

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.

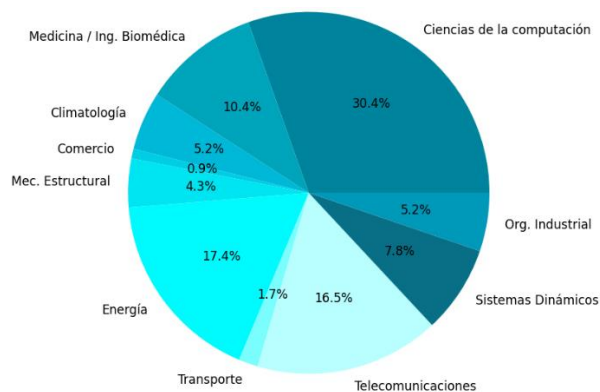


Fig. 1. Campos de investigación que hicieron uso del CRO en sus publicaciones.

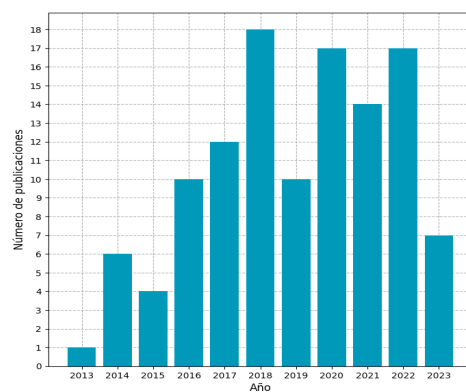


Fig. 2. Evolución del número de publicaciones por año relacionadas con el CRO desde 2013 hasta junio de 2023.

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.

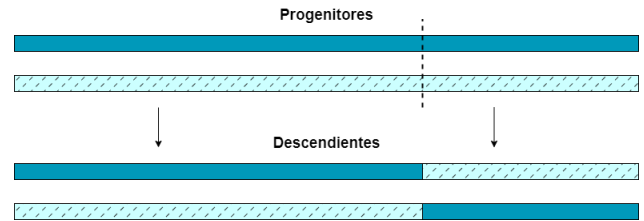


Fig. 3. Esquema de funcionamiento del operador 1Px.

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.

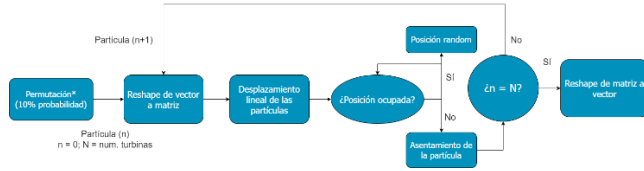


Fig. 4. Diagrama de flujo del *Particle Swarm Optimization* (PSO).

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}).

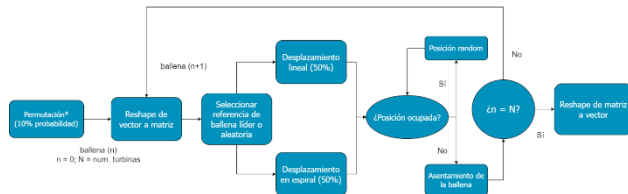


Fig. 5. Diagrama de flujo del *Whale Optimization Algorithm* (WOA).

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

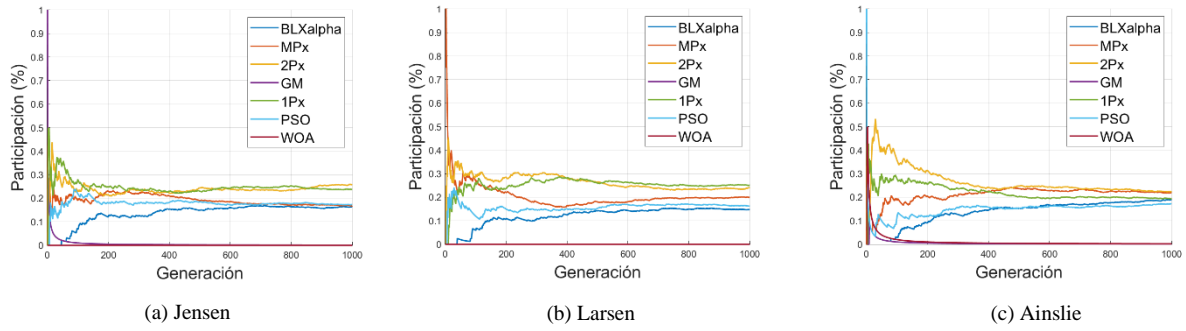


Fig. 6. Evolución de participación de los sustratos del CRO-SL (7 sustratos).

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.

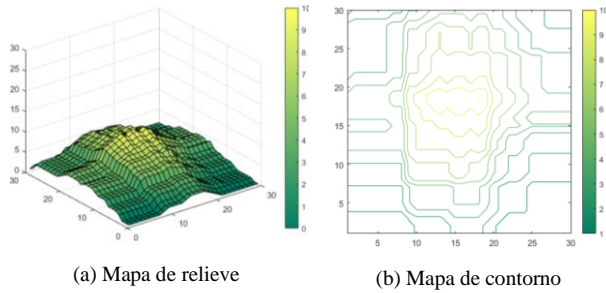


Fig. 7. Mapas de relieve y contorno del terreno.

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.

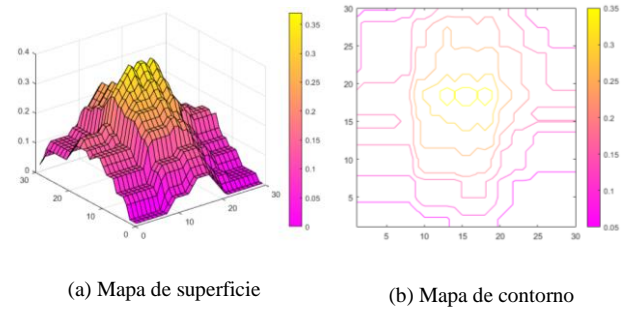


Fig. 8. Mapas de superficie y contorno de la rugosidad.

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*