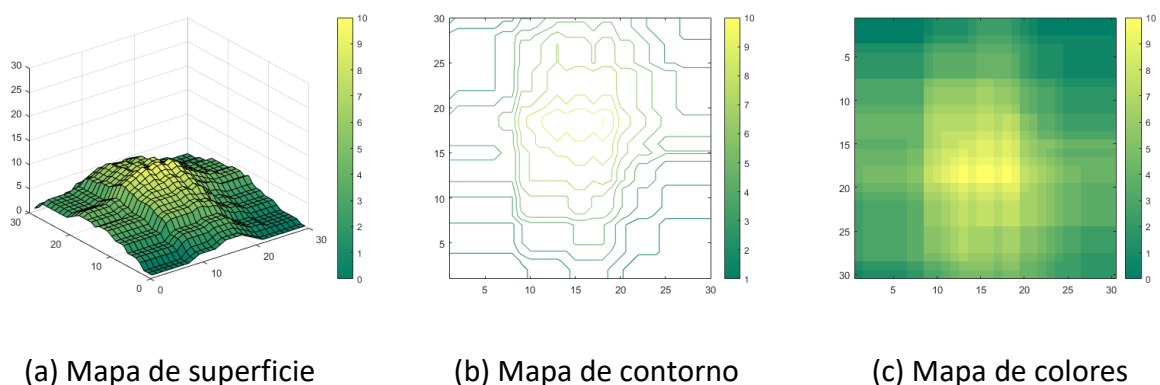


Figura 6.7. Representación del mapa de relieve utilizado.

Las cotas máximas y mínimas de la altitud del terreno se han escalado al rango comprendido entre $[0, 10]$ m. A mayor rango de altitudes, mayor será el incremento de energía que producirá la distribución de turbinas eólicas.

Se ha seleccionado el modelo de estela de Ainslie debido a que es el modelo que menos diferencia muestra entre la energía máxima y mínima producida, mostradas en la Tabla 6.2 de la Sección 6.1. Esto provoca que cualquier cambio en los parámetros de cálculo del déficit de velocidad incidente afecten notoriamente más que en otros modelos, y, por ende, sea más sencillo observar cambios en las magnitudes de interés.

Tabla 6.9. *Impacto del relieve en la energía producida por las distribuciones (modelo de Ainslie).*

Algoritmo	Solución	Energía producida con relieve (kWh)	Energía producida sin relieve (kWh)
BGA	Jensen	$7.7737 \cdot 10^7$	$7.7736 \cdot 10^7$
	Larsen	$7.7499 \cdot 10^7$	$7.7736 \cdot 10^7$
	Ainslie	$7.7935 \cdot 10^7$	$7.7896 \cdot 10^7$
CRO-SL (4 sustratos)	Jensen	$7.7658 \cdot 10^7$	$7.7656 \cdot 10^7$
	Larsen	$7.7184 \cdot 10^7$	$7.7182 \cdot 10^7$
	Ainslie	$7.7925 \cdot 10^7$	$7.7882 \cdot 10^7$
CRO-SL (7 sustratos)	Jensen	$7.7829 \cdot 10^7$	$7.7828 \cdot 10^7$
	Larsen	$7.7526 \cdot 10^7$	$7.7524 \cdot 10^7$
	Ainslie	$7.8005 \cdot 10^7$	$7.8004 \cdot 10^7$

Se puede observar una leve incidencia de los desniveles del relieve en la producción de energía de las distintas distribuciones obtenidas por los algoritmos, cuyos valores de *fitness* se recalcularon con el modelo de Ainslie, independientemente del modelo utilizado en el proceso de optimización. Como ya se mencionó, el hecho de incorporar un mapa de altitudes al terreno provoca un aumento en la distancia radial del cono de estela, lo que aumenta la distancia entre turbinas eólicas, disminuyendo el efecto que tiene la estela sobre la velocidad incidente. Con este experimento, se ha demostrado que la adaptación del cálculo de distancias que utiliza la función de fitness para acoplar la matriz de relieves ha sido exitosa. Dado que se

trabajan con extensiones de terreno muy amplias debido a que se simulan aerogeneradores comerciales y se aplica una distancia de seguridad equivalente a 5 veces el diámetro del rotor de la turbina eólica, se podría aplicar una magnitud de desnivel todavía mayor, de manera que la diferencia entre las configuraciones tomadas en el experimento de esta sección fuese más notoria.

6.4.2. Influencia de los cambios de rugosidad superficial en la energía generada

Para la evaluación de los cambios de rugosidad del terreno, se ha hecho uso del modelo de estela de Jensen, el cual emplea este parámetro de forma directa tanto en el cálculo del déficit de velocidad como en la pendiente que describe su cono de estela. En este caso, se ha tomado la hipótesis de que la estela descrita por Jensen mantiene una pendiente constante dada por el valor de ' z_0 ' en el punto de generación de la estela a la hora de delimitarla. Sin embargo, para el cálculo del valor de la velocidad en un punto afectado por la estela descrita en la Ecuación (2), la pendiente del cono de la estela (k) sí tomará el valor dado al emplear en la ecuación la ' z_0 ' de dicho punto y no de donde se genera la estela, recalculando de esa forma la Ecuación (3) expuesta en la Sección 2.2.2.1 del Capítulo 2. Estos cambios provocarán ligeras variaciones no uniformes de los resultados que dependerán de los valores asignados a cada punto del terreno y que podrán incrementar o disminuir los resultados conseguidos con el valor de longitud de rugosidad superficial constante.

Es importante poder visualizar el tipo de rugosidades que tiene el terreno para poder hacer una aproximación de las zonas más óptimas para la colocación de los aerogeneradores, por ello, la Figura 6.8 recoge los distintos mapas de representación de la rugosidad escogido para llevar a cabo una serie de pruebas que verifiquen el correcto funcionamiento de la función 'genRelief()' que genera la matriz de rugosidad y relieve, así como la adaptación del modelo de estela de Jensen para utilizar los valores de dicha función en lugar de tomar un valor constante.

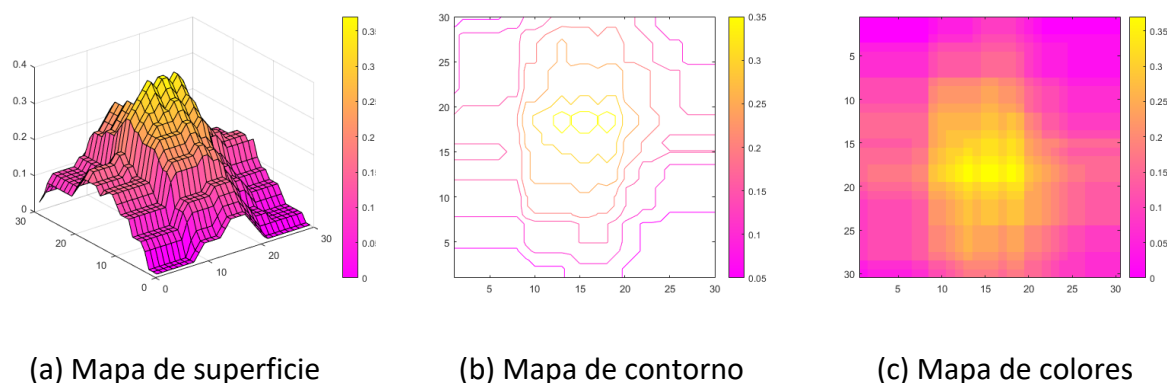


Figura 6.8. Representación del mapa de rugosidad superficial utilizado.

En la Tabla 6.10 se recogen los resultados de los experimentos llevados a cabo con el modelo de estela de Jensen, aplicando rugosidades variables o manteniéndola constante. Es importante destacar, que, en el caso planteado, el uso de la matriz de rugosidades generada junto con la ley de distribución de longitudes de rugosidad establecida, provocan una disminución de todas las distribuciones óptimas analizadas a través del modelo de estela Jensen, el cual utiliza de forma directa el valor de z_0 en el cálculo del déficit de velocidad.

Tabla 6.10. Impacto de la longitud de rugosidad en la energía producida (modelo de Jensen).

Algoritmo	Solución	Energía producida con rugosidad variable (kWh)	Energía producida con rugosidad constante (kWh)
BGA	Jensen	$7.5866 \cdot 10^7$	$7.7818 \cdot 10^7$
	Larsen	$7.7256 \cdot 10^7$	$7.7444 \cdot 10^7$
	Ainslie	$7.5477 \cdot 10^7$	$7.7678 \cdot 10^7$
CRO-SL (4 sustratos)	Jensen	$7.7023 \cdot 10^7$	$7.7722 \cdot 10^7$
	Larsen	$7.6644 \cdot 10^7$	$7.7244 \cdot 10^7$
	Ainslie	$7.7422 \cdot 10^7$	$7.7547 \cdot 10^7$
CRO-SL (7 sustratos)	Jensen	$7.7547 \cdot 10^7$	$7.7856 \cdot 10^7$
	Larsen	$7.6153 \cdot 10^7$	$7.7722 \cdot 10^7$
	Ainslie	$7.7499 \cdot 10^7$	$7.7606 \cdot 10^7$

Denotar que muchos modelos no hacen uso de la longitud de rugosidad superficial para sus cálculos, o suponen un valor constante de ésta, véase el modelo Park2 (Rathmann et al., 2018). Esto hace que esta implementación tenga un alcance muy reducido, pudiendo usarse únicamente en una gama muy reducida de modelos, casi todos basados en el modelo de estela de Jensen. Además, dado que la ley de rugosidad establecida sobre el terreno en este experimento depende de la diferencia de altitud, esto limita de igual forma su aplicación al estudio de granjas *onshore* con desniveles.

6.5. Análisis del modelo de estela de Ainslie

Tras la incorporación del modelo de Ainslie, se ha podido observar que el déficit de velocidad producido por su estela depende mucho del valor asignado al parámetro σ_θ , el cual depende en gran parte de la rugosidad superficial del terreno. Valores bajos de σ_θ provocan disminuciones mayores en la velocidad incidente del viento interno de la estela, por lo que es necesario ajustar minuciosamente dicha magnitud para tener un comportamiento adecuado del modelo.

El modelo de Ainslie, al igual que el modelo de estela descrito por Larsen, hace que sea necesario el uso de un número considerable de ecuaciones implícitas para poder obtener el valor de la velocidad incidente en un punto afectado por las estelas. Este hecho hace que aumenten los tiempos de cálculo del modelo de estela que, en otros modelos compuestos por ecuaciones más simples como el modelo de estela de Jensen, cuya ejecución es notablemente más alta. En la Tabla 6.11 se pueden observar los tiempos de ejecución promedio de los 3 modelos de estela empleados en las comparativas, pudiendo observar que el planteado por Ainslie se posiciona como una alternativa intermedia entre los modelos de Jensen y Larsen, en lo referente a tiempo de inversión para el cálculo de soluciones.

Tabla 6.11. *Tiempos de ejecución por modelo de estela en configuración de 30 aerogeneradores.*

Modelo	Tiempo de ejecución en 20 iteraciones (s)	Tiempo de ejecución promedio por iteración (s)
Jensen	25.2272	1.2614
Larsen	32.2178	1.6109
Ainslie	28.3428	1.4171

Haciendo alusión a la Tabla 6.2 expuesta en la Sección 6.1, se puede destacar que el modelo de Ainslie tiene poca penalización a la hora de describir su estela, siendo el valor mínimo de energía generada el más alto de entre los 3 modelos comparados, lo que provoca que el margen comprendido entre el mínimo y el máximo sea el más reducido en cuanto a valores posibles, cuyas magnitudes alcanzan cotas relativamente muy altas en comparación con los otros dos modelos. Este modelo ha sido de gran utilidad para comprobar el nivel de afectación de la adición de la matriz de desniveles a la energía generada durante la Sección 6.4.1, ya que los otros dos modelos utilizados en este estudio (Jensen y Larsen) no mostraban cambios ante un desnivel máximo tan bajo como el establecido de 10 m.

7. Conclusiones y trabajo futuro

En este capítulo se recogen las conclusiones tras el análisis de los distintos procesos de optimización y la evaluación de las distintas herramientas de software implantadas en la función de *fitness* y el algoritmo CRO-SL. La Sección 7.1 se dedica a la exposición de dichas conclusiones para posteriormente, en la Sección 7.2, analizar el cumplimiento de los objetivos expuestos en las Secciones 3.1 y 3.2 del Capítulo 3. Finalmente, para concluir esta memoria, se establecen una serie de líneas futuras de investigación en la Sección 7.4, las cuales, se pretenden abordar en estudios posteriores.

7.1. Conclusiones

Ante la problemática que surge a la hora de colocar aerogeneradores sobre un terreno establecido para la instalación y puesta en marcha de un parque eólico, surge el dilema de la metodología a utilizar para solucionar el problema. Se decidió optimizar una configuración de un terreno discretizado de proporciones 30x30 con 30 turbinas eólicas a posicionar, a través del algoritmo CRO-SL, y buscar mejorar sus resultados a través de la incorporación de 3 nuevos sustratos basados en el operador 1Px, el algoritmo PSO y el WOA. También para complementar la herramienta y poder llevar a cabo análisis más realistas, se implementó un nuevo modelo de estela en la librería de modelos elaborada (modelo de Ainslie) y se procedió a la creación de una función capaz de aportar nuevos datos a la función de *fitness* como terrenos con altitud no uniforme y distintas rugosidades superficiales asociadas.

La evolución del valor de la función *fitness* del mejor individuo de la población en el BGA sigue una tendencia de crecimiento bastante suave y continua, mientras que la seguida por el CRO-SL, sobre todo con 4 sustratos, tiende a tener un crecimiento más escalonado debido a la duración superior de los periodos de estancamiento y los cambios bruscos del mejor valor de *fitness* de las soluciones. Al implementar los nuevos sustratos, estableciendo así una configuración total de 7 sustratos, el CRO-SL suaviza notoriamente la curva de crecimiento e incrementa la velocidad de obtención de nuevas soluciones válidas para establecerse dentro del coral, incrementando así los valores de energía generada máxima y media de las soluciones encontradas.

Respecto a la valoración de la implementación de los 3 sustratos (1Px, PSO y WOA), se puede concluir que la adición de estos sustratos beneficia globalmente a la obtención de soluciones, permitiendo alcanzar distribuciones de aerogeneradores con mayores valores de producción de energía. Sin embargo, a pesar de esto, sólo 2 sustratos, el 1Px y PSO, han demostrado tener una gran repercusión sobre la búsqueda de nuevas soluciones aptas para incorporarse al coral. El WOA ha tenido una participación paupérrima, limitándose a intervenir en generaciones muy tempranas y desapareciendo posteriormente conforme avanzaban las iteraciones. A pesar de esto, el resultado final de la implementación ha sido muy satisfactorio y se puede decir que el beneficio general sacado del uso de los nuevos sustratos diseñados es notoriamente alto.

Los resultados vistos en la Sección 6.4 del Capítulo 6 dejan claro el correcto funcionamiento de la función generadora de matrices de relieves y rugosidades. Sin embargo, el uso de esta función está muy limitado a ciertos modelos, ya que no todos los modelos hacen uso del parámetro de longitud de rugosidad (z_0) y ciertos modelos necesitan desniveles muy grandes para mostrar cambios en sus predicciones de energía generada con respecto a la predicha para un parque eólico de tipología *offshore* completamente plano.

7.2. Consecución de objetivos

En las Secciones 3.1 y 3.2 del Capítulo 3 se establecieron los objetivos generales y específicos de este TFM, cuyo cumplimiento se debate y argumenta a continuación.

En lo referente a los objetivos específicos, el OE1 fue abordado y discutido en la Sección 2.1 del Capítulo 2, donde se realizó un protocolo SLR para responder a una serie de RQ planteadas de cara a esbozar correctamente el estado actual de las investigaciones y publicaciones relacionadas con el algoritmo CRO y su evolución a lo largo de los años.

Con la inclusión del modelo de estela de Ainslie se ha cumplido con el OE2, añadiendo un modelo adicional y haciendo que el repositorio de modelos de estela elaborado con el que se cuenta actualmente, tras la elaboración del TFG y TFM, sea de 9 modelos (Jensen, Park2, Frandsen, Larsen, Ishihara, Zhang, Bastankhah simplificado, Bastankhah y Ainslie).

Según lo establecido en el OE6, con el fin de evaluar el grado de mejora conseguido tras la incorporación de 3 nuevos sustratos al flujo de optimización del CRO-SL original, se ha

comparado la nueva configuración formada por un total de 7 sustratos con la configuración previa de este algoritmo formada únicamente por 4 sustratos. También, a fin de verificar el correcto funcionamiento del operador 1Px para este tipo de problemas y tener una referencia del grado de optimización alcanzado por otro algoritmo evolutivo distinto ante el mismo problema, se ha realizado otra comparativa con los resultados obtenidos por el BGA.

Tras el análisis de las soluciones obtenidas haciendo uso de los 3 nuevos sustratos implantados llevadas a cabo en la Sección 6.1 del Capítulo 6, se puede concluir que el algoritmo CRO-SL ha sufrido una mejora muy notoria en su funcionamiento al implementar los 3 nuevos sustratos, mostrando incrementos medios de rendimiento medio entre todos los modelos de estela de un 0.68% con respecto al BGA y un 1.5% respecto a la configuración inicial del CRO-SL que contaba con sólo 4 sustratos. Estos resultados demuestran que la adición de los nuevos sustratos ha resultado satisfactoria consiguiendo completar el OE3.

En el OE4 se buscaba elaborar un análisis de la repercusión de cada uno de los sustratos sobre las soluciones obtenidas, hecho que se realizó en la Sección 6.2 del Capítulo 6. Se observó un alto grado de participación de 2 de los sustratos (1Px y PSO) mientras que el sustrato correspondiente al WOA tuvo una labor más residual al solo tener participación en generaciones muy prematuras.

Finalmente, de acuerdo con lo establecido en el OE5, se ha elaborado un *script* que contiene la definición de una función capaz de generar, a modo de *output*, dos matrices que contienen la información relativa a los cambios de altitud y las distintas rugosidades superficiales en función de las coordenadas del terreno escogidas para llevar a cabo el cálculo del déficit de velocidades. En la Sección 5.5 del Capítulo 5 se describe el desarrollo y el funcionamiento de la función, mientras que en la Sección 6.4 del Capítulo 6 se evalúa el funcionamiento de dicha aplicación, la cual obtuvo los resultados esperados.

7.3. Impacto social, económico y medioambiental del estudio

En términos económicos, la producción de una mayor cantidad de energía utilizando un menor número de aerogeneradores reduce significativamente los costos de transporte de componentes, instalación y mantenimiento de los dispositivos. Además, al necesitar menos turbinas eólicas para generar electricidad en una zona específica, se pueden destinar recursos

adicionales a la diversificación de inversiones en otros tipos de energías renovables, como la energía solar fotovoltaica o hidroeléctrica, o a la construcción de nuevos parques eólicos en otros territorios.

En cuanto al beneficio medioambiental, aunque la energía eólica tiene un régimen de funcionamiento variable, aprovechar al máximo los períodos de operación ayuda a reducir la dependencia de fuentes de energía más contaminantes, como el carbón o el petróleo, lo que mejora la calidad del aire de las zonas residenciales. Sin embargo, es importante realizar un estudio previo de las zonas de instalación para minimizar el impacto negativo en el hábitat de las especies locales, especialmente las aves migratorias, que pueden verse afectadas por las turbinas eólicas (Farfán Aguilar, 2009).

Las energías renovables, en particular la energía eólica, son fundamentales para lograr una transición hacia un futuro más sostenible. Como se mencionó en la Sección 1.1 del Capítulo 1, grandes organizaciones como la ONU y la Unión Europea han establecido objetivos, como la Coalición Global por la Neutralidad del Carbono y el Horizonte Europa, para reducir las emisiones de gases de efecto invernadero. Los Objetivos de Desarrollo Sostenible (ODS), adoptados en 2015 por la ONU como parte de la Agenda 2030, son un conjunto de 17 objetivos globales que abordan los desafíos más urgentes de nuestro planeta, incluyendo la pobreza, la protección del medio ambiente, la promoción de la igualdad y la justicia social. Este estudio en particular contribuye a varios de estos objetivos y metas, siendo el ODS 7 el más relevante, ya que se alinea perfectamente con la temática de este trabajo en la mayoría de sus metas específicas, aunque también tiene incidencia sobre el ODS 8 y ODS 12. En la Tabla 7.1 se desglosan los ODS en las distintas metas específicas vinculadas a la labor de investigación e implementaciones realizadas en este TFM.

Tabla 7.1. ODS y metas específicas vinculadas al estudio realizado.

OBJETIVO	META ESPECÍFICA	DESCRIPCIÓN
ODS 7: Garantizar el acceso a una energía asequible, segura, sostenible y moderna.	ODS 7.1	De aquí a 2030, garantizar el acceso universal a servicios energéticos asequibles, fiables y modernos.
	ODS 7.2	De aquí a 2030, aumentar considerablemente la proporción de energía renovable en el conjunto de fuentes energéticas.
	ODS 7.3	De aquí a 2030, duplicar la tasa mundial de mejora de la eficiencia energética.
	ODS 7.b	De aquí a 2030, ampliar la infraestructura y mejorar la tecnología para prestar servicios energéticos modernos y sostenibles para todos los países en desarrollo.
ODS 8: Promover el crecimiento económico inclusivo y sostenible, el empleo y el trabajo decente para todos.	ODS 8.4	Mejorar progresivamente, de aquí a 2030, la producción y consumo eficientes de los recursos mundiales y procurar desvincular el crecimiento económico de la degradación del medio ambiente, conforme al Marco Decenal de Programas sobre modalidades de Consumo y Producción Sostenibles, empezando por los países desarrollados.
ODS 12: Garantizar modalidades de consumo y producción sostenibles.	ODS 12.2	De aquí a 2030, lograr la gestión sostenible y el uso eficiente de los recursos naturales.

7.4. Líneas de trabajo futuro

En futuros trabajos relacionados con esta temática, sería de interés estudiar la búsqueda de distribuciones óptimas de turbinas añadiendo en los cálculos las pérdidas sufridas por el ángulo de resbalamiento que ven las palas del rotor cuando cambia la dirección el viento. Eliminar la hipótesis de ángulo de resbalamiento nulo en los cálculos aumentaría la precisión de los cálculos y permitiría obtener magnitudes de energía generada más cercanas a las reales. El vínculo automatizado de algoritmos metaheurísticos que utilicen resultados obtenidos con tecnologías que permiten el análisis de volúmenes finitos a través de CFD, como OpenFOAM o ANSYS Fluent, podría suponer un gran avance en la precisión de resultados y correlación con los valores reales mostrados experimentalmente en los túneles de viento.

También se podría realizar un estudio de las distintas soluciones obtenidas, tanto en este TFM como las conseguidas en (Romero Barrera, 2023), con el fin de obtener una serie de

parámetros que permitan el diseño de nuevos algoritmos o sustratos para el CRO-SL, o restricciones que permitan reducir el espacio de búsqueda. Esto permitiría reducir los tiempos de ejecución de los algoritmos y permitir que el usuario obtenga soluciones de manera más rápida, sin perder calidad en sus resultados.

La utilización de turbinas eólicas en el ámbito de la defensa también es una temática bastante estudiada. Se podría realizar estudios buscando el efecto contrario a la hora de escoger su colocación, es decir, distribuirlas de tal forma que disminuyan la energía del viento de forma abrupta y rápida con el fin de parar ciclones o tornados en zonas con mucha frecuencia de aparición de éstos.

Las turbinas eólicas, al igual que cualquier otro dispositivo de alta complejidad, puede sufrir averías y estar períodos de tiempo relativamente largos sin operar. Este hecho puede suponer que la eficiencia de la distribución de turbinas en comparación con otras colocaciones se vea disminuida. Un estudio interesante que realizar a futuro es el grado de afectación de la parada de funcionamiento de ciertas turbinas eólicas a la eficiencia de la granja en comparación con otras soluciones, que, a pleno funcionamiento de aerogeneradores, mostrasen una producción menor.

Otra línea que sería de interés llevar a cabo, es la modelización matemática de las estelas generadas por las turbinas eólicas en órbita marciana, lo cual permitiría el uso de este nuevo modelo a la hora de optimizar la colocación de aerogeneradores sobre el suelo de Marte para la generación de energía por los vientos solares y vientos propios de la atmósfera del planeta. Esto podría ayudar a la generación de energía en dicho planeta y su uso para una posible colonización o la construcción de pequeñas bases de investigación en dicho planeta.

Referencias bibliográficas

- Ainslie, J. F. (1985). Development of an eddy viscosity model for wind turbine wakes. *7th BWEA Wind Energy Conference, Oxford, UK*, (pp. 61–66).
- Ainslie, J. F. (1986). Wake modelling and the prediction of turbulence properties. *Proceedings of the 8th British wind energy association conference, Cambridge*, (pp. 19–21).
- Ainslie, J. F. (1988). Calculating the flowfield in the wake of wind turbines. *Journal of Wind Engineering and Industrial Aerodynamics*, 27, 213-224. doi:[https://doi.org/10.1016/0167-6105\(88\)90037-2](https://doi.org/10.1016/0167-6105(88)90037-2)
- Avia, F. (1992). *Curva de potencia del aerogenerador (AWEC-60)*. Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (Ciemat).
- Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection*. London: Murray.
- del Río, P., Resch, G., Ortner, A., Liebmann, L., Busch, S., & Panzer, C. (2017). A techno-economic analysis of EU renewable electricity policy pathways in 2030. *Energy Policy*, 104, 484-493. doi:<https://doi.org/10.1016/j.enpol.2017.01.028>
- Doran, G. T., & others. (1981). There's a SMART way to write management's goals and objectives. *Management review*, 70, 35–36.
- Engineering Sciences Data Unit. (1974). *Characteristics of atmospheric turbulence near the ground, Part II, EDU Datasheet 74031*. London: Great Britain Engineering Sciences Data Unit.
- Farfán Aguilar, M. Á., Vargas, J., Duarte, J., & Real, R. (2009, December). What is the impact of wind farms on birds? A case study in southern Spain. *Biodiversity and Conservation*, 18, 3743-3758. doi:10.1007/s10531-009-9677-4
- Gaona, G., Balcázar, L., & Jaime, I. (2015, April). Estimación de longitud de rugosidad aerodinámica a partir de imágenes MODIS y LandSat.
- Gómez, O., Ibáñez, O., Valsecchi, A., Bermejo, E., Molina, D., & Cordón, O. (2020). Performance analysis of real-coded evolutionary algorithms under a computationally expensive optimization scenario: 3D–2D Comparative Radiography. *Applied Soft Computing*, 97, 106793. doi:<https://doi.org/10.1016/j.asoc.2020.106793>

- Hau, E. (2006, January). *Wind Turbines: Fundamentals, Technologies, Application, Economics*. doi:10.1007/3-540-29284-5
- IRENA. (2023). *Renewable capacity statistics 2023*. Abu Dhabi.
- ISO. (1975). *ISO 2533:1975 Standard Atmosphere*. International Organization for Standardization, Geneva, Suiza. Retrieved from <https://www.iso.org/standard/7472.html>
- Jensen, N. O. (1983). *A note on wind generator interaction* (Vol. 2411). Citeseer.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks, 4*, pp. 1942-1948 vol.4. doi:10.1109/ICNN.1995.488968
- Kumar, A. (2000, December). Estimating the flammable mass of a vapor cloud. by John L. Woodward Center for Chemical Process Safety (CCPS) American Institute of Chemical Engineers, New York, NY (1999) 292 pages Pub G-60, ISBN 0-8169-0778-1. *Environmental Progress - ENVIRON PROG*, 19. doi:10.1002/ep.670190405
- Larsen, G. C. (1988). *A Simple Wake Calculation Procedure*. Risø National Laboratory.
- Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, 95, 51-67. doi:<https://doi.org/10.1016/j.advengsoft.2016.01.008>
- Naciones Unidas. (n.d.). <https://www.un.org/es/climatechange/net-zero-coalition>. Retrieved 06 24, 2023, from Organización de las Naciones Unidas: <https://www.un.org/es/climatechange/net-zero-coalition>
- Orús, A. (2023, March). Energía Eólica: Generación por CC.AA en España en 2022. *Energía Eólica: Generación por CC.AA en España en 2022*. Retrieved from <https://es.statista.com/estadisticas/1004140/generacion-eolica-por-region-en-espana/>
- Pérez-Aracil, J., Casillas-Pérez, D., Jiménez-Fernández, S., Prieto-Godino, L., & Salcedo-Sanz, S. (2022). A versatile multi-method ensemble for wind farm layout optimization. *Journal of Wind Engineering and Industrial Aerodynamics*, 225, 104991. doi:<https://doi.org/10.1016/j.jweia.2022.104991>

- Pierik, J. T., Dekker, J. W., Braam, H., Bulder, B. H., Winkelaar, D., Larsen, G. C., . . . Molly, J. P. (1999). European wind turbine standards II (EWTS-II). In E. L. Petersen, P. Hjulær Jensen, K. Rave, P. Helm, & H. Ehmann (Ed.), *Wind energy for the next millennium. Proceedings* (pp. 568–571). James and James Science Publishers.
- Rathmann, O. S., Hansen, B. O., Hansen, K. S., Mortensen, N. G., & Murcia Leon, J. P. (2018). *The Park2 Wake Model - Documentation and Validation*.
- Rehman, S., Khan, S., & Alhems, L. (2020, January). The effect of acceleration coefficients in Particle Swarm Optimization algorithm with application to wind farm layout design. *FME Transactions*, 48, 922-930. doi:10.5937/fme2004922R
- Rehman, S., Khan, S., & Alhems, L. (2020, January). The effect of acceleration coefficients in Particle Swarm Optimization algorithm with application to wind farm layout design. *FME Transactions*, 48, 922-930. doi:10.5937/fme2004922R
- Romero Barrera, A. J. (22 de Mayo de 2023). *DISEÑO DE ALGORITMOS METAHEURÍSTICOS PARA EL PROBLEMA DE COLOCACIÓN DE TURBINAS EÓLICAS SOBRE EL TERRENO*. Trabajo Fin de Grado, Universidad Rey Juan Carlos, Escuela de Ingeniería de Fuenlabrada, Fuenlabrada. Obtenido de <https://burjcdigital.urjc.es/handle/10115/22188>
- Saavedra-Moreno, B., Salcedo-Sanz, S., Paniagua-Tineo, A., Gascón-Moreno, J., & Portilla-Figueras, J. A. (2011). Optimal Evolutionary Wind Turbine Placement in Wind Farms Considering New Models of Shape, Orography and Wind Speed Simulation. In J. Cabestany, I. Rojas, & G. Joya (Ed.), *Advances in Computational Intelligence* (pp. 25–32). Berlin: Springer Berlin Heidelberg.
- Salcedo-Sanz, S. (2016, October). A review on the coral reefs optimization algorithm: New development lines and current applications. *Progress in Artificial Intelligence*, 6. doi:10.1007/s13748-016-0104-2
- Salcedo-Sanz, S., Del Ser, J., Landa-Torres, I., Gil-Lopez, S., & Portilla-Figueras, A. (2014, July). The Coral Reefs Optimization Algorithm: A Novel Metaheuristic for Efficiently Solving Optimization Problems. *TheScientificWorldJournal*, 2014, 739768. doi:10.1155/2014/739768

- Salcedo-Sanz, S., Gallo-Marazuela, D., Pastor-Sánchez, A., Carro-Calvo, L., Portilla-Figueras, A., & Prieto, L. (2014). Offshore wind farm design with the Coral Reefs Optimization algorithm. *Renewable Energy*, 63, 109-115. doi:<https://doi.org/10.1016/j.renene.2013.09.004>
- Salcedo-Sanz, S., Pastor-Sánchez, Á., Gallo-Marazuela, D., & Portilla-Figueras, A. (2013, October). A Novel Coral Reefs Optimization Algorithm for Multi-objective Problems., (pp. 326-333). doi:10.1007/978-3-642-41278-3_40
- Salcedo-Sanz, S., Pastor-Sánchez, A., Ser, J. D., Prieto, L., & Geem, Z. W. (2015). A Coral Reefs Optimization algorithm with Harmony Search operators for accurate wind speed prediction. *Renewable Energy*, 75, 93-101. doi:<https://doi.org/10.1016/j.renene.2014.09.027>
- Statista Search Department. (2022). *Número de aerogeneradores instalados en España de 2011 a 2021*. Obtenido de <https://es.statista.com/estadisticas/1049622/numero-de-aerogeneradores-instalados-en-espana/#:~:text=Esta%20estad%C3%ADstica%20muestra%20la%20evoluci%C3%B3n,total%20de%2021.574%20aerogeneradores%20instalados.>
- Vermeij, M., Salcedo-Sanz, S., & Bulnes, J. (2017, January). New Coral Reefs-based Approaches for the Model Type Selection Problem: A Novel Method to Predict a Nation's Future Energy Demand. *International Journal of Bio-Inspired Computation*, 10, 1. doi:10.1504/IJBIC.2017.10004324
- Yan, C., Ma, J., Luo, H., & Patel, A. (2019). Hybrid binary Coral Reefs Optimization algorithm with Simulated Annealing for Feature Selection in high-dimensional biomedical datasets. *Chemometrics and Intelligent Laboratory Systems*, 184, 102-111. doi:<https://doi.org/10.1016/j.chemolab.2018.11.010>
- Zergane, S., Smaïli, A., & Masson, C. (2018, February). Optimization of Wind Turbine Placement in a Wind Farm using a New Pseudo-Random Number Generation Method. *Renewable Energy*, 125. doi:10.1016/j.renene.2018.02.082
- Zhang, D., Liang, Y., Li, C., Xiao, Y., & Hu, G. (2022, October). Applicability of Wake Models to Predictions of Turbine-Induced Velocity Deficit and Wind Farm Power Generation. *Energies*, 15, 7431. doi:10.3390/en15197431

Anexo I. Glosario de acrónimos

- 1Px: *1 Point Crossover*.
- 2Px: *2 Points Crossover*.
- AVDRSS: *Axial Velocity Déficit Rankine Source Surface*.
- BGA: *Basic Genetic Algorithm*.
- CFD: *Computer Fluid Dynamics*.
- CRO: *Coral Reef Optimization*.
- CRO-HS: *Coral Reef Optimization with Harmony Search*.
- CRO-SL: *Coral Reef Optimization with Substrate Layers*.
- EC: *Exclusion Criteria*.
- GM: *Gaussian Mutation*.
- HAWT: *Horizontal Axis Wind Turbine*.
- IC: *Inclusion Criteria*.
- ISA: *International Standard Atmosphere*.
- ML: *Machine Learning*.
- MO-CRO: *Multi Objective – Coral Reef Optimization*.
- MPx: *Multipoint Crossover*.
- NASA: *National Aeronautics and Space Administration*.
- NERF: *Netherlands European Research Foundation*.
- NREL: *National Renewable Energy Laboratory*.
- ODS: *Objetivos de Desarrollo Sostenible*.
- OE: *Objetivo Específico*.
- OG: *Objetivo General*.
- PSO: *Particle Swarm Optimization*.

- QC: *Quality Criteria*.
- RANS: *Reynolds average Navier-Stokes*.
- RN: Redes Neuronales.
- RQ: *Research Questions*.
- SLR: *Systematic Literature Review*.
- TFG: Trabajo Fin de Grado.
- TFM: Trabajo Fin de Máster.
- TSR: *Tip Speed Ratio*.
- UAH: Universidad de Alcalá.
- URJC: Universidad Rey Juan Carlos.
- VAWT: *Vertical Axis Wind Turbine*.
- WOA: *Whale Optimization Algorithm*.

Anexo II. Glosario de variables

- β : ángulo de resbalamiento de la corriente incidente sobre la pala del rotor.
- ω : factor de inercia (algoritmo PSO).
- ω_r : velocidad angular de rotación del rotor.
- b : anchura adimensional de la estela.
- c_{pt} : coeficiente de potencia del aerogenerador.
- c_T : coeficiente de tracción del rotor.
- D_r : diámetro del rotor.
- D_w : diámetros de la estela.
- E_{gen} : energía generada por una distribución de aerogeneradores.
- $E_{m\acute{a}x}$: energía máxima generada por 30 aerogeneradores sin afectación de estelas.
- $E_{m\acute{i}n}$: energía mínima generada por 30 aerogeneradores con solapamiento total de estelas.
- I_0 : intensidad de turbulencia ambiental.
- I_a : intensidad de turbulencia añadida.
- I_w : intensidad de turbulencia interior de la estela.
- $R_{9,6}$: radio de la estela, descrita por el modelo de Larsen, a una distancia aguas abajo de 9.6 veces el diámetro del rotor del aerogenerador que la produce.
- $Ren(\%)$: rendimiento de una solución de carácter porcentual.
- $\{S\}$: conjunto de soluciones optimizadas de los algoritmos metaheurísticos.
- U_∞ : velocidad incidente del aire sin perturbar.
- \hat{U}_M : velocidad adimensional en el eje axial de la estela a una distancia determinada.
- U_x : velocidad incidente del aire en unas coordenadas concretas, transversal al plano del rotor.
- $\{WM\}$: conjunto de modelos de estela.
- z : altura de la turbina.
- Z : matriz de relieve del terreno.
- z_0 : longitud de rugosidad superficial del terreno.
- Z_0 : matriz de longitudes de rugosidad superficial del terreno.

Anexo III. Listados de código

El código perteneciente al TFG (Romero Barrera, 2023) son pertenencia del autor, y por extensión, de los tutores y la institución donde se elaboró el trabajo. Los programas completos derivados del mismo y que fueron utilizados para el análisis contienen *scripts* de dicho trabajo, y, por tanto, no se permite su difusión ni uso hasta que el autor haga público el repositorio: https://github.com/antoniojromero/tfm_antoniojromero

En caso de requerir la consulta de los *scripts* de MATLAB completos para evaluar algo específico de la memoria, de forma justificada, se deja una lista de correos electrónicos a los que contactar para requerir una copia:

- Autor: Antonio J. Romero Barrera (email: antonio.romero@urjc.es).
- Tutora: Laura M^a. Cornejo Bueno (email: laura.cornejo@uah.es).
- Cotutor: David Casillas Pérez (email: david.casillas@urjc.es).

A continuación, se exponen una serie de scripts que contienen las funciones y fragmentos de código incorporados en este TFM, como las funciones de los 3 nuevos sustratos incorporados al CRO-SL, el fragmento que controla el *tuning* de hiperparámetros del algoritmo PSO, la función que genera la matriz de desniveles del terreno y la función del nuevo modelo de estela incorporado, así como el fragmento correspondiente a la generación de datos en la estructura 'data' correspondientes este modelo a través de la función 'genData()'. El uso o modificación de estos *scripts* quedan restringidos hasta la apertura al público del repositorio previamente mencionado.

Script 1. *Modelo de estela de Ainslie.*

```
function UX = wm_ainslie(m, vMod, xUDef, yUDef, data)

    Ia2 = 0.29*(2).^(-1/3).*sqrt( 1-sqrt(1-data.CT) ); % Intensidad de turbulencia en la
estela
    Iw = sqrt(data.I0^2 + Ia2.^2);
    Def2 = data.CT - 0.05 - (16*data.CT - 0.5)*data.I0/1000;
    D_w =sqrt( 3.56*data.CT/(8*Def2*(1-0.5*Def2)) )*data.D_r;
    % Se adimensionaliza la distancia radial con el diametro de estela a 2
    % diametros de distancia del rotor
    def2 = Def2/sqrt((1 + 7.12*(data.sigma_theta*(xUDef(xUDef(:,m)>0,m)/D_w).^2)));
    Udef = def2.'.*exp(-3.56*(yUDef(xUDef(:,m)>0,m)/D_w).^2); % valor adimensional
    Udef = sum(Udef);
    ux = vMod*(1 - Udef); % dimensionalizacion de la velocidad (m/s)

end
```

Script 2. *Sustrato 1Px con mutación para algoritmo CRO-SL.*

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SISTRATO 1Px MUTATIVO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Realizado por Antonio J. Romero %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function p = onePxMut_integer(p, f, cn, vvec, data)

% BROADCAST SPAWNING
pBroad = rand(1, 1); % probabilidad de BroadSpawning
if pBroad <= cn.BroadSpaw
    % Etapa de Cruce: 1Px
    mutationIndProb = 0.15; % Probabilidad de mutacion del individuo
    mutationGenProb = 0.3; % Probabilidad de mutacion del gen
    onePxSelectParam= 0.5; % Parametro de seleccion de candidatos

    larvae1 = p.coral_filled(cn.ind_indx,:);
    % Seleccion de 'nCandidates' para el cruce
    nCandidates = round(size(p.coral_filled,1)*0.3);

    candidatesFit = f.coral(f.coral > 0);
    candidatesInd = randperm(size(f.coral(f.coral > 0),1), nCandidates);
    candidatesFit = candidatesFit(candidatesInd);
    [~, sortedIndexes] = sort(candidatesFit,1,'descend');
    selectProbMatrix = onePxSelectParam*((1-onePxSelectParam).^((0:(nCandidates-2))'));
    r = rand;
    larvae2 = candidatesInd(sortedIndexes(r > selectProbMatrix));

    if isempty(larvae2)
        larvae2 = p.coral_filled(candidatesInd(end),:);
    else
        larvae2 = p.coral_filled(candidatesInd(1),:);
    end
    larvae1 = sort(larvae1, 'ascend');
```

```

larvae2 = sort(larvae2, 'ascend');

count = 0; % contador
done = false; % criterio de parada
while(done == false)
    crossoverPoint = randi(cn.n_v-1);
    if (larvae1(crossoverPoint) < larvae2(crossoverPoint+1)) &
(larvae2(crossoverPoint) < larvae1(crossoverPoint+1))
        newLarvae(1, :) = [larvae1(1:crossoverPoint) larvae2(crossoverPoint+1:end)];
        newLarvae(2, :) = [larvae2(1:crossoverPoint) larvae1(crossoverPoint+1:end)];
        done = true;
    end
    count = count + 1;
    if count > cn.n_v
        done = true;
        newLarvae(1, :) = larvae1;
        newLarvae(2, :) = larvae2;
    end
end
% Etapa de Mutación
for index=1:size(newLarvae,1)
    x = setdiff(1:data.Kgr^2, newLarvae(index,:));
    % Index for Mutations
    indexesPop = rand(1, size(newLarvae,1)) < mutationIndProb;
    indexesInd = rand(1, size(newLarvae,2)) < mutationGenProb;
    if indexesPop(index) == 1
        for k=1:size(indexesInd,2)
            if indexesInd(k) == 1
                int_rand = randi(numel(x),1);
                prev_genvalue = newLarvae(index, k);

                newLarvae(index, k) = x(int_rand) ; % Se intercambia el gen por el
valor aleatorio

                x(int_rand) = prev_genvalue; % Se libera el valor del gen para el
vector aleatorio
            end
        end
    end
    newLarvae(index, :) = sort(newLarvae(index, :));
end
% Etapa de Tournament: Problema de maximizacion
individuals = non_repeated(newLarvae, data);
fit = EvaluatePopulation(individuals, vvec, data);

if fit(1) < fit(2) % Tournament
    newValues = newLarvae(2, :);
else
    newValues = newLarvae(1, :);
end
% BROODING
else % operador de brooding
    newValues = brooding_integer(p.coral_filled(cn.ind_indx, :), cn);
end
newValues = checkLimits(newValues, cn);
p.larvae(cn.ind_indx, :) = newValues;
end

```

Script 3. *Sustrato PSO para algoritmo CRO-SL.*

```

function p = PSO_integer(p, cn, vvec, data)

% Inicialización de parametros PSO
% tuning de hiperparámetros realizado para 30x30 con 30 turbinas
data.t = 0;
data.w = 1.0; % Coeficiente de inercia
data.c1 = 1.3; % Coeficiente de aceleracion personal
data.c2 = 1.0; % Coeficiente de aceleracion social
data.v0 = false;

% POSIBILIDAD DE PERMUTACION
r = rand();
% 10% de probabilidad de permutacion total
if (r < 0.1) || sum((sort(p.coral_filled(cn.ind_indx,:)) == sort(p.best(1,:)))) == 0%
Permutamos las particulas aleatoriamente d elas poblaciones (mayor exploracion)
    n = length(p.coral_filled(cn.ind_indx,:));
    perm_ind = randperm(n); % indices permutados
    p.coral_filled(cn.ind_indx,:) = p.coral_filled(cn.ind_indx,perm_ind);
end
newValues = zeros(1,cn.n_v);
v = zeros(1, cn.n_v);
particle_group = p.coral_filled(cn.ind_indx, :); % grupo de partículas que se van a
desplazar
xBest = p.xbest_filled(cn.ind_indx,:);
gBest = p.best(1,:); % El mejor individuo en esta iteracion
pBroad = rand(1, 1); % Probabilidad de BroadSpawning

% BROADCAST SPAWNING
if pBroad <= cn.BroadSpaw % Proceso de BroadSpawning
    for k = 1:cn.n_v

        x = zeros(1,2);
        gBestCoord = zeros(1,2);
        gr = zeros(1,data.Kgr^2);
        gr(1, gBest(k)) = 1; % se cambian las posiciones de los individuos a 2
        gr = reshape(gr, data.Kgr, data.Kgr); % Se cambia de posicion de vector a matriz
        [gBestCoord(1), gBestCoord(2)] = find(gr); % se obtienen las coordenadas de gBest

        %if (bestIndividualIndex ~= i)
        % Reshape de las mejores posiciones de las turbinas en el individuo
        xBestCoord = zeros(1,2);
        gr = zeros(1, data.Kgr^2); % size 1x400
        gr(1, xBest(k)) = 1; % se cambian las posiciones de los individuos a 1
        gr = reshape(gr, data.Kgr, data.Kgr); % Se cambia de posicion de vector a matriz
        [xBestCoord(1),xBestCoord(2)] = find(gr); % se obtienen las coordenadas de la
particula
        % Reshape de las posiciones de las turbinas en el individuo

```

```

gr = zeros(1, data.Kgr^2); % size 1x400
gr(1, particle_group(1,k)) = 1; % se cambian las posiciones de los individuos a 1
gr = reshape(gr, data.Kgr, data.Kgr); % Se cambia de posicion de vector a matriz
[x(1),x(2)] = find(gr); % se obtienen las coordenadas de la partícula

% ECUACION DEL MOVIMIENTO
% velocidad
v_candidate = (data.w.*v(1,k).*x./norm(x))...
+ rand*data.c1*(xBestCoord(1,:) - x)...
+ rand*data.c2*(gBestCoord(1,:) - x);
v(1,k) = norm(v_candidate);
% posicion
x_candidate = x + round(v_candidate); % Nueva posicion a valorar
% restriccion de limites
if(x_candidate(1) <= 0) | (x_candidate(2) <= 0)
    x_candidate(find(x_candidate <= 0)) = 1;
end
if(x_candidate(1) > data.Kgr) | (x_candidate(2) > data.Kgr)
    x_candidate(find(x_candidate > data.Kgr)) = data.Kgr;
end

ground = zeros(data.Kgr); % Generamos el grid de posiciones
ground(x_candidate(1),x_candidate(2)) = 1; % sustituimos en el vector de
posiciones el numero 1, el resto son 0
newValues(1,k) = find(ground); % encontramos el valor no nulo del vector de
posiciones
end

% Comprobar que se situa en posicion correcta %%%%%%%%%%%
[unique_values, unique_index] = unique(particle_group(1,:));
repeated_index = setdiff(1:size(particle_group,2), unique_index);
overlimit = find(particle_group(1,:)>data.Kgr^2);
underlimit = find(particle_group(1,:)<1);
if (sum(overlimit) > 0) || (sum(underlimit) > 0) || (sum(repeated_index) > 0)
    exit = 0;
    while(exit<1)
        if(sum(overlimit) == 0) && (sum(underlimit) == 0) && (sum(repeated_index) ==
0)
            exit = 1;
        else
            if (sum(repeated_index) > 0)
                particle_group(1,repeated_index) = randi(data.Kgr^2);
            end

            if (sum(overlimit) > 0)
                particle_group(1,overlimit) = randi(data.Kgr^2);
            end
            if (sum(underlimit) > 0)
                r = 1 + (1+6)*rand(1,length(particle_group(1,underlimit))); %rango de
resta entre [1,6]
                particle_group(1,underlimit) = randi(data.Kgr^2);
            end
        end
    end
    [unique_values, unique_index] = unique(particle_group(1,:));
    repeated_index = setdiff(1:size(particle_group,2), unique_index);
    overlimit = find(particle_group(1,:)>data.Kgr^2);

```

```

        underlimit = find(particle_group(1,:) < 1);
    end
end
%BROODING
else % Se llama al operador brooding
    newValues = brooding_integer(p.coral_filled(cn.ind_indx, :), cn);
end
newValues = checkLimits(newValues, cn);
p.larvae(cn.ind_indx, :) = newValues;
end
end

```

Script 4. *Sustrato WOA para algoritmo CRO-SL.*

```
%%%%%%%%%%  
SISTRATO WOA MUTATIVO %%%%%%%%%%  
Realizado por Antonio J. Romero %%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function p = WOA_integer(p, cn, vvec, data)  
% POSIBILIDAD DE PERMUTACION  
    r = rand();  
    % 10% de probabilidad de permutacion total  
    if (r < 0.1) || sum((sort(p.coral_filled(cn.ind_indx,:))==sort(p.best(1,:))))==0  
Permutamos las particulas aleatoriamente d elas poblaciones (mayor exploracion)  
        n = length(p.coral_filled(cn.ind_indx,:));  
        perm_ind = randperm(n); % indices permutados  
        p.coral_filled(cn.ind_indx,:) = p.coral_filled(cn.ind_indx,perm_ind);  
    end  
    % Inicialización de parametros WOA  
    a_values = -1*linspace(-2,0, cn.n_rep); % parametro controlador del movimiento  
    a = a_values(cn.currentIter); % el parámetro de control de movimiento desciende    A = 2*rand(1,2)*a-a; % amplitud  
    C = 2*rand(1,2);  
    l = 2*rand()-1; % rango entre [-1,1]  
    b = 0.5; % parametro de forma de la elipse  
  
    % POSIBILIDAD DE PERMUTACION  
  
newValues = zeros(1,cn.n_v);  
v = zeros(1, cn.n_v);  
particle_group = p.coral_filled(cn.ind_indx, :); % grupo de partículas que se van axBest = p.xbest_filled(cn.ind_indx,:);  
gBest = p.best(1,:); % El mejor individuo en esta iteracion  
pBroad = rand(1, 1); % Probabilidad de BroadSpawning  
Xrand = p.coral_filled(randperm(size(p.coral_filled,1)),:);  
  
if pBroad <= cn.BroadSpaw % Proceso de BroadSpawning  
    %%%%%%%%%%  
    r = rand();
```

```

for k = 1:cn.n_v

    x = zeros(1,2);
    gBestCoord = zeros(1,2);
    gr = zeros(1,data.Kgr^2);
    gr(1, gBest(k)) = 1; % se cambian las posiciones de los individuos a 2
    gr = reshape(gr, data.Kgr, data.Kgr); % Se cambia de posicion de vector a matriz
    [gBestCoord(1), gBestCoord(2)] = find(gr); % se obtienen las coordenadas de gBest

    %if (bestIndividualIndex ~= i)
    % Reshape de las mejores posiciones de las turbinas en el individuo
    xBestCoord = zeros(1,2);
    gr = zeros(1, data.Kgr^2); % size 1x400
    gr(1, xBest(k)) = 1; % se cambian las posiciones de los individuos a 1
    gr = reshape(gr, data.Kgr, data.Kgr); % Se cambia de posicion de vector a matriz
    [xBestCoord(1),xBestCoord(2)] = find(gr); % se obtienen las coordenadas de la
particula

    % Reshape de posición de turbina random del grupo
    gr = zeros(1, data.Kgr^2);
    gr(1, Xrand(k)) = 1; % se cambian las posiciones de los individuos a 1
    gr = reshape(gr, data.Kgr, data.Kgr); % Se cambia de posicion de vector a matriz
    [xRandCoord(1),xRandCoord(2)] = find(gr);
    % Reshape de las posiciones de las turbinas en el individuo
    gr = zeros(1, data.Kgr^2); % size 1x400
    gr(1, particle_group(1,k)) = 1; % se cambian las posiciones de los individuos a 1
    gr = reshape(gr, data.Kgr, data.Kgr); % Se cambia de posicion de vector a matriz
    [x(1),x(2)] = find(gr); % se obtienen las coordenadas de la particula

    % ECUACION DEL MOVIMIENTO

    % posicion

    if r >= 0.5 % avance en espiral logaritmica
        D = abs(gBestCoord(1,:) - x); % D'
        x_candidate = round( D.*exp(b*1).*cos(2*pi*1) + gBestCoord(1,:) ); % Nueva
posicion a valorar
    else % avance lineal
        if abs(A) < 1
            D = abs(C.*gBestCoord(1,:) - x); % D
            x_candidate =round( gBestCoord(1,:) - A.*D );
        else

            D = abs(C.*gBestCoord(1,:) - x); % D
            x_candidate =round( xRandCoord(1,:) - A.*D );
        end
    end
    % restriccion de limites
    if(x_candidate(1) <= 0) | (x_candidate(2) <= 0);
        x_candidate(find(x_candidate <= 0)) = 1;
    end
    if(x_candidate(1) > data.Kgr) | (x_candidate(2) > data.Kgr);
        x_candidate(find(x_candidate > data.Kgr)) = data.Kgr;
    end

    ground = zeros(data.Kgr); % Generamos el grid de posiciones
    ground(x_candidate(1),x_candidate(2)) = 1; % sustituimos en el vector de

```

```

posiciones el numero 1, el resto son 0
    newValues(1,k) = find(ground); % encontramos el valor no nulo del vector de
posiciones
end

% Comprobar que se situa en posicion correcta %%%%%%%%%%%
[unique_values, unique_index] = unique(particle_group(1,:));
repeated_index = setdiff(1:size(particle_group,2), unique_index);
overlimit = find(particle_group(1,:) > data.Kgr^2);
underlimit = find(particle_group(1,:) < 1);
if (sum(overlimit) > 0) || (sum(underlimit) > 0) || (sum(repeated_index) > 0)
    exit = 0;
    while(exit < 1)
        if(sum(overlimit) == 0) && (sum(underlimit) == 0) && (sum(repeated_index) ==
0)
            exit = 1;
        else
            if (sum(repeated_index) > 0)
                particle_group(1,repeated_index) = randi(data.Kgr^2);
            end

            if (sum(overlimit) > 0)
                particle_group(1,overlimit) = randi(data.Kgr^2);
            end
            if (sum(underlimit) > 0)
                r = 1 + (1+6)*rand(1,length(particle_group(1,underlimit))); %rango de
resta entre [1,6]
                particle_group(1,underlimit) = randi(data.Kgr^2);
            end
        end
    end
end
%%%%%%%%%%
else % Se llama al operador brooding
    newValues = brooding_integer(p.coral_filled(cn.ind_indx, :), cn);
end
newValues = checkLimits(newValues, cn);
p.larvae(cn.ind_indx, :) = newValues;
end

```


Script 5. Función de generación de relieve y de longitud rugosidad variable.

```
% El siguiente código genera un spline de dominio [1,30]x[1,30] y recorrido
% entre [0,1], llamada sp_rel. La función puede ser llamada para evaluar
% cualquier punto del dominio. Este código guarda la spline en un .mat para
% que pueda ser cargada desde donde se desee.
% Posteriormente, se calcula el cambio de rugosidad superficial con
% respecto al cambio de altitud producido por los relieves generados en
% función del input Z0 (long. rugosidad a altitud media del terreno).

function [relMatrix, rugMatrix] = terrainRelief(Kgr, Npts, Z0)
    gr = zeros(Kgr); % Se genera matriz de ceros (desnivel nulo)
    gr(randperm(Kgr^2,Npts)) = 1;

    % Matriz de relieve del terreno
    nsz = Npts; % número de puntos
    h1 = fspecial('average',nsz); % filtro promediador
    h2 = fspecial('disk',nsz); % filtro promediador circular
    h3 = fspecial('gaussian',nsz,1); % filtro gaussiano rotacionalmente simétrico
    z = imfilter(gr,h1); % Se aplica filtro promediador h1

    a = 1/(max(z,[],"all") - min(z,[],"all"));
    b = - min(z,[],"all")*a;
    z = a*z+b; % matriz de altitudes

    tol = 1e1; % tolerancia
    x = 1:30; % array de coordenada x
    y = 1:30; % array de coordenada y
    [xx,yy] = meshgrid(x,y);
    sp_rel = spaps({x,y},z,tol);

    relMatrix = z; % Matriz final de relieve

    % Matriz de longitud de rugosidad del terreno

    z_med = sum(sum(relMatrix))/Kgr^2; % altitud media de los valores de la matriz de relieve

    % Ley de rugosidad (modificar en función del terreno)
    rugMatrix = Z0/2 * relMatrix./z_med; % matriz de rugosidad
    rugMatrix(isinf(rugMatrix)) = min(min(rugMatrix));

end
```


Anexo IV. Valores de rugosidad superficial

Tabla 7.2. *Valores de rugosidad para distintas tipologías de terreno. Información obtenida de (Kumar, 2000).*

z_0 (m)	Clase general del terreno	Terrenos específicos
$1.0 \cdot 10^{-4}$	Extensión agua/desierto.	Mar abierto calmado o suelo plano laminado con capa de nieve.
$1.0 \cdot 10^{-3}$	Extensión agua/desierto.	Áreas costeras.
$2.0 \cdot 10^{-3}$	Extensión agua/desierto.	Suelo de cultivo nevado.
$5.0 \cdot 10^{-3}$	Suelo nivelado cubierto de vegetación.	Césped (30 cm aproximadamente).
$1.0 \cdot 10^{-2}$	Suelo nivelado cubierto de vegetación.	Pocos árboles (en invierno).
$2.5 \cdot 10^{-2}$	Pistas de aeropuerto.	Árboles aislados.
$3.0 \cdot 10^{-2}$	Tierras de cultivo.	Casas rurales.
$5.0 \cdot 10^{-2}$	Tierras de cultivo.	Pocos árboles (en verano).
$8.0 \cdot 10^{-2}$	Tierras de cultivo.	Muchos setos.
0.2	Terreno rural.	Muchos setos, árboles, pequeños edificios...
0.4	Muchos niveles boscosos.	Límites de ciudades.
0.5	Terreno rural.	Centro de pequeños pueblos.
1.0	Bosques.	Centro de grandes pueblos, ciudades...
1.5 – 3.0	Con desniveles y montañosos.	Centro de ciudades con muchos edificios de gran altura.

Anexo V. Valores de referencia de σ_θ

Se recogen los valores de referencia correspondientes a las distintas longitudes de rugosidad del terreno, obtenidas de (Engineering Sciences Data Unit, 1974).

Tabla 7.3. *Valores de referencia de la fluctuación de la dirección del viento.*

Altitud respecto a la superficie (m)	σ_θ		
	$z_0 = 0.1$	$z_0 = 0.05$	$z_0 = 0.01$
10	0.16	0.14	0.11
20	0.14	0.12	0.10
40	0.12	0.11	0.09
60	0.11	0.10	0.08
100	0.10	0.09	0.07

Anexo VI. Tabla ponderativa entre MATLAB y Python

Esta tabla se utilizó en (Romero Barrera, 2023) para la selección del lenguaje de programación a utilizar en el desarrollo de los distintos *scripts* que conforman el código de los distintos algoritmos metaheurísticos.

Tabla 7.4. Comparativa de características entre MATLAB y Python (NumPy).

CARACTERÍSTICAS	MATLAB	PYTHON (NUMPY)
Rendimiento	Optimizado para cálculos numéricos y matriciales, lo que resulta en una mayor eficiencia para operaciones de matrices de gran tamaño.	NumPy también está optimizado para operaciones de matrices, pero puede ser menos eficiente que MATLAB en ciertos casos.
Velocidad de cálculo	MATLAB es generalmente más rápido que NumPy en operaciones de matrices de gran tamaño.	NumPy es menos eficiente que MATLAB en algunas operaciones de matrices, pero su velocidad de cálculo se puede mejorar con el uso de bibliotecas adicionales como Numba y Cython.
Funciones y paquetes integrados	Amplio conjunto de funciones integradas para operaciones matriciales, estadísticas y visualización de datos.	NumPy tiene una amplia variedad de funciones integradas para operaciones matriciales, pero puede tener menos funciones integradas para visualización de datos y estadísticas.

Diseño e implementación de sustratos en algoritmo CRO-SL para la distribución de aerogeneradores

Antonio José Romero Barrera

Universidad Internacional de la Rioja, Logroño (España)

Fecha: 13/07/2023

RESUMEN

Los algoritmos metaheurísticos se han erigido como una forma eficaz para abordar problemas complejos en distintos campos. En este trabajo se pretende encontrar la distribución de turbinas eólicas que maximice la energía generada anualmente dentro de un parque eólico. Haciendo uso del algoritmo poblacional *Coral Reef Optimization with Substates Layer* (CRO-SL) e incorporando 3 sustratos adicionales basados en el operador 1Px y los algoritmos *Particle Swarm Optimization* (PSO) y *Whale Optimization* (WOA), se consigue una mejora en la tendencia de crecimiento de las soluciones y la calidad de los resultados obtenidos al distribuir aerogeneradores sobre un terreno discretizado de proporciones 30x30 de tipología *offshore*. Se han utilizado un total de 3 modelos de estela (Jensen, Larsen y Ainslie) que permiten estimar el déficit de velocidad que afecta a los aerogeneradores y calcular la energía generada a través de una curva de potencia de un aerogenerador AWEC-60.

I. INTRODUCCIÓN

En un mundo en constante crecimiento, tanto a nivel tecnológico como a nivel poblacional, es esencial hacer un uso adecuado de las nuevas herramientas que surgen para mejorar la calidad de vida de los ciudadanos. Sin embargo, la acción humana ha tenido graves consecuencias en el planeta, como la contaminación, el calentamiento global, la deforestación, la superpoblación y la pérdida de biodiversidad. Para contrarrestar estos efectos negativos, instituciones como la Unión Europea (UE) y la Organización de Naciones Unidas (ONU) han implementado iniciativas como el Horizonte 2030, la Coalición por la Neutralidad del Carbono y los Objetivos de Desarrollo Sostenible (ODS).

Entre las energías renovables, la energía eólica ocupa un lugar destacado. En España, la energía eólica tiene una participación mayoritaria dentro del mercado de energías renovables, representando aproximadamente el 22%. El país cuenta con más de 21,500 turbinas eólicas, generando anualmente una energía superior a los 60 TWh [1]. Regiones como Castilla y León, Aragón y Galicia se destacan por ser los principales focos de producción de energía eólica en España [2].

Por lo tanto, es fundamental optimizar el funcionamiento de los aerogeneradores al determinar correctamente su posición en relación con otros. Esto permitirá minimizar los efectos negativos producidos por las estelas generadas por el giro de los rotores de las turbinas eólicas. Cuando el aire atraviesa el rotor de un aerogenerador, se produce una alteración del flujo que perturba la estabilidad y dirección del viento, generando una turbulencia que desacelera la corriente aguas abajo. Esta turbulencia se encuentra contenida en una estela de forma cónica, que delimita el efecto deficitario en ese volumen de control.

Para maximizar la eficiencia y producción de energía de los parques eólicos, es necesario llevar a cabo análisis y estudios exhaustivos sobre la ubicación óptima de las turbinas, considerando factores como la velocidad y dirección del viento,

el relieve del terreno y la interacción entre las estelas generadas por las turbinas. Estas evaluaciones permiten minimizar las interferencias entre turbinas y maximizar la generación de energía limpia y renovable.

II. ESTADO DEL ARTE

En este estudio, se realiza un análisis exhaustivo del algoritmo CRO (*Coral Reef Optimization*) [3] y sus variantes, con el objetivo de proporcionar una visión integral del estado de la investigación en este campo. Se abordan cuatro preguntas de investigación en las que se estudia: (RQ1) los ámbitos de aplicación del algoritmo CRO, (RQ2) las modificaciones surgidas del algoritmo base y (RQ3) la tendencia de crecimiento del número de publicaciones relacionadas con el CRO a lo largo de los años y (RQ4) la colaboración entre investigadores y las afiliaciones institucionales para comprender la red de investigación en torno al algoritmo CRO.

Los resultados revelan que el algoritmo CRO ha encontrado aplicaciones en diversos sectores, destacando en ciencias de la computación, telecomunicaciones, energía y medicina/ingeniería biomédica, tal y como se muestra en el gráfico circular desarrollado, expuesto la Figura 1. En el trabajo de investigación realizado Salcedo-Sanz [4], se realiza una revisión exhaustiva del algoritmo CRO, abordando sus aplicaciones en diversos sectores. Se analizan casos de uso en energía, como la colocación de aerogeneradores y la predicción de vientos, en telecomunicaciones, como la optimización de redes móviles [5], y en ciencias de la computación, como el *clustering* de conjuntos de datos [6].

En el ámbito energético, se destaca el artículo [7] que estudia el rendimiento del CRO en el diseño de parques eólicos *offshore*. Se comparan los resultados con otras metaheurísticas, como la

unir
LA UNIVERSIDAD
EN INTERNET

Universidad
de Alcalá

Universidad
Rey Juan Carlos

PALABRAS CLAVE

algoritmos, CRO-SL, granja eólica, metaheurísticas, optimización.

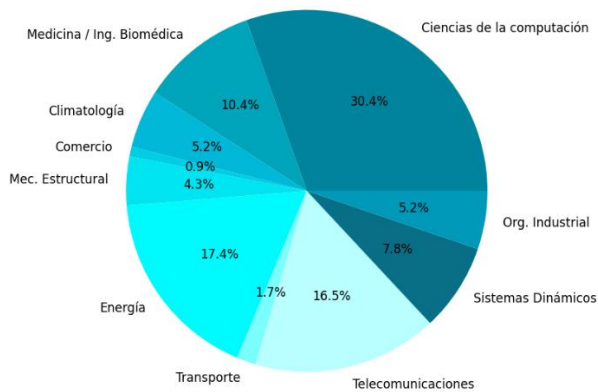


Fig. 1. Campos de investigación que hicieron uso del CRO en sus publicaciones.

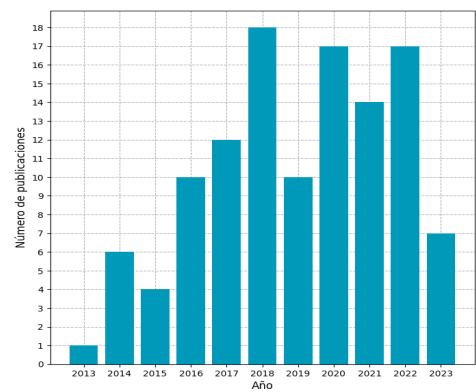


Fig. 2. Evolución del número de publicaciones por año relacionadas con el CRO desde 2013 hasta junio de 2023.

Evolución Diferencial (ED) y la Harmony Search (HS), demostrando que el algoritmo CRO proporciona soluciones más precisas y efectivas para la optimización en problemas relacionados con la energía eólica.

Además, se ha aplicado el algoritmo CRO en el análisis de factores climatológicos para energías renovables. En el estudio [8] se presenta el algoritmo híbrido CRO-HS, que combina el CRO con la metodología de la Harmony Search, mejorando su desempeño. Se evalúa su rendimiento en la predicción de la velocidad del viento utilizando datos de torres meteorológicas en Estados Unidos y España, obteniendo resultados superiores a las versiones básicas de los algoritmos CRO y HS.

Se han desarrollado variantes del algoritmo, como el ya mencionado CRO-HS (*Coral Reef Optimization with Harmony Search*), que combina metodologías de diferentes algoritmos metaheurísticos para mejorar su rendimiento en problemas específicos. También se ha explorado el uso del CRO en problemas de codificación variable y en la optimización multiobjetivo [9] o el CRO-SL [10] sobre el que se centra este trabajo, y que divide la población del coral en distintos grupos denominados sustratos sobre los que se aplica un operador determinado.

En el gráfico de evolución del número de publicaciones por año de la Figura 2, se observa una tendencia fluctuante en el número de publicaciones relacionadas con el CRO, con un pico máximo en el año 2018. Aunque el número de publicaciones es menor en comparación con otros algoritmos metaheurísticos establecidos, el algoritmo CRO ha logrado difundirse a nivel internacional, con la participación de investigadores de diferentes países.

En definitiva, se destaca la relevancia y el potencial del algoritmo CRO en la resolución de problemas de optimización en diversos campos. Las colaboraciones entre investigadores y las afiliaciones institucionales muestran la existencia de una red de investigación en crecimiento en torno al algoritmo CRO. El algoritmo CRO y sus variantes continúan siendo objeto de investigación y desarrollo, lo que abre nuevas oportunidades para la aplicación de esta técnica en futuros estudios y proyectos que centrados en nuevas temáticas.

III. OBJETIVOS Y METODOLOGÍA

El objetivo general de este TFM es la mejora del funcionamiento y ejecución del algoritmo CRO-SL, a través de la incorporación de nuevos sustratos, que hagan que se obtengan

mejores soluciones que el algoritmo CRO-SL inicial que cuenta con 4 sustratos activos por defecto, con sus correspondientes probabilidades de alojar individuos. Los sustratos implementados serán diseñados y ajustados por un proceso de *tuning* de hiperparámetros seleccionados, orientado a la optimización de posicionamiento de aerogeneradores sobre terrenos planos (parques eólicos *offshore*), y también desarrollando la funcionalidad de analizar las distribuciones de turbinas con distintas altitudes (parques eólicos *onshore*). Entre los objetivos específicos a cumplir para satisfacer el objetivo general, contamos con las siguientes metas:

1. El objetivo es analizar el impacto de las investigaciones científicas en el desarrollo del algoritmo CRO en los últimos años, identificando los problemas en los que se ha aplicado el CRO-SL y evaluando la calidad de las soluciones obtenidas.
2. Se busca ampliar la librería de modelos de estela existentes mediante la incorporación de un nuevo modelo que no se haya utilizado previamente. El objetivo es asegurar que este nuevo modelo pueda realizar predicciones adecuadas y funcione correctamente.
3. El objetivo es mejorar el rendimiento del algoritmo CRO-SL agregando nuevos sustratos basados en operadores no utilizados o algoritmos metaheurísticos. Estos nuevos sustratos buscan optimizar el posicionamiento de turbinas eólicas.
4. Se pretende analizar la influencia de cada sustrato en las soluciones obtenidas mediante el análisis con el CRO-SL. El objetivo es evaluar el comportamiento de los sustratos a lo largo de las generaciones y medir su grado de mejora en la optimización del posicionamiento de turbinas eólicas.
5. El objetivo es incorporar la posibilidad de generar terrenos con relieve y adaptar los modelos de estela para analizar las distribuciones con desniveles. Se busca desarrollar una función que permita una predicción tridimensional del déficit de velocidad causado por las estelas de las turbinas eólicas en entornos con terrenos irregulares.
6. El objetivo es realizar una comparativa entre el CRO-SL modificado con 7 sustratos, el CRO-SL inicial con 4 sustratos y el algoritmo BGA. El objetivo es evaluar el desempeño y la aptitud de cada algoritmo en la resolución del problema de distribución de aerogeneradores, considerando aspectos como la capacidad de encontrar soluciones óptimas, la eficiencia computacional y la estabilidad de los resultados.

Para llevar a cabo estos objetivos, se seleccionó ponderativamente la metodología y las herramientas para llevar a

cabo los análisis. Se consideraron opciones como el uso de herramientas basadas en dinámica de fluidos computacional (CFD) o los modelos de estela, los cuales resultaron ser la opción más eficiente.

Se realiza una comparativa de resultados registrando la energía media generada por la población y la energía máxima producida por el mejor individuo en cada generación. Esto permite evaluar la evolución y la velocidad de mejora de los algoritmos utilizados. También se analiza la disminución de velocidad generada por cada modelo de estela y su impacto en la distribución de aerogeneradores.

Además, se realiza un análisis comparativo de las soluciones optimizadas obtenidas utilizando los diferentes algoritmos metaheurísticos y los distintos modelos de estela. Se examina la ocupación de turbinas en el terreno seleccionado y se evalúa la efectividad de cada algoritmo en función de su capacidad para optimizar esta distribución.

IV. CONTRIBUCIÓN

Incorporación modelo de estela de Ainslie

El modelo de estela de Ainslie [11], se incorpora como el octavo modelo en este estudio, junto con otros siete modelos utilizados anteriormente. El modelo de Ainslie se implementa como una función en la estructura de datos existente y hace uso de la función 'genData()' para inicializarse. Para definir los límites de la estela de Ainslie, se utilizan los límites del modelo de Jensen, lo que permite una estimación casi nula del déficit cerca de la periferia del cono, validando así su uso combinado.

Para establecer la velocidad final que incide sobre el rotor de las turbinas eólicas afectadas por una estela turbulenta, se aplica la Ecuación (1), de forma que el déficit generado dependerá de la velocidad adimensional sobre el eje axial de la estela a una distancia 'x' determinada ($\hat{U}_M(x)$), la distancia radial desde la turbina afectada hasta dicho eje (r), el diámetro del rotor (D_r).

$$U_x(r) = U_\infty - U_{def} = U_\infty \cdot \left(1 - \hat{U}_M(x) \cdot e^{-3.56 \left(\frac{r}{b-D_r}\right)^2}\right) \quad (1)$$

Las ecuaciones que rigen las predicciones de velocidad incidente del modelo de Ainslie se almacenan en la función 'wm_ainslie()', que recibe parámetros como la estructura de datos, el módulo del viento, el indicador de la turbina y las distancias longitudinal y radial. Estos parámetros son suficientes para calcular la velocidad deficitaria y estimar la velocidad final incidente en el aerogenerador inmerso en la estela turbulenta.

Implementación de 1Px mutativo como sustrato

Se ha decidido incorporar el operador de cruce por corte de 1 punto (1Px) como sustrato al algoritmo CRO-SL debido al buen rendimiento demostrado por el Algoritmo Genético Básico (BGA) en terrenos de alta densidad de ocupación en un estudio previo [12]. Esto amplía el comportamiento exploratorio de las soluciones al introducir cambios en la posición de las turbinas. Sin embargo, se observó en el estudio anterior que cambios bruscos en la posición de las turbinas aumentaban el efecto deficitario causado por las estelas, lo que resultaba en una disminución de la energía generada. Con la incorporación de este sustrato más conservador, se busca mejorar el comportamiento

explotativo del algoritmo CRO-SL.

El sustrato funciona seleccionando otra larva del coral para cruzarse con la larva del sustrato. Se elige aleatoriamente a un tercio de las larvas y se les asigna una probabilidad de ser seleccionadas para el cruce según su valor de la función de *fitness*. Se selecciona un punto aleatorio para realizar el corte en la cadena de genes de ambas larvas, y se combinan para formar dos nuevas larvas. Se verifica que no haya genes repetidos en las larvas generadas y luego se elige la larva con el mejor valor de la función de *fitness* para intentar asentarse dentro del coral.

A continuación, se aplica un proceso de mutación a la larva seleccionada, con un 15% de probabilidades de que ocurra. Si se activa la mutación, hay un 30% de probabilidades de que cada gen de la larva se modifique aleatoriamente. La larva resultante de todo este proceso pasa a la etapa de asentamiento en el coral.

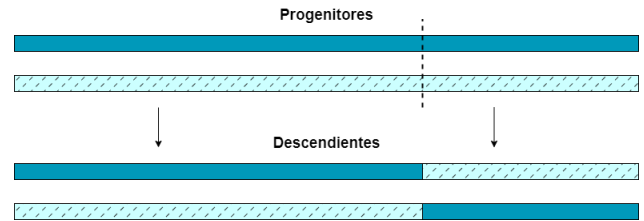


Fig. 3. Esquema de funcionamiento del operador 1Px.

Implementación de PSO como sustrato

Ajuste de hiperparámetros:

El estudio realizado en [13] analiza el efecto de los coeficientes de aceleración en los resultados obtenidos por el algoritmo PSO, por lo que es importante ajustar los hiperparámetros para su correcto funcionamiento. Con el objetivo de optimizar el sustrato, se establecen rangos de valores para los hiperparámetros del PSO, como el factor de inercia (ω), la aceleración personal (c_1) y la aceleración social (c_2). Se utiliza la función 'meshgrid()' de MATLAB para generar una matriz tridimensional con todas las combinaciones de valores de los coeficientes. Se ajusta el PSO mediante la ejecución de la nueva estructura orientada a ajustar los hiperparámetros, evaluando todas las combinaciones de valores especificados. Se establece una población de 15 individuos y 40 iteraciones por cada conjunto de hiperparámetros. Los resultados muestran que los valores óptimos para los hiperparámetros en un terreno con 30 turbinas son $[c_1, c_2, \omega] = [1.3, 1.0, 1.0]$, con un valor máximo de *fitness* de $7.7774 \cdot 10^7$ kWh. Se menciona que el uso de diferentes modelos de estela puede afectar los resultados óptimos obtenidos.

Adaptación del PSO como sustrato:

La implantación del PSO como sustrato sigue un funcionamiento similar al PSO en su forma algorítmica [14]. Se agrupan los genes de las larvas en grupos de partículas que siguen a la partícula líder, la cual pertenece a la larva con mejor *fitness*. Hay una probabilidad del 10% de que las partículas cambien de grupo con otra partícula del mismo individuo y sigan a una nueva partícula líder. Para realizar estos cambios de grupo, se permuta el orden de los genes de la larva, lo que altera la referencia de la partícula líder a la que seguían e intercambia su mejor posición histórica con la de la otra partícula. Esto modifica los patrones de movimiento, permitiendo explorar nuevas soluciones y evitar convergencias.

En diagrama de la Figura 4 se muestra el proceso de ejecución del sustrato. Se pondera la permutación de los genes de la larva con una probabilidad del 10% para darle un factor exploratorio.

Posteriormente, las partículas que conforman la larva se transforman de un formato vectorial a una matriz de posiciones, y se aplican desplazamientos lineales según el grupo al que pertenezcan y su referencia. Se verifica que no haya repeticiones de posiciones dentro de las partículas de la larva, y se realiza una transformación para volver a la forma original de la larva a partir de la matriz de posiciones.

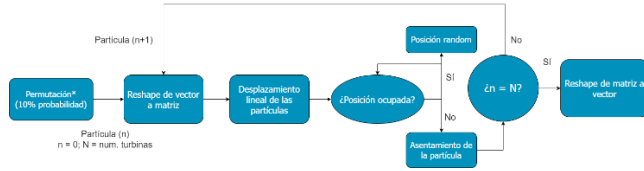


Fig. 4. Diagrama de flujo del *Particle Swarm Optimization* (PSO).

Implementación de WOA como sustrato

El uso de un sustrato basado en el algoritmo *Whale Optimization Algorithm* (WOA) puede ayudar a romper la duplicidad de individuos en generaciones altas y contrarrestar la tendencia convergente de algunos sustratos del algoritmo CRO-SL. Para implementar este sustrato, se ha utilizado la base del algoritmo PSO y se han ajustado los hiperparámetros según los valores recomendados en la literatura [15].

Los hiperparámetros del WOA, como la amplitud, el control de amplitud, la contracción, la distancia contraída y la distancia, se establecen dentro de rangos específicos. Estos valores influyen en el comportamiento del sustrato y en el desplazamiento de las ballenas hacia las coordenadas de referencia asignadas. El sustrato del WOA funciona de manera similar al sustrato del PSO, con pequeñas modificaciones en el movimiento de las turbinas en el plano. Las ballenas que componen la larva pueden desplazarse de manera rectilínea hacia una posición de referencia o seguir una trayectoria en espiral, con una probabilidad del 50% para cada opción. La posición de referencia de las ballenas varía según el valor del hiperparámetro de amplitud (\bar{A}).

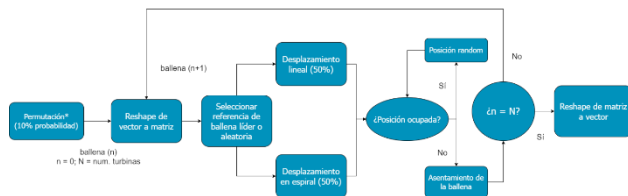


Fig. 5. Diagrama de flujo del *Whale Optimization Algorithm* (WOA).

Función generadora de desniveles y rugosidad del terreno

La función 'genRelief()' genera dos outputs: una matriz de desniveles que representa la coordenada z del plano y modifica la longitud radial del cono, lo que disminuye el déficit de velocidad entre la turbina generadora de la estela y la turbina afectada. Esto resulta en un mayor módulo de velocidad del viento en las coordenadas especificadas.

La ejecución de la función 'genRelief()' requiere los paquetes de MATLAB *Curve Fitting Toolbox* e *Image Processing Toolbox*. Para generar la matriz de relieves, se aplica un filtro promediador utilizando la función 'fspecial()' y se especifica el número de puntos de entrada.

En cuanto a la generación de la matriz de rugosidad, se establece una ley de distribución de rugosidad que depende del tipo de terreno. Para la distribución de longitudes de rugosidad superficial se adaptó la ecuación de Lettau [16], la cual calcula

los componentes de la matriz de rugosidad (Z_0) en función de la longitud de rugosidad de referencia (z_0), la matriz de relieve (Z) y la altitud media (z_{med}). Esta ecuación se puede modificar para crear distribuciones de rugosidad más complejas.

Además, se adapta la función 'getDistTrapz()' para tener en cuenta el desnivel entre las turbinas generadora y afectada por la estela. También se modifican las ecuaciones de algunos modelos de estela, como el modelo de Jensen, para incorporar el uso de la rugosidad variable almacenada en la matriz de rugosidad y recalcular los parámetros dependientes de ella en cada punto de cálculo.

V. RESULTADOS

Se han elaborado una serie de experimentos para evaluar el correcto funcionamiento de las implementaciones realizadas sobre el algoritmo CRO-SL con el que se contaba. Se procede a la exposición de los resultados conseguidos en una serie de experimentos, los cuales miden la capacidad de obtención de soluciones óptimas, la participación de los distintos sustratos en la mejora de soluciones, la obtención de la mejor solución global, la afectación sobre las predicciones estimadas tras la incorporación de relieve/rugosidad variable sobre terreno y los tiempos de ejecución del nuevo modelo de estela implementado, definido Ainslie.

Resultados del proceso de optimización

Con el fin de comprobar el grado de mejora que sufre el algoritmo CRO-SL tras la incorporación de 3 sustratos nuevos (1Px, PSO y WOA) se realiza una comparativa entre los resultados obtenidos por la versión estándar el CRO-SL que contaba con 4 sustratos únicamente y el Algoritmo Genético Básico (BGA).

En la Tabla I se registran los valores de energía anual generada, estimada por los distintos algoritmos y modelos de estela, haciendo uso de la matriz de vectores de vientos reales establecida, de tamaño 2×8760 , la cual recoge la velocidad y dirección del viento durante todas las horas de un año en un punto de la provincia de Badajoz (España).

TABLA I

PRODUCCIÓN ANUAL DE ENERGÍA DE LAS DISTRIBUCIONES OPTIMIZADAS

ALGORITMO	MODELO DE ESTELA	ENERGÍA (kWh)
BGA	Jensen	$7.7818 \cdot 10^7$
	Larsen	$7.7527 \cdot 10^7$
	Ainslie	$7.7896 \cdot 10^7$
CRO-SL (4 sustratos)	Jensen	$7.7722 \cdot 10^7$
	Larsen	$7.4619 \cdot 10^7$
	Ainslie	$7.7882 \cdot 10^7$
CRO-SL (7 sustratos)	Jensen	$7.7856 \cdot 10^7$
	Larsen	$7.6144 \cdot 10^7$
	Ainslie	$7.8004 \cdot 10^7$

Los valores más elevados de producción de energía por cada modelo de estela son conseguidos por el algoritmo CRO-SL al que se le han incorporado los nuevos sustratos basados en el operador 1Px y los algoritmos PSO y WOA, haciendo un cómputo total de 7 sustratos. Tras él, se posiciona el BGA como el segundo mejor algoritmo a la hora de optimizar las

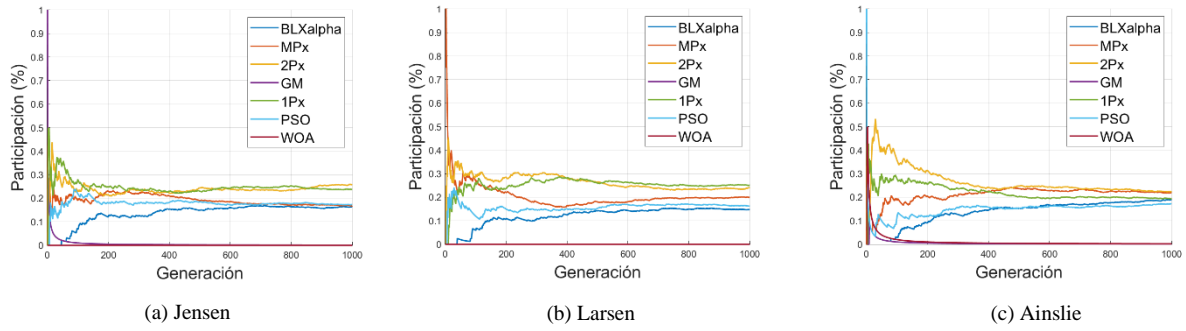


Fig. 6. Evolución de participación de los sustratos del CRO-SL (7 sustratos).

distribuciones de aerogeneradores, muy por encima del algoritmo CRO-SL original, que contaba únicamente con 4 sustratos (BLX α , MPx, 2Px y GM), el cual consigue las peores soluciones.

Participación de los sustratos en las soluciones óptimas

En la Tabla II se recoge el sumatorio de las generaciones en las que cada uno de los sustratos ha generado la mejor de las larvas, con los resultados de los 3 sustratos incorporados enfatizados. Tanto el CRO-SL con cuatro sustratos como el que incorporaba siete sustratos fueron ejecutados 1000 generaciones por cada uno de los modelos de estela seleccionados, dando un total de 3000 generaciones entre todos ellos.

Es importante destacar, que en el desglose de resultados de participación para cada uno de los modelos de estela (1000 generaciones cada uno) mostraban porcentajes de participación muy similares para todos ellos. Lo que demuestra un impacto leve del modelo de estela que se use en la ejecución de los sustratos dentro del algoritmo y, por tanto, permitiendo realizar el sumatorio de participación sin perder información relevante sobre el funcionamiento.

TABLA II

Nº. DE GENERACIONES EN LAS QUE DOMINA CADA SUSTRATO

SUSTRATO	CRO-SL (4 SUSTRATOS)	CRO-SL (7 SUSTRATOS)
BLX α	1115	499
MPx	837	586
2Px	925	721
GM	133	2
1Px	-	683
PSO	-	507
WOA	-	2
Nº. Gen.	3000	3000

Según los resultados obtenidos, los sustratos más importantes pertenecen a operadores de cruce, entre los que se encuentra el operador 1Px incorporado en este estudio, con una participación en 683 generaciones. De igual forma, otro sustrato que adquiere bastante relevancia es el sustrato basado en el algoritmo PSO, el cual se añadió a la estructura junto con el 1Px, y que consigue los mejores resultados en 507 generaciones, en la línea de otros sustratos como el MPx (586) o el BLX α (499). El último de los sustratos añadidos, el WOA, queda apartado en un segundo plano consiguiendo las mejores larvas únicamente en 2 generaciones. En la Figura 6 se puede revisar la evolución de la participación de

los distintos sustratos del CRO-SL modificado, a lo largo de las 1000 generaciones establecidas para los distintos modelos de estela escogidos.

Producto cartesiano de las distribuciones por modelo

Dado que las mejores soluciones fueron alcanzadas por el algoritmo CRO-SL de 7 sustratos, se procede a la realización del producto cartesiano de soluciones haciendo uso de las 3 distribuciones obtenidas tras el proceso de optimización llevado a cabo durante 1000 generaciones para cada uno de los modelos de estela.

Es imprescindible tener en cuenta el porcentaje medio de rendimiento de cada una de las soluciones conseguidas según la siguiente ecuación:

$$Ren(\%) = \frac{E_{gen} - E_{min}}{E_{max} - E_{min}} \cdot 100 \quad (2)$$

Esto permite que una solución no se vea excesivamente perjudicada por los márgenes mínimos de predicción establecidos por un modelo de estela, como puede ser el modelo de estela de Larsen que tiende a penalizar mucho el solapamiento de estelas producidas por los aerogeneradores. En la Tabla III se recogen los valores mínimos y máximos que pueden predecir los modelos de estela para una configuración de proporción de terreno de 30x30 con 30 turbinas eólicas posicionadas.

TABLA III

PREDICCIONES DE ENERGÍA MÁXIMAS Y MÍNIMAS DE LOS MODELOS DE ESTELA PARA CONFIGURACIÓN DE 30x30 CON 30 AEROGENERADORES

MODELO DE ESTELA	ENERGÍA MÍNIMA (kWh)	ENERGÍA MÁXIMA (kWh)
Jensen	$4.0803 \cdot 10^7$	$7.8064 \cdot 10^7$
Larsen	$9.5351 \cdot 10^6$	$7.8064 \cdot 10^7$
Ainslie	$7.1726 \cdot 10^7$	$7.8064 \cdot 10^7$

En la Tabla IV se pueden examinar los rendimientos obtenidos por las distintas distribuciones obtenidas a raíz de la ejecución del proceso de optimización del algoritmo CRO-SL tras la ejecución del producto cartesiano de las soluciones con cada uno de los modelos de estela. El rendimiento medio marca que la mejor de las soluciones fue obtenida por el modelo de estela de Jensen con un rendimiento del 95.88%, tal y como se enfatiza en el promedio la tabla.

TABLA IV

RENDIMIENTO DEL PRODUCTO CARTESIANO DE LAS SOLUCIONES CON
LOS MODELOS DE ESTELA (RENDIMIENTO, %)

SOLUCIÓN MODELO	JENSEN	LARSEN	AINSLIE
JENSEN	99.44%	96.85%	92.26%
LARSEN	91.45%	97.20%	91.41%
AINSLIE	96.27%	86.99%	99.05%
PROMEDIO	95.88%	93.68%	94.24%

Afectación del relieve y los cambios en la rugosidad

Resultado de la comparativa de los cambios de altitud:

Para llevar a cabo la validación de la función generadora de la matriz de desniveles y su correcta implementación en la función de *fitness* a la hora de prever la energía generada en cada distribución, se ha establecido un margen de variación comprendido entre [0, 10] m. Los mapas de relieve y de contorno utilizados para los experimentos se pueden ver en la Figura 7.

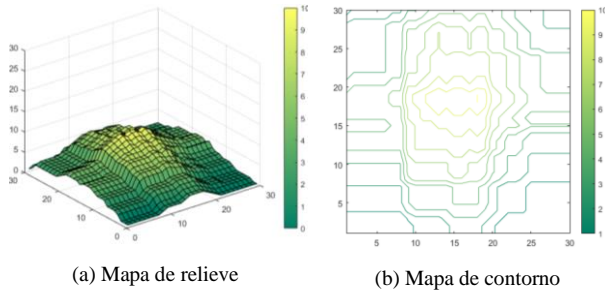


Fig. 7. Mapas de relieve y contorno del terreno.

La Tabla V marca las diferencias producidas entre las distribuciones óptimas obtenidas por cada modelo de estela haciendo uso de los distintos algoritmos metaheurísticos planteados. Se hará uso del modelo de estela de Ainslie para analizar cada una de las soluciones obtenidas durante el proceso de optimización, debido a su alta volatilidad en la predicción de resultados, lo cual ayuda a denotar la diferencia ante cambios de altitud tan suaves como el establecido.

TABLA V

COMPARATIVA DE ENERGÍA ESTIMADA POR LOS MODELOS DE ESTELA
SEGÚN LA TIPOLOGÍA DE RELIEVE DEL TERRENO (AINSLIE)

ALGORITMO	SOLUCIÓN	CON RELIEVE (kWh)	SIN RELIEVE (kWh)
BGA	Jensen	$7.7737 \cdot 10^7$	$7.7736 \cdot 10^7$
	Larsen	$7.7499 \cdot 10^7$	$7.7736 \cdot 10^7$
	Ainslie	$7.7935 \cdot 10^7$	$7.7896 \cdot 10^7$
CRO-SL (4 sustratos)	Jensen	$7.7658 \cdot 10^7$	$7.7656 \cdot 10^7$
	Larsen	$7.7184 \cdot 10^7$	$7.7182 \cdot 10^7$
	Ainslie	$7.7925 \cdot 10^7$	$7.7882 \cdot 10^7$
CRO-SL (7 sustratos)	Jensen	$7.7829 \cdot 10^7$	$7.7828 \cdot 10^7$
	Larsen	$7.7526 \cdot 10^7$	$7.7524 \cdot 10^7$
	Ainslie	$7.8005 \cdot 10^7$	$7.8004 \cdot 10^7$

Estas diferencias se encuentran entre un rango de valores situado en torno a $10^3 - 10^4$ kWh en función de la distribución

seleccionada para realizar el cálculo de la función de *fitness* con el modelo de estela de Ainslie y la distribución de turbinas utilizada.

Resultados de la comparativa de los cambios de rugosidad:

En función de la matriz de relieve establecida, se genera una matriz de rugosidad (Z_0) que otorga un valor de longitud de rugosidad a cada coordenada del terreno, afectando a los resultados obtenidos por algunos modelos de estela que hacen uso de este parámetro como el modelo de Jensen, el cual se utilizará para analizar todas las soluciones óptimas obtenidas tras la ejecución de los algoritmos metaheurísticos con cada modelo de estela y compararlas posteriormente. En la Figura 8 se pueden observar los mapas de superficie y de contorno que registran los distintos valores de la matriz.

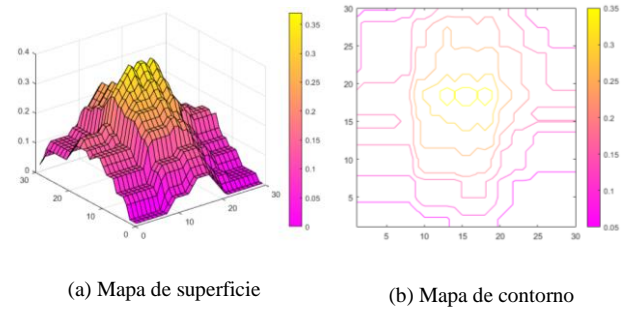


Fig. 8. Mapas de superficie y contorno de la rugosidad.

Al contrario que sucedía con las variaciones de altitud debido a los desniveles, la implementación de la generación de longitud de rugosidad superficial establecida provoca cambios muy notorios en la mayoría de los casos analizados. La diferencia entre los valores variables y constantes de rugosidad se sitúan en magnitudes comprendidas entre $10^5 - 10^6$ kWh, lo cual denota un gran impacto sobre la energía estimada por el modelo de Jensen.

TABLA VI

COMPARATIVA DE ENERGÍA ESTIMADA POR LOS MODELOS DE ESTELA
SEGÚN LA TIPOLOGÍA DE RUGOSIDAD DEL TERRENO (JENSEN)

ALGORITMO	SOLUCIÓN	VARIABLE (kWh)	CONSTANTE (kWh)
BGA	Jensen	$7.5866 \cdot 10^7$	$7.7818 \cdot 10^7$
	Larsen	$7.7256 \cdot 10^7$	$7.7444 \cdot 10^7$
	Ainslie	$7.5477 \cdot 10^7$	$7.7678 \cdot 10^7$
CRO-SL (4 sustratos)	Jensen	$7.7023 \cdot 10^7$	$7.7722 \cdot 10^7$
	Larsen	$7.6644 \cdot 10^7$	$7.7244 \cdot 10^7$
	Ainslie	$7.7422 \cdot 10^7$	$7.7547 \cdot 10^7$
CRO-SL (7 sustratos)	Jensen	$7.7547 \cdot 10^7$	$7.7856 \cdot 10^7$
	Larsen	$7.6153 \cdot 10^7$	$7.7722 \cdot 10^7$
	Ainslie	$7.7499 \cdot 10^7$	$7.7606 \cdot 10^7$

Tiempos de ejecución del modelo de estela de Ainslie

Como se ha mostrado en los resultados obtenidos en la Tabla I, el modelo de Ainslie incorporado, describe predicciones de generación de energía notablemente más elevadas que los modelos de Jensen y Larsen. Esto provoca una alta volatilidad entre los valores acotados entre el límite máximo y mínimo de predicción.

TABLA VII

TIEMPOS DE EJECUCIÓN DE LOS MODELOS DE ESTELA

MODELO DE ESTELA	TIEMPO DE EJECUCIÓN 20 ITERACIONES (s)	TIEMPO DE EJECUCIÓN PROMEDIO (s)
Jensen	25.2272	1.2614
Larsen	32.2178	1.6109
Ainslie	28.3428	1.4171

En la Tabla VII, se han recogido los tiempos de ejecución de los tres algoritmos utilizados a lo largo de 20 iteraciones, y el valor temporal promedio que ha tardado en ejecutar cada iteración. Lo que deja al modelo de estela de Jensen como el más rápido en ejecutarse con una velocidad promedio de 1.2614 s, seguido de los modelos de Ainslie con 1.4171 s y Larsen con 1.6109 s.

VI. DISCUSIÓN

En lo referente al análisis de los resultados conseguidos tras el proceso de optimización con cada uno de los algoritmos metaheurísticos presentados y sus distintas configuraciones, el algoritmo BGA sigue siendo el más veloz en alcanzar las 1000 generaciones, mientras que las dos versiones del algoritmo CRO-SL se ejecutan más lentamente. En términos de los resultados de energía producida, basados en la función de *fitness* de las mejores distribuciones encontradas, se puede concluir que la implementación de los 3 nuevos sustratos en el CRO-SL ha sido exitosa. Estas soluciones dominan entre el conjunto de soluciones obtenidas por cada algoritmo en los 3 modelos de estela analizados (Larsen, Ainslie y Jensen). Los valores de rendimiento de esta versión modificada del CRO-SL mejoran significativamente los resultados, especialmente con los modelos de Larsen y Ainslie, aunque la mejora es menos notable con el modelo de Jensen.

Es importante destacar que las soluciones obtenidas por el algoritmo CRO-SL de 7 sustratos muestran un crecimiento más pronunciado y rápido en comparación con las soluciones obtenidas con 4 sustratos. Esto significa que los valores promedio de las soluciones alcanzadas con 7 sustratos se acercan más al valor máximo alcanzado desde las primeras generaciones, mientras que, con 4 sustratos, es necesario avanzar en generaciones más avanzadas para lograr resultados similares.

Se observa que el modelo de estela de Jensen alcanzó el mejor valor de *fitness* en una generación más temprana que los otros modelos. Esto podría deberse a una actuación más eficiente del sustrato del PSO, cuyos hiperparámetros se optimizaron utilizando este modelo. Los modelos de Larsen y Ainslie obtuvieron sus mejores soluciones en una etapa generacional posterior, cuando las soluciones estaban más asentadas. El uso de los modelos de estela de Ainslie y Larsen requiere la aplicación de múltiples ecuaciones implícitas para obtener la velocidad incidente en puntos afectados por las estelas. Esto genera tiempos de cálculo más largos en comparación con modelos más simples como el modelo de estela de Jensen. En términos de tiempo de inversión para el cálculo de soluciones, el modelo propuesto por Ainslie se posiciona como una alternativa intermedia entre los modelos de Jensen y Larsen.

En cuanto a la participación de los sustratos a lo largo de las generaciones, se detecta que ciertos sustratos tienen un impacto más notable en las primeras iteraciones, cuando el valor medio de *fitness* del coral es bajo. Esto se debe a que las larvas del coral son fácilmente reemplazables en esta etapa. A medida que el

proceso evoluciona, se estabiliza la participación de los sustratos, y los menos efectivos son anulados, mostrando un comportamiento más uniforme entre los sustratos restantes.

Incorporar desniveles en la ubicación de turbinas eólicas implica un aumento en la distancia entre ellas y en la superficie de terreno requerida. Esto reduce el déficit de velocidad sufrido y aumenta la velocidad del viento en las turbinas afectadas. Aunque los desniveles tienen una leve incidencia en la producción de energía, se ha demostrado que la adaptación del cálculo de distancias utilizando la función de *fitness* es exitosa.

El uso de mapas de altitudes en el terreno incrementa la distancia radial del cono de estela, reduciendo el efecto deficitario de la turbulencia sobre la velocidad del viento incidente y aumentando la energía generada del conjunto de turbinas eólicas. La implementación de matrices de rugosidades con el uso de una ley de distribución de longitudes de rugosidad puede suponer una variación considerable en los resultados de predicción energética generada. Sin embargo, como ya se mencionó, esta implementación es limitada, ya que solo es compatible con modelos basados en el modelo de estela de Jensen y que consideran dicho parámetro en el cálculo de las velocidades internas de la estela turbulenta, tales como el modelo Park2 [17].

VII. CONCLUSIONES

En este estudio, se utilizó el algoritmo CRO-SL para optimizar la colocación de aerogeneradores en un terreno para un parque eólico. Se discretizó el terreno en un *grid* de 30x30 con 30 turbinas. Se agregaron tres nuevos sustratos basados en el operador 1Px, el algoritmo PSO y el WOA para mejorar los resultados. Se incorporó un nuevo modelo de estela de Ainslie a la librería existente y se consideraron terrenos con altitud no uniforme y diferentes longitudes de rugosidad superficial.

La evolución de la función de *fitness* mostró un crecimiento suave y continuo en el algoritmo BGA, mientras que en el CRO-SL, especialmente con los nuevos sustratos, se observó un crecimiento más escalonado debido a los periodos de estancamiento y cambios bruscos en los valores de *fitness*. La inclusión de los nuevos sustratos suavizó la curva de crecimiento y aceleró la generación de soluciones válidas, lo que se tradujo en un aumento en la energía generada.

La adición de los sustratos 1Px y PSO demostró mejorar la obtención de soluciones y aumentar la producción de energía. Sin embargo, el sustrato WOA tuvo un impacto limitado y prácticamente residual. A pesar de esto, la implementación de los nuevos sustratos fue beneficioso en general.

La función generadora de relieves y rugosidades funcionó correctamente, aunque su aplicabilidad está limitada a ciertos modelos de estela que consideran el parámetro de longitud de rugosidad (z_0) y requieren grandes desniveles para mostrar cambios significativos en las predicciones de energía generada en parques eólicos marinos.

Este estudio aborda varias metas de los Objetivos de Desarrollo Sostenible (ODS), especialmente las incluidas en el ODS 7, cumpliendo un total de 4 de sus 5 metas específicas (ODS 7.1, ODS 7.2, ODS 7.3 y ODS 7.b). Además, también tiene un impacto más colateral sobre los ODS 8 y 12, relacionándose en determinados aspectos con las metas ODS 8.4 y ODS 12.2.

Para futuros estudios, se sugieren varias áreas de investigación que podrían mejorar la optimización del posicionamiento de turbinas eólicas en diferentes contextos:

- Considerar el ángulo de resbalamiento en los cálculos de la función de *fitness*.
- Explorar la vinculación automatizada de algoritmos metaheurísticos con herramientas de análisis CFD más precisas.
- Identificar parámetros que permitan el diseño de nuevos algoritmos y restricciones que reduzcan el espacio de búsqueda de soluciones.
- Investigar el uso de turbinas eólicas en aplicaciones de defensa, como frenar ciclones o tornados, ampliando el alcance de la colocación de turbinas más allá de la generación de energía.
- Evaluar el impacto de los fallos y períodos de inactividad de las turbinas en la eficiencia de la distribución de turbinas, comparándola con otras configuraciones y comprendiendo mejor el rendimiento de una granja eólica en situaciones de interrupción.
- Realizar modelizaciones matemáticas de las estelas generadas por turbinas eólicas en Marte, explorando su aplicación en la generación de energía en el planeta y su relevancia para la exploración espacial y la colonización.

Aerodynamics, 27, 213-224. doi:[https://doi.org/10.1016/0167-6105\(88\)90037-2](https://doi.org/10.1016/0167-6105(88)90037-2)

- [12] Romero Barrera, A. J. (22 de Mayo de 2023). *DISEÑO DE ALGORITMOS METAHEURÍSTICOS PARA EL PROBLEMA DE COLOCACIÓN DE TURBINAS EÓLICAS SOBRE EL TERRENO*. Trabajo Fin de Grado, Universidad Rey Juan Carlos, Escuela de Ingeniería de Fuenlabrada, Fuenlabrada. Obtenido de <https://burjcdigital.urjc.es/handle/10115/22188>
- [13] Rehman, S., Khan, S., & Alhems, L. (January de 2020). The effect of acceleration coefficients in Particle Swarm Optimization algorithm with application to wind farm layout design. *FME Transactions*, 48, 922-930. doi:10.5937/fme2004922R
- [14] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, 4, págs. 1942-1948 vol.4. doi:10.1109/ICNN.1995.488968
- [15] Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, 95, 51-67. doi:<https://doi.org/10.1016/j.advengsoft.2016.01.008>
- [16] Gaona, G., Balcázar, L., & Jaime, I. (2015, April). Estimación de longitud de rugosidad aerodinámica a partir de imágenes MODIS y LandSat.
- [17] Rathmann, O. S., Hansen, B. O., Hansen, K. S., Mortensen, N. G., & Murcia Leon, J. P. (2018). *The Park2 Wake Model - Documentation and Validation*.

REFERENCIAS

- [1] IRENA. (2023). Renewable capacity statistics 2023. Abu Dhabi.
- [2] Orús, A. (March of 2023). Energía Eólica: Generación por CC.AA en España en 2022. *Energía Eólica: Generación por CC.AA en España en 2022*. Obtenido de <https://es.statista.com/estadisticas/1004140/generacion-eolica-por-region-en-espana/>
- [3] Salcedo-Sanz, S., Del Ser, J., Gil-López, S., Landa-Torres, I., Portilla-Figueras, J.A.: The Coral Reefs Optimization Algorithm: A new metaheuristic algorithm for hard optimization problems. In: Proc. of the 15th International Conference on Applied Stochastic Models and Data Analysis (ASMDA), Mataró, Barcelona (2013)
- [4] Salcedo-Sanz, S. (October of 2016). A review on the coral reefs optimization algorithm: New development lines and current applications. *Progress in Artificial Intelligence*, 6. doi:10.1007/s13748-016-0104-2
- [5] A. Asghari and M. K. Sohrabi, "Multiobjective Edge Server Placement in Mobile-Edge Computing Using a Combination of Multiagent Deep Q-Network and Coral Reefs Optimization," in *IEEE Internet of Things Journal*, vol. 9, no. 18, pp. 17503-17512, 15 Sept.15, 2022, doi: 10.1109/JIOT.2022.3161950.
- [6] Tsai, CW., Chang, WY., Wang, YC. et al. A high-performance parallel coral reef optimization for data clustering. *Soft Comput* 23, 9327–9340 (2019). <https://doi.org/10.1007/s00500-019-03950-3>
- [7] Salcedo-Sanz, S., Gallo-Marazuela, D., Pastor-Sánchez, A., Carro-Calvo, L., Portilla-Figueras, A., & Prieto, L. (2014). Offshore wind farm design with the Coral Reefs Optimization algorithm. *Renewable Energy*, 63, 109-115. doi:<https://doi.org/10.1016/j.renene.2013.09.004>
- [8] Salcedo-Sanz, S., Pastor-Sánchez, A., Ser, J. D., Prieto, L., & Geem, Z. W. (2015). A Coral Reefs Optimization algorithm with Harmony Search operators for accurate wind speed prediction. *Renewable Energy*, 75, 93-101. doi:<https://doi.org/10.1016/j.renene.2014.09.027>
- [9] Salcedo-Sanz, Sancho & Pastor-Sánchez, Álvaro & Portilla-Figueras, Antonio & Prieto, Luis. (2015). Effective multi-objective optimization with the coral reefs optimization algorithm. *Engineering Optimization*. 1-19. 10.1080/0305215X.2015.1078139.
- [10] S. Salcedo-Sanz, C. Camacho-Gómez, D. Molina and F. Herrera, "A coral reefs optimization algorithm with substrate layers and local search for large scale global optimization," 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 2016, pp. 3574-3581, doi: 10.1109/CEC.2016.7744242.
- [11] Ainslie, J. F. (1988). Calculating the flowfield in the wake of wind turbines. *Journal of Wind Engineering and Industrial*

Design and implementation of substrates in CRO-SL algorithm for the distribution of wind turbines

Antonio José Romero Barrera

Universidad Internacional de la Rioja, Logroño (España)

Date: 13/07/2023

unir
LA UNIVERSIDAD
EN INTERNET

 **Universidad
de Alcalá**

 **Universidad
Rey Juan Carlos**

ABSTRACT

Metaheuristic algorithms have emerged as an effective approach to address complex problems in various fields. This study aims to find the distribution of wind turbines that maximizes the annually generated energy within a wind farm. By using the population-based algorithm Coral Reef Optimization with Substates Layer (CRO-SL) and incorporating 3 additional substrates based on the 1Px operator and the Particle Swarm Optimization (PSO) and Whale Optimization (WOA) algorithms, an improvement in the growth trend of solutions and the quality of the obtained results is achieved when distributing wind turbines over a discretized offshore terrain of size 30x30. A total of 3 wake models (Jensen, Larsen, and Ainslie) have been used to estimate the velocity deficit affecting the wind turbines and calculate the generated energy through a power curve of an AWECC-60 wind turbine.

KEYWORDS

algorithms, CRO-SL, metaheuristics, optimization, wind farm.

I. INTRODUCTION

In a world in constant growth, both at the technological level and at the population level, it is essential to make proper use of the new tools that arise to improve the quality of life of citizens. However, human action has had serious consequences on the planet, such as pollution, global warming, deforestation, overpopulation and loss of biodiversity. To counteract these negative effects, institutions such as the European Union (EU) and the United Nations (UN) have implemented initiatives such as Horizon 2030, the Coalition for Carbon Neutrality and the Sustainable Development Goals (SDGs).

Among renewable energies, wind energy occupies a prominent place. In Spain, wind energy has a majority share within the renewable energy market, representing approximately 22%. The country has more than 21,500 wind turbines, generating more than 60 TWh annually [1]. Regions such as Castilla y León, Aragón and Galicia stand out as the main sources of wind energy production in Spain [2].

Therefore, it is essential to optimize the operation of wind turbines by correctly determining their position in relation to others. This will minimize the negative effects produced by the contrails generated by the rotation of the rotors of wind turbines. When air passes through the rotor of a wind turbine, there is an alteration of the flow that disturbs the stability and direction of the wind, generating turbulence that slows the downstream current. This turbulence is contained in a conical trail, which delimits the deficit effect in that control volume.

To maximize the efficiency and energy production of wind

farms, it is necessary to carry out comprehensive analyses and studies on the optimal location of the turbines, considering factors such as wind speed and direction, terrain relief and the interaction between the contrails generated by the turbines. These assessments minimize interference between turbines and maximize the generation of clean and renewable energy.

II. STATE OF THE ART

In this study, an exhaustive analysis of the CRO (*Coral Reef Optimization*) algorithm [3] and its variants is carried out, with the aim of providing a comprehensive view of the state of research in this field. Four research questions are addressed in which it is studied: (RQ1) the fields of application of the CRO algorithm, (RQ2) the modifications arising from the base algorithm and (RQ3) the growth trend in the number of publications related to the CRO over the years and (RQ4) the collaboration between researchers and institutional affiliations to understand the research network around the CRO algorithm.

The results reveal that the CRO algorithm has found applications in various sectors, highlighting in computer science, telecommunications, energy and medicine / biomedical engineering, as shown in the developed pie chart, shown in Figure 1. In the research work carried out Salcedo-Sanz [4], an exhaustive review of the CRO algorithm is carried out, addressing its applications in various sectors. Use cases are analyzed in energy, such as wind turbine placement and wind prediction, in

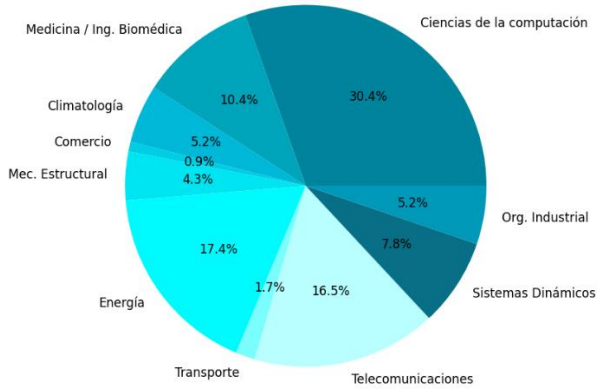


Fig. 1. Fields of research that made use of the CRO in their publications.

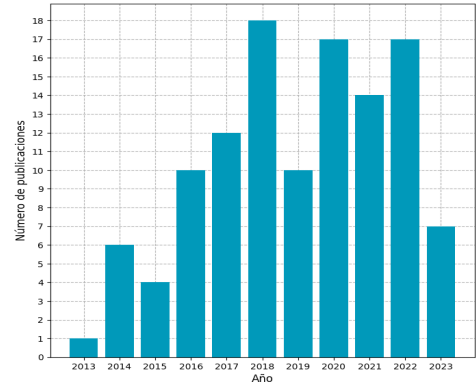


Fig. 2. Evolution of the number of publications per year related to the CRO from 2013 to June 2023.

telecommunications, such as mobile network optimization [5], and in computer science, such as *clustering* of datasets [6].

In the energy field, the article [7] that studies the performance of the CRO in the design of offshore wind farms stands out. The results are compared with other metaheuristics, such as Differential Evolution (ED) and Harmony Search (HS), demonstrating that the CRO algorithm provides more accurate and effective solutions for optimization in problems related to wind energy.

In addition, the CRO algorithm has been applied in the analysis of climatological factors for renewable energies. The study [8] presents the hybrid CRO-HS algorithm, which combines CRO with Harmony Search methodology, improving its performance. Its performance in wind speed prediction is evaluated using data from weather towers in the United States and Spain, obtaining results superior to the basic versions of the CRO and HS algorithms.

Variants of the algorithm have been developed, such as the aforementioned CRO-HS (*Coral Reef Optimization with Harmony Search*), which combines methodologies from different metaheuristic algorithms to improve its performance on specific problems. The use of the CRO in variable coding problems and in multiobjective optimization [9] or the CRO-SL [10] on which this work focuses has also been explored, and which divides the coral population into different groups called substrates on which a certain operator is applied.

In the graph of evolution of the number of publications per year of Figure 2, a fluctuating trend in the number of publications related to the CRO is observed, with a maximum peak in 2018. Although the number of publications is lower compared to other established metaheuristic algorithms, the CRO algorithm has managed to spread internationally, with the participation of researchers from different countries.

In short, the relevance and potential of the CRO algorithm in solving optimization problems in various fields is highlighted. Collaborations between researchers and institutional affiliations show the existence of a growing research network around the CRO algorithm. The CRO algorithm and its variants continue to be the subject of research and development, which opens new opportunities for the application of this technique in future studies and projects focused on new topics.

III. OBJECTIVES AND METHODOLOGY

The general objective of this TFM is the improvement of the operation and execution of the CRO-SL algorithm, through the incorporation of new substrates, which make better solutions than the initial CRO-SL algorithm that has 4 active substrates by default, with their corresponding probabilities of hosting individuals. The implemented substrates will be designed and adjusted by a tuning process of selected hyperparameters, aimed at optimizing the positioning of wind turbines on flat terrain (offshore wind farms), and also developing the functionality of analyzing the distributions of turbines with different altitudes (*onshore* wind farms).). Among the specific objectives to be met to meet the general objective, we have the following goals:

1. The objective is to analyze the impact of scientific research on the development of the CRO algorithm in recent years, identifying the problems in which the CRO-SL has been applied and evaluating the quality of the solutions obtained.
2. It seeks to expand the library of existing wake models by incorporating a new model that has not been previously used. The goal is to ensure that this new model can make adequate predictions and works correctly.
3. The goal is to improve the performance of the CRO-SL algorithm by adding new substrates based on unused operators or metaheuristic algorithms. These new substrates seek to optimize the positioning of wind turbines.
4. It is intended to analyze the influence of each substrate in the solutions obtained by analysis with the CRO-SL. The objective is to evaluate the behavior of substrates over generations and measure their degree of improvement in optimizing the positioning of wind turbines.
5. The objective is to incorporate the possibility of generating terrains with relief and adapt the steele models to analyze the distributions with unevenness. The aim is to develop a function that allows a three-dimensional prediction of the speed deficit caused by wind turbine contrails in environments with irregular terrain.
6. The objective is to make a comparison between the modified CRO-SL with 7 substrates, the initial CRO-SL with 4 substrates and the BGA algorithm. The objective is to evaluate the performance and aptitude of each algorithm in solving the wind turbine distribution problem, considering aspects such as the ability to find optimal solutions, computational efficiency and stability of the results.

To carry out these objectives, the methodology and tools to carry out the analyses were selected weightatively. Options such as the use of tools based on computational fluid dynamics (CFD)

or wake models were considered, which turned out to be the most efficient option.

A comparison of results is made by recording the average energy generated by the population and the maximum energy produced by the best individual in each generation. This makes it possible to evaluate the evolution and speed of improvement of the algorithms used. The decrease in speed generated by each wake model and its impact on the distribution of wind turbines are also analyzed.

In addition, a comparative analysis of the optimized solutions obtained using the different metaheuristic algorithms and the different wake models is carried out. The occupation of turbines in the selected terrain is examined and the effectiveness of each algorithm is evaluated according to its ability to optimize this distribution.

IV. CONTRIBUTION

Ainslie wake model incorporation

The Ainslie wake model [11], is incorporated as the eighth model in this study, along with seven other previously used models. The Ainslie model is implemented as a function in the existing data structure and makes use of the 'genData()' function to initialize. To define the boundaries of the Ainslie stele, the limits of the Jensen model are used, allowing an almost zero estimate of the deficit near the periphery of the cone, thus validating their combined use.

To establish the final velocity that falls on the rotor of wind turbines affected by a turbulent wake, Equation (1) is applied, so that the deficit generated will depend on the dimensionless velocity on the axial axis of the wake at a certain distance ($\hat{U}_M(x)$), the radial distance from the affected turbine to said axis (r), the diameter of the rotor (D_r)

$$U_x(r) = U_\infty - U_{def} = U_\infty \cdot \left(1 - \hat{U}_M(x) \cdot e^{-3.56 \left(\frac{r}{b \cdot D_r} \right)^2} \right) \quad (1)$$

The equations governing the incident velocity predictions of the Ainslie model are stored in the 'wm_ainslie()' function, which receives parameters such as data structure, wind modulus, turbine gauge, and longitudinal and radial distances. These parameters are sufficient to calculate the deficit velocity and estimate the final velocity incident on the wind turbine immersed in the turbulent wake.

Implementation of mutative 1Px as a substrate

It has been decided to incorporate the 1-point (1Px) slice crossing operator as a substrate to the CRO-SL algorithm due to the good performance demonstrated by the Basic Genetic Algorithm (BGA) in high occupancy density terrains in a previous study [12]. This extends the exploratory behavior of the solutions by introducing changes in the position of the turbines. However, it was observed in the previous study that sudden changes in the position of the turbines increased the deficit effect caused by the contrails, resulting in a decrease in the energy generated. With the incorporation of this more conservative substrate, it seeks to improve the exploitative behavior of the CRO-SL algorithm.

The substrate works by selecting another coral larva to

interbreed with the substrate larva. One-third of the larvae are randomly chosen and assigned a probability of being selected for crossbreeding based on their fitness function value. A random point is selected to make the cut in the gene chain of both larvae, and they combine to form two new larvae. It is verified that there are no repeated genes in the generated larvae and then the larva with the best value of the fitness function is chosen to try to settle inside the coral.

Next, a mutation process is applied to the selected larva, with a 15% chance of it occurring. If the mutation is activated, there is a 30% chance that each gene in the larva will be randomly modified. The larva resulting from this whole process passes to the stage of settlement in the coral.

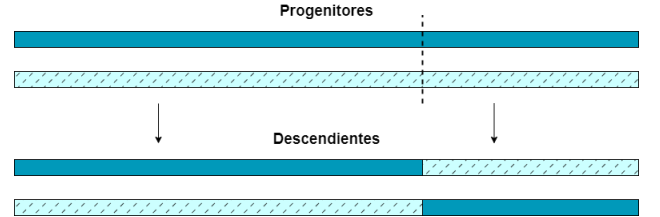


Fig. 3. Scheme of operation of the 1Px operator.

Implementation of PSO as a substrate

Hyperparameter tuning:

The study carried out in [13] analyzes the effect of the acceleration coefficients on the results obtained by the PSO algorithm, so it is important to adjust the hyperparameters for their correct operation. In order to optimize the substrate, ranges of values are established for the hyperparameters of the PSO, such as the inertia factor (ω), the personal acceleration (c_1) and social acceleration (c_2). The MATLAB 'meshgrid()' function is used to generate a three-dimensional matrix with all combinations of coefficient values. The PSO is adjusted by executing the new structure aimed at adjusting the hyperparameters, evaluating all combinations of specified values. A population of 15 individuals and 40 iterations is established for each set of hyperparameters. The results show that the optimal values for hyperparameters in a field with 30 turbines are $[c_1, c_2, \omega] = [1.3, 1.0, 1.0]$, with a maximum fitness value of $7.7774 \cdot 10^7$ kWh. It is mentioned that the use of different wake models can affect the optimal results obtained.

Adaptation of PSO as a substrate:

The implantation of PSO as a substrate follows a similar operation to PSO in its algorithmic form [14]. The genes of the larvae are grouped into groups of particles that follow the leader particle, which belongs to the larva with better fitness. There is a 10% chance that the particles will switch groups with another particle from the same individual and follow a new leading particle. To make these group changes, the order of the larva's genes is permuted, which alters the reference of the leader particle they followed and exchanges its best historical position with that of the other particle. This modifies movement patterns, allowing new solutions to be explored and convergence avoided.

The diagram in Figure 4 shows the substrate execution process. The permutation of the larva's genes is weighted with a probability of 10% to give it an exploratory factor. Subsequently, the particles that make up the larva are transformed from a vector format to a matrix of positions, and linear displacements are applied according to the group to which they belong and their reference. It is verified that there are no repetitions of positions

within the particles of the larva, and a transformation is made to return to the original shape of the larva from the matrix of positions.

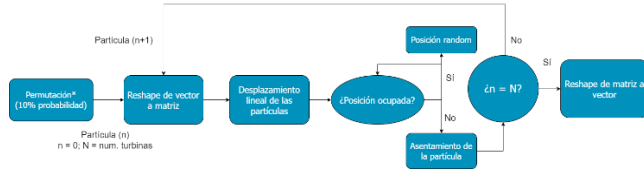


Fig. 4. Particle Swarm Optimization (PSO) flowchart.

Implementing WOA as a substrate

The use of a substrate based on the *Whale Optimization Algorithm* (WOA) can help break the duplicity of individuals in high generations and counteract the convergent trend of some substrates of the CRO-SL algorithm. To implement this substrate, the basis of the PSO algorithm has been used and the hyperparameters have been adjusted according to the values recommended in the literature [15].

WOA hyperparameters, such as amplitude, amplitude control, contraction, contracted distance, and distance, are set within specific ranges. These values influence the behavior of the substrate and the movement of the whales towards the assigned reference coordinates. The WOA substrate functions similarly to the PSO substrate, with minor modifications to the movement of the turbines in the plane. The whales that make up the larva can move rectilinearly towards a reference position or follow a spiral path, with a probability of 50% for each option. The reference position of whales varies according to the value of the amplitude hyperparameter (\vec{A}).

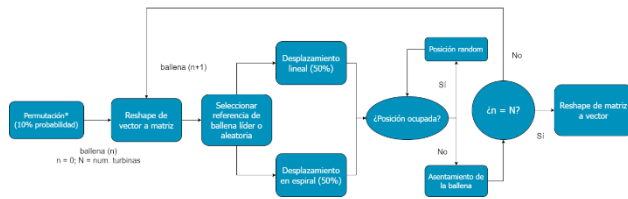


Fig. 5. Whale Optimization Algorithm (WOA) flowchart.

Function that generates relief and roughness on the terrain

The 'genRelief()' function generates two outputs: a slope matrix that represents the z-coordinate of the plane and modifies the radial length of the cone, which reduces the speed deficit between the wake generating turbine and the affected turbine. This results in a higher modulus of wind speed at the specified coordinates.

Execution of the 'genRelief()' function requires the MATLAB *Curve Fitting Toolbox* and *Image Processing Toolbox* packages. To generate the relief matrix, you apply an averaging filter using the 'special()' function and specify the number of entry points.

Regarding the generation of the roughness matrix, a roughness distribution law is established that depends on the type of terrain. For the distribution of surface roughness lengths, the Lettau equation [16] was adapted, which calculates the components of the roughness matrix. (Z_0) depending on the reference roughness length (z_0), the relief matrix (Z) and the average altitude (z_{med}). Esta ecuación se puede modificar para crear distribuciones de rugosidad más complejas.

In addition, the function 'getDistTrapz()' is adapted to take into

account the difference in level between the generating turbines and affected by the wake. The equations of some wake models, such as the Jensen model, are also modified to incorporate the use of the variable roughness stored in the roughness matrix and recalculate the parameters dependent on it at each calculation point.

V. RESULTS

A series of experiments have been developed to evaluate the correct functioning of the implementations carried out on the CRO-SL algorithm with which it was counted. We proceed to the presentation of the results achieved in a series of experiments, which measure the ability to obtain optimal solutions, the participation of the different substrates in the improvement of solutions, obtaining the best overall solution, the affectionation on the estimated predictions after the incorporation of relief / variable roughness on the ground and the execution times of the new model of wake implemented, defined Ainslie.

Results of the optimization process

In order to verify the degree of improvement suffered by the CRO-SL algorithm after the incorporation of 3 new substrates (IPx, PSO and WOA) a comparison is made between the results obtained by the standard version the CRO-SL that had only 4 substrates and the Basic Genetic Algorithm (BGA)

Table I records the values of annual energy generated, estimated by the different algorithms and wake models, using the matrix of real wind vectors established, size 2x8760, which collects the speed and direction of the wind during all hours of a year at a point in the province of Badajoz (Spain).

TABLE I
ANNUAL ENERGY PRODUCTION FROM OPTIMIZED DISTRIBUTIONS

ALGORITMO	WAKE MODEL	ENERGY (kWh)
BGA	Jensen	$7.7818 \cdot 10^7$
	Larsen	$7.7527 \cdot 10^7$
	Ainslie	$7.7896 \cdot 10^7$
CRO-SL (4 sustratos)	Jensen	$7.7722 \cdot 10^7$
	Larsen	$7.4619 \cdot 10^7$
	Ainslie	$7.7882 \cdot 10^7$
CRO-SL (7 sustratos)	Jensen	$7.7856 \cdot 10^7$
	Larsen	$7.6144 \cdot 10^7$
	Ainslie	$7.8004 \cdot 10^7$

The highest values of energy production for each wake model are achieved by the CRO-SL algorithm to which the new substrates based on the IPx operator and the PSO and WOA algorithms have been incorporated, making a total computation of 7 substrates. After him, the BGA is positioned as the second best algorithm when it comes to optimizing wind turbine distributions, well above the original CRO-SL algorithm, which had only 4 substrates (BLX α , MPx, 2Px y GM), which gets the worst solutions.

Substrate participation in optimal solutions

Table II shows the sum of the generations in which each of the substrates has generated the best of the larvae, with the results of the 3 incorporated substrates emphasized. Both the CRO-SL with

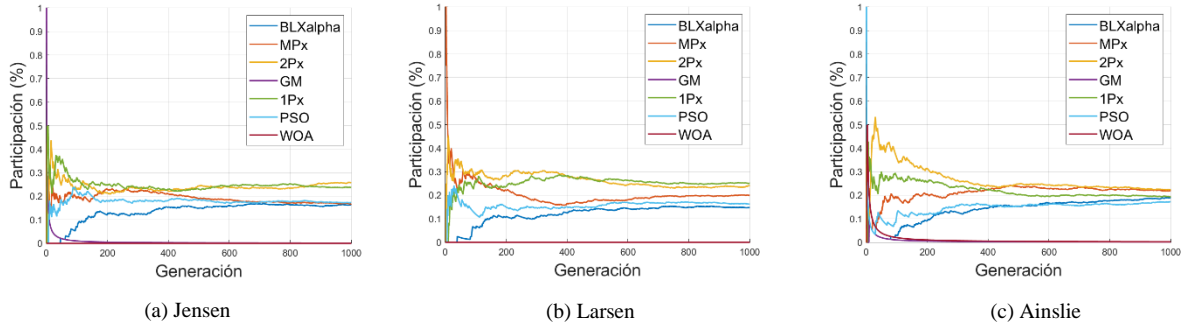


Fig. 6. Evolution of participation of CRO-SL substrates (7 substrates).

four substrates and the one that incorporated seven substrates were executed 1000 generations for each of the selected wake models, giving a total of 3000 generations between all of them.

It is important to note that in the breakdown of participation results for each of the wake models (1000 generations each) they showed very similar participation percentages for all of them. This demonstrates a slight impact of the wake model used in the execution of the substrates within the algorithm and, therefore, allowing the sum of participation without losing relevant information about the operation.

TABLE II

Nº. OF GENERATIONS IN WHICH EVERY SUBSTRATE DOMINATES

SUBSTRATE	CRO-SL (4 SUBSTRATES)	CRO-SL (7 SUBSTRATES)
BLX α	1115	499
MPx	837	586
2Px	925	721
GM	133	2
1Px	-	683
PSO	-	507
WOA	-	2
Nº. Gen.	3000	3000

According to the results obtained, the most important substrates belong to crossing operators, among which is the 1Px operator incorporated in this study, with a participation in 683 generations. Similarly, another substrate that acquires considerable relevance is the substrate based on the PSO algorithm, which was added to the structure along with 1Px, and which achieves the best results in 507 generations, in the line of other substrates such as MPx (586) or BLX α (499). The last of the added substrates, the WOA, is separated in the background getting the best larvae only in 2 generations. Figure 6 shows the evolution of the participation of the different substrates of the modified CRO-SL, throughout the 1000 generations established for the different wake models chosen.

Cartesian product of distributions by model

Since the best solutions were achieved by the CRO-SL algorithm of 7 substrates, we proceed to the realization of the cartesian product of solutions using the 3 distributions obtained after the optimization process carried out during 1000 generations for each of the wake models.

It is essential to consider the average percentage of

performance of each of the solutions achieved according to the following equation:

$$Ren(\%) = \frac{E_{gen} - E_{min}}{E_{max} - E_{min}} \cdot 100 \quad (2)$$

This allows a solution not to be excessively harmed by the minimum prediction margins established by a wake model, such as the Larsen wake model that tends to greatly penalize the overlap of contrails produced by wind turbines. Table III shows the minimum and maximum values that wake models can predict for a 30x30 terrain ratio configuration with 30 wind turbines positioned.

TABLE III

MAXIMUM AND MINIMUM ENERGY PREDICTED BY EACH WAKE MODEL

WAKE MODEL	MINIMUM ENERGY (kWh)	MAXIMUM ENERGY (kWh)
Jensen	$4.0803 \cdot 10^7$	$7.8064 \cdot 10^7$
Larsen	$9.5351 \cdot 10^6$	$7.8064 \cdot 10^7$
Ainslie	$7.1726 \cdot 10^7$	$7.8064 \cdot 10^7$

Table IV can examine the yields obtained by the different distributions obtained as a result of the execution of the optimization process of the CRO-SL algorithm after the execution of the Cartesian product of the solutions with each of the wake models. The average yield marks that the best of the solutions was obtained by the Jensen wake model with a yield of 95.88%, as emphasized in the average table.

TABLE IV

CARTESIAN PRODUCT PERFORMANCE OF SOLUTIONS WITH WAKE MODELS (PERFORMANCE, %)

SOLUTION MODEL	JENSEN	LARSEN	AINSLIE
JENSEN	99.44%	96.85%	92.26%
LARSEN	91.45%	97.20%	91.41%
AINSLIE	96.27%	86.99%	99.05%
AVERAGE	95.88%	93.68%	94.24%

Relief impairment and changes in roughness

Result of the comparison of altitude changes:

To carry out the validation of the generating function of the slope matrix and its correct implementation in the fitness function when predicting the energy generated in each distribution, a variation margin between $[0, 10]$ m has been established. The relief and contour maps used for the experiments can be seen in Figure 7.

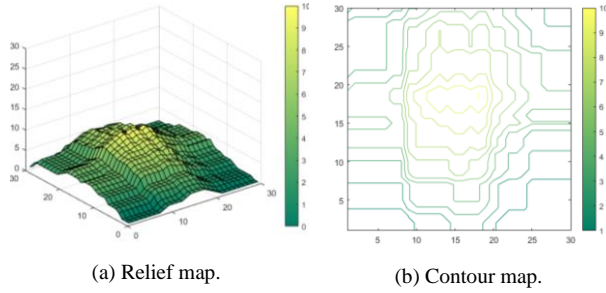


Fig. 7. Relief and terrain contour maps.

Table V marks the differences produced between the optimal distributions obtained by each stele model using the different metaheuristic algorithms proposed. The Ainslie wake model will be used to analyze each of the solutions obtained during the optimization process, due to its high volatility in the prediction of results, which helps to denote the difference in the face of altitude changes as smooth as the established one.

TABLE V

COMPARISON OF ENERGY ESTIMATED BY THE WAKE MODELS ACCORDING TO THE TERRAIN RELIEF TYPOLOGY (AINSLIE)

ALGORITHM	SOLUTION	RELIEF (kWh)	NO RELIEF (kWh)
BGA	Jensen	$7.7737 \cdot 10^7$	$7.7736 \cdot 10^7$
	Larsen	$7.7499 \cdot 10^7$	$7.7736 \cdot 10^7$
	Ainslie	$7.7935 \cdot 10^7$	$7.7896 \cdot 10^7$
CRO-SL (4 substrates)	Jensen	$7.7658 \cdot 10^7$	$7.7656 \cdot 10^7$
	Larsen	$7.7184 \cdot 10^7$	$7.7182 \cdot 10^7$
	Ainslie	$7.7925 \cdot 10^7$	$7.7882 \cdot 10^7$
CRO-SL (7 substrates)	Jensen	$7.7829 \cdot 10^7$	$7.7828 \cdot 10^7$
	Larsen	$7.7526 \cdot 10^7$	$7.7524 \cdot 10^7$
	Ainslie	$7.8005 \cdot 10^7$	$7.8004 \cdot 10^7$

These differences are between a range of values around $10^3 - 10^4$ kWh depending on the distribution selected to perform the calculation of the fitness function with the Ainslie wake model and the turbine distribution used.

Results of the comparison of roughness changes:

Depending on the set relief matrix, a roughness matrix is generated (Z_0) that gives a roughness length value to each terrain coordinate, affecting the results obtained by some wake models that make use of this parameter such as the Jensen model, which will be used to analyze all the optimal solutions obtained after the execution of the metaheuristic algorithms with each wake model and compare them later. In Figure 8 you can see the surface and contour maps that record the different values of the matrix.

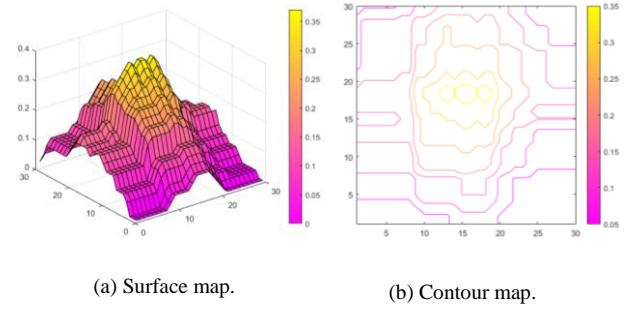


Fig. 8. Surface and contour maps of roughness.

Unlike altitude variations due to unevenness, the implementation of the generation of established surface roughness length causes very noticeable changes in most of the cases analyzed. The difference between the variable and constant roughness values is in magnitudes between $10^5 - 10^6$ kWh, which denotes a large impact on the energy estimated by the Jensen model.

TABLE VI

COMPARISON OF ENERGY ESTIMATED BY WAKE MODELS ACCORDING TO THE TYPE OF ROUGHNESS OF THE TERRAIN (JENSEN)

ALGORITHM	SOLUTION	VARIABLE (kWh)	CONSTANT (kWh)
BGA	Jensen	$7.5866 \cdot 10^7$	$7.7818 \cdot 10^7$
	Larsen	$7.7256 \cdot 10^7$	$7.7444 \cdot 10^7$
	Ainslie	$7.5477 \cdot 10^7$	$7.7678 \cdot 10^7$
CRO-SL (4 substrates)	Jensen	$7.7023 \cdot 10^7$	$7.7722 \cdot 10^7$
	Larsen	$7.6644 \cdot 10^7$	$7.7244 \cdot 10^7$
	Ainslie	$7.7422 \cdot 10^7$	$7.7547 \cdot 10^7$
CRO-SL (7 substrates)	Jensen	$7.7547 \cdot 10^7$	$7.7856 \cdot 10^7$
	Larsen	$7.6153 \cdot 10^7$	$7.7722 \cdot 10^7$
	Ainslie	$7.7499 \cdot 10^7$	$7.7606 \cdot 10^7$

Ainslie wake model runtimes

As shown in the results obtained in Table I, the incorporated Ainslie model describes significantly higher energy generation predictions than the Jensen and Larsen models. This causes high volatility between the bounded values between the maximum and minimum prediction limit.

TABLE VII

WAKE MODELS RUNTIMES

WAKE MODEL	RUNTIME 20 ITERATIONS (s)	AVERAGE RUNTIME (s)
Jensen	25.2272	1.2614
Larsen	32.2178	1.6109
Ainslie	28.3428	1.4171

In Table VII, the execution times of the three algorithms used over 20 iterations have been collected, and the average time value it has taken to execute each iteration. Which leaves the Jensen wake model as the fastest to run with an average speed of 1.2614 s, followed by the Ainslie models with 1.4171 s and Larsen with 1.6109 s.

VI. DISCUSSION

Regarding the analysis of the results achieved after the optimization process with each of the metaheuristic algorithms presented and their different configurations, the BGA algorithm remains the fastest to reach 1000 generations, while the two versions of the CRO-SL algorithm run more slowly. In terms of the results of energy produced, based on the *fitness* function of the best distributions found, it can be concluded that the implementation of the 3 new substrates in the CRO-SL has been successful. These solutions dominate among the set of solutions obtained by each algorithm in the 3 wake models analyzed (Larsen, Ainslie and Jensen). The performance values of this modified version of the CRO-SL significantly improve the results, especially with the Larsen and Ainslie models, although the improvement is less noticeable with the Jensen model.

Importantly, solutions obtained by the 7-substrate CRO-SL algorithm show more pronounced and faster growth compared to solutions obtained with 4 substrates. This means that the average values of the solutions achieved with 7 substrates are closer to the maximum value achieved from the first generations, while, with 4 substrates, it is necessary to advance in more advanced generations to achieve similar results.

It is noted that Jensen's wake model achieved the best fitness value in an earlier generation than the other models. This could be due to a more efficient performance of the PSO substrate, whose hyperparameters were optimized using this model. Larsen's and Ainslie's models got their best solutions at a later generational stage, when the solutions were more established. The use of the Ainslie and Larsen wake models requires the application of multiple implicit equations to obtain the incident velocity at points affected by the contrails. This results in longer calculation times compared to simpler models such as the Jensen wake model. In terms of investment time for the calculation of solutions, the model proposed by Ainslie is positioned as an intermediate alternative between the models of Jensen and Larsen.

Regarding the participation of substrates over generations, it is detected that certain substrates have a more noticeable impact in the first iterations, when the average fitness value of the coral is low. This is because coral larvae are easily replaceable at this stage. As the process evolves, the participation of the substrates stabilizes, and the fewer effective ones are cancelled, showing a more uniform behaviour between the remaining substrates.

Incorporating unevenness in the location of wind turbines implies an increase in the distance between them and in the required land surface. This reduces the speed deficit suffered and increases the wind speed in the affected turbines. Although slopes have a slight impact on energy production, it has been shown that the adaptation of distance calculation using the fitness function is successful.

The use of altitude maps in the terrain increases the radial distance of the wake cone, reducing the deficit effect of turbulence on the speed of the incident wind and increasing the energy generated from the wind turbine array. The implementation of roughness matrices with the use of a roughness length distribution law can suppose a considerable variation in the results of energy prediction generated. However, as already mentioned, this implementation is limited, since it is only compatible with models based on the Jensen wake model and that consider this parameter in the calculation of the internal velocities of the turbulent wake, such as the Park2 model [17].

VII. CONCLUSION

In this study, the CRO-SL algorithm was used to optimize the placement of wind turbines on land for a wind farm. The land was discretized in a 30x30 grid with 30 turbines. Three new substrates based on the 1Px operator, PSO algorithm and WOA were added to improve results. A new model of Ainslie wake was incorporated into the existing library and terrains with non-uniform altitude and different lengths of surface roughness were considered.

The evolution of the fitness function showed a smooth and continuous growth in the BGA algorithm, while in the CRO-SL, especially with the new substrates, a more staggered growth was observed due to periods of stagnation and abrupt changes in fitness values. The inclusion of the new substrates smoothed the growth curve and accelerated the generation of valid solutions, which resulted in an increase in the energy generated.

The addition of 1Px and PSO substrates was shown to improve solutions and increase energy production. However, the WOA substrate had a limited and practically residual impact. Despite this, the implementation of the new substrates was beneficial overall.

The function generating reliefs and roughness worked correctly, although its applicability is limited to certain wake models that consider the roughness length parameter () and require large slopes to show significant changes in the predictions of energy generated in offshore wind farms. z_0

This study addresses several targets of the Sustainable Development Goals (SDGs), especially those included in SDG 7, fulfilling a total of 4 of its 5 specific targets (SDG 7.1, SDG 7.2, SDG 7.3 and SDG 7.b). In addition, it also has a more collateral impact on SDGs 8 and 12, relating in certain aspects to SDG 8.4 and SDG 12.2.

For future studies, several areas of research are suggested that could improve the optimization of wind turbine positioning in different contexts:

- Consider the slip angle in fitness function calculations.
- Explore automated linking of metaheuristic algorithms with more accurate CFD analysis tools.
- Identify parameters that allow the design of new algorithms and constraints that reduce the space for finding solutions.
- Investigate the use of wind turbines in defence applications, such as curbing cyclones or tornadoes, expanding the scope of turbine placement beyond power generation.
- Evaluate the impact of turbine failures and downtime on turbine distribution efficiency, comparing it to other configurations and better understanding a wind farm's performance in outage situations.
- Perform mathematical modelling of the contrails generated by wind turbines on Mars, exploring their application in energy generation on the planet and their relevance for space exploration and colonization.

REFERENCES

- [1] IRENA. (2023). Renewable capacity statistics 2023. Abu Dhabi.
- [2] Orús, A. (March of 2023). Energía Eólica: Generación por CC.AA en España en 2022. *Energía Eólica: Generación por CC.AA en España en 2022*. Obtenido de <https://es.statista.com/estadisticas/1004140/generacion-eolica-por-region-en-espana/>

- [3] Salcedo-Sanz, S., Del Ser, J., Gil-López, S., Landa-Torres, I., Portilla-Figueras, J.A.: The Coral Reefs Optimization Algorithm: A new metaheuristic algorithm for hard optimization problems. In: Proc. of the 15th International Conference on Applied Stochastic Models and Data Analysis (ASMDA), Mataró, Barcelona (2013)
- [4] Salcedo-Sanz, S. (October of 2016). A review on the coral reefs optimization algorithm: New development lines and current applications. *Progress in Artificial Intelligence*, 6. doi:10.1007/s13748-016-0104-2
- [5] A. Asghari and M. K. Sohrabi, "Multiobjective Edge Server Placement in Mobile-Edge Computing Using a Combination of Multiagent Deep Q-Network and Coral Reefs Optimization," in *IEEE Internet of Things Journal*, vol. 9, no. 18, pp. 17503-17512, 15 Sept.15, 2022, doi: 10.1109/JIOT.2022.3161950.
- [6] Tsai, CW., Chang, WY., Wang, YC. et al. A high-performance parallel coral reef optimization for data clustering. *Soft Comput* 23, 9327–9340 (2019). <https://doi.org/10.1007/s00500-019-03950-3>
- [7] Salcedo-Sanz, S., Gallo-Marazuela, D., Pastor-Sánchez, A., Carro-Calvo, L., Portilla-Figueras, A., & Prieto, L. (2014). Offshore wind farm design with the Coral Reefs Optimization algorithm. *Renewable Energy*, 63, 109-115. doi:<https://doi.org/10.1016/j.renene.2013.09.004>
- [8] Salcedo-Sanz, S., Pastor-Sánchez, A., Ser, J. D., Prieto, L., & Geem, Z. W. (2015). A Coral Reefs Optimization algorithm with Harmony Search operators for accurate wind speed prediction. *Renewable Energy*, 75, 93-101. doi:<https://doi.org/10.1016/j.renene.2014.09.027>
- [9] Salcedo-Sanz, Sancho & Pastor-Sánchez, Álvaro & Portilla-Figueras, Antonio & Prieto, Luis. (2015). Effective multi-objective optimization with the coral reefs optimization algorithm. *Engineering Optimization*. 1-19. 10.1080/0305215X.2015.1078139.
- [10] S. Salcedo-Sanz, C. Camacho-Gómez, D. Molina and F. Herrera, "A coral reefs optimization algorithm with substrate layers and local search for large scale global optimization," 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 2016, pp. 3574-3581, doi: 10.1109/CEC.2016.7744242.
- [11] Ainslie, J. F. (1988). Calculating the flowfield in the wake of wind turbines. *Journal of Wind Engineering and Industrial Aerodynamics*, 27, 213-224. doi:[https://doi.org/10.1016/0167-6105\(88\)90037-2](https://doi.org/10.1016/0167-6105(88)90037-2)
- [12] Romero Barrera, A. J. (22 de Mayo de 2023). *DISEÑO DE ALGORITMOS METAHEURÍSTICOS PARA EL PROBLEMA DE COLOCACIÓN DE TURBINAS EÓLICAS SOBRE EL TERRENO*. Trabajo Fin de Grado, Universidad Rey Juan Carlos, Escuela de Ingeniería de Fuenlabrada, Fuenlabrada. Obtenido de <https://burjcdigital.urjc.es/handle/10115/22188>
- [13] Rehman, S., Khan, S., & Alhems, L. (January de 2020). The effect of acceleration coefficients in Particle Swarm Optimization algorithm with application to wind farm layout design. *FME Transactions*, 48, 922-930. doi:10.5937/fme2004922R
- [14] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, 4, págs. 1942-1948 vol.4. doi:10.1109/ICNN.1995.488968
- [15] Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, 95, 51-67. doi:<https://doi.org/10.1016/j.advengsoft.2016.01.008>
- [16] Gaona, G., Balcázar, L., & Jaime, I. (2015, April). Estimación de longitud de rugosidad aerodinámica a partir de imágenes MODIS y LandSat.
- [17] Rathmann, O. S., Hansen, B. O., Hansen, K. S., Mortensen, N. G., & Murcia Leon, J. P. (2018). *The Park2 Wake Model - Documentation and Validation*.