

Análisis de algoritmos evolutivos contemporáneos en la optimización de distribución de parques eólicos

Eduardo H. Haro¹, Alfonso Ramos-Michel¹, Jorge Arturo Pelayo López¹

¹Departamento de Ingenierías, Universidad de Guadalajara CUCSUR, Av. Independencia Nacional No. 151, Autlán de Navarro, Jalisco, México

Resumen: Los parques eólicos representan una de las principales apuestas sustentables para el futuro. Sin embargo, uno de sus obstáculos es lograr una distribución óptima. Este problema es conocido como optimización de distribución de parques eólicos, y su solución garantiza la mayor producción energética posible al menor costo de instalación. Para ello, se han utilizado diversas herramientas computacionales, siendo el algoritmo de evolución diferencial una de las mejores alternativas. Por esta razón, en el presente trabajo se presenta un análisis comparativo entre diversas variantes de dicho algoritmo, propuestas durante la última década. Con el fin de comprender las ventajas y limitaciones de estas variantes, se propone la optimización de un problema de distribución de parques eólicos mediante un caso experimental. Los resultados muestran las ventajas de estos algoritmos en términos de precisión y costo computacional, al mismo tiempo que se destacan las principales características de estos algoritmos que contribuyen específicamente a la distribución de parques eólicos.

Palabras clave: evolución diferencial, optimización, parques eólicos, algoritmos evolutivos.

1. Introducción

Los sistemas de energía sustentable son uno de los pilares científicos en la actualidad debido a sus implicaciones para nuestro desarrollo futuro, el cual, depende en gran medida de nuestra capacidad para cubrir la demanda energética mundial sin afectar al medio ambiente. En ese sentido, dentro de las alternativas energéticas, la energía eólica se posiciona como uno de los mejores métodos en términos de producción y costo [1]. Desafortunadamente, este método requiere de grandes extensiones territoriales dónde colocar las decenas de turbinas eólicas y dicho territorio requiere a su vez condiciones geográficas específicas para hacer viable la construcción del parque, lo que provoca que no puedan desarrollarse en cualquier lugar [1]. Por ello, es imperativo distribuir lo mejor posible dichas turbinas para obtener el mayor beneficio energético del parque, y este problema de distribución se conoce como Optimización de Distribución de Parques Eólicos – Wind Farm Layout Optimization (WFLO, por sus siglas en inglés) [2].

Existen diversas maneras de afrontar el problema WFLO, no obstante, las mejores alternativas siempre han sido los modelos computacionales basados en sistemas de Inteligencia Artificial (IA) [3]. Precisamente, los sistemas de IA más aplicados a la solución del WFLO han sido los algoritmos metaheurísticos, los cuáles se caracterizan entre otras cosas, por tratarse de modelos de optimización cuyo

Recibido: mayo 15, 2025

Aceptado: julio 24, 2025

autor de correspondencia:

eduardo.hernandezh@academicos.udg.mx

<https://doi.org/10.66482/tp4sdy21>

comportamiento iterativo se inspira en procesos biológicos tales como métodos de caza de algunos animales, métodos de reproducción, o incluso fenómenos naturales [4]. Si bien, los algoritmos metaheurísticos existen desde los años 60s, su aplicación en el problema WFLO se dio durante la década de los 90s y hoy en día sigue siendo uno de los temas más importantes en la optimización de sistemas de energía. Ahora bien, dentro de los algoritmos utilizados para esta optimización uno de los que más destaca es el algoritmo Evolución Diferencial - Differential Evolution (DE, por sus siglas en inglés). Este algoritmo fue propuesto en 1997 por Storn y Price [5], y ha generado una gran cantidad de variantes debido a su diseño simple y eficiente desempeño, por lo que, a día de hoy el DE es uno de los pilares de la optimización metaheurística.

El DE y sus variantes han demostrado un gran desempeño en la solución del WFLO, lo cual, es debido en gran medida a su adaptabilidad en los espacios de búsqueda. Mientras otras metaheurísticas como las bio-inspiradas funcionan a base de modelos matemáticos complejos que suelen verse afectados al enfrentar problemas restrictivos, los algoritmos evolutivos como el DE usan esto a su favor adaptando el movimiento de sus partículas al entorno de búsqueda, lo que hace que el DE y sus variantes tengan una ventaja natural sobre los bio-inspirados en problemas como el WFLO.

Precisamente, son estas características las que han propiciado que algunas de estas variantes hayan sido aplicadas a modelos de WFLO como el Evolución Diferencial basado en memoria histórica adaptada - Success-History based Adaptive Differential Evolution (SHADE, por sus siglas en inglés) o variantes ofreciendo resultados interesantes. No obstante, es importante mencionar que la gran mayoría de trabajos relacionados tienden a modificar al algoritmo de tal modo que se adapte de forma muy específica al problema, como es el caso del SHADE adaptado mediante una búsqueda multi local - Multi local Search SHADE (MS-SHADE, por sus siglas en inglés) [6]. En principio, no hay problema con dichos cambios pues los resultados son competentes, sin embargo, se trata de algoritmos tan especializados en un problema particular que involucra rehacer adaptaciones para otros problemas, por lo que el algoritmo base no demuestra ser eficiente en sí mismo.

Sin embargo, esta creciente publicación de nuevas variantes del DE, así como el desarrollo de modelos WFLO cada vez más sofisticados, crean la necesidad de conocer las ventajas y limitaciones de este algoritmo en la actualidad para la solución de dicho problema. Si bien, existen algunos trabajos que recopilan los avances más importantes relacionados, no existe hasta la fecha de la creación de este artículo un trabajo que analice exclusivamente el impacto del DE y sus variantes contemporáneas [7]. Por ende, para solventar esta carencia antes mencionada, en este trabajo de investigación se presenta un estudio comparativo entre el DE y ocho de sus variantes más importantes de la última década. Para ello, se analiza el desempeño de dichos algoritmos en un problema WFLO que consta de un caso de

estudio de eclipse completo. Respecto a los algoritmos a comparar, se toma a consideración únicamente variantes del DE en su forma base, es decir, para ofrecer una experimentación justa solo se comparan algoritmos no adaptados específicamente al problema WFLO. Por otra parte, el término eclipse hace referencia a un fenómeno conocido como efecto estela, el cual ocurre cuando dos turbinas dentro de un parque eólico se encuentran lo suficientemente próximas como para que la primera genere una estela que afecte el rendimiento operacional de la segunda [8]. Los resultados obtenidos muestran las grandes capacidades operacionales del DE y sus variantes. Sin embargo, una variante en particular demostró ser superior al resto de competidores en términos de estabilidad, precisión, costo computacional y tasa de convergencia.

El resto del artículo se estructura de la siguiente forma, la Sección 2 explica en mayor detalle el problema WFLO, la estructura del DE y sus variantes utilizadas en este estudio son analizadas en la Sección 3, en la sección 4 se analiza la forma de optimizar el WFLO, mientras que los resultados computacionales se presentan en la Sección 5, finalmente, la Sección 6 discute las conclusiones y posibles trabajos futuros.

2. El problema de distribución de parques eólicos

Tal como se explicó anteriormente, en los parques eólicos existe un fenómeno conocido como efecto estela, el cual consiste en la creación de perturbación eólica provocada por una turbina hacia una segunda turbina. Esto a su vez genera que la turbina afectada presente un rendimiento operacional inestable y hasta indeseable. Es por ello que, para conocer el impacto y características correctas de un efecto estela en una turbina se han diseñado diversos modelos matemáticos que busca predecir este comportamiento. En este trabajo de investigación se abordará el modelo de Jensen propuesto en 1983 [9]. Dicho modelo considera una velocidad de viento constante V_0 , la cual, al impactar sobre una turbina genera un efecto estela con diámetro inicial D_0 y vertiente k que depende directamente de la distancia x a la que se encuentre la segunda turbina. Una vez que la estela impacta sobre esta segunda turbina, la velocidad del viento cambia a $V(x)$ afectando la turbina en un área $D(x)$; este comportamiento se aprecia en la Figura 1.

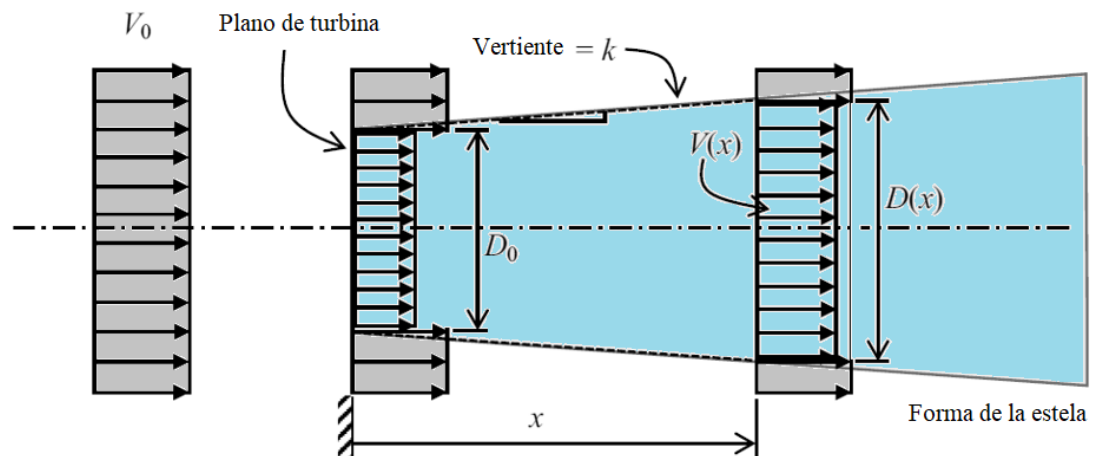


Figura 1. Modelo de decaimiento del efecto estela.

Con base en el modelo de Jensen, la velocidad del viento se describe como:

$$1 - \frac{V(x)}{V_0} = \alpha \left(\frac{D_0^2}{D(x)^2} \right) \quad (1)$$

donde, el diámetro de la estela en la turbina afectada $D(x)$ se calcula como se muestra en la Ecuación 2. En dicha ecuación, k establece un valor constante que define la altura del poste de la turbina y la rugosidad del terreno del parque eólico. Por otra parte, al retornar a la Ecuación 1, otro valor a considerar es α , el cual establece el déficit de velocidad en el área de la estela y se modela como se muestra en la Ecuación 3, en donde C_T corresponde al coeficiente de empuje de la turbina.

$$D(x) = D_0 + 2kx \quad (2)$$

$$\alpha = 1 - \sqrt{1 - C_T} \quad (3)$$

Hasta este momento solo se ha considerado el hecho de que el efecto estela se produzca entre dos turbinas, sin embargo, en un parque eólico real este efecto se produce desde múltiples direcciones. Por ello, es importante considerar este efecto de manera dinámica y para tal caso se modela dicha energía como lo muestra la Ecuación 4.

$$\left(1 - \frac{V}{V_0}\right)^2 = \sum_{i=1}^{N_{EC}} \left(1 - \frac{V(x_i)}{V_0}\right)^2 \quad (4)$$

En esta ecuación, N_{EC} establece el número de turbinas que eclipsan a la turbina a analizar, siendo i precisamente el número de turbina actual y $V(x_i)$ su velocidad de viento correspondiente. Con base en todo lo anterior, se puede calcular la energía total producida por el parque eólico de la siguiente manera:

$$P_t = \sum_{i=1}^{N_{WT}} \frac{1}{2} \eta_i \rho_{air} A_i V_i^3 \quad (5)$$

donde, N_{WT} es el número total de turbinas en el parque eólico, mientras que η_i y ρ_{air} corresponden a la eficiencia de la turbina y la densidad del aire, respectivamente. Por otra parte, la velocidad del viento V_i^3 se establece en m/s mientras que el área del disco de la turbina A_i , así como su intersección con otra turbina se modela mediante la Ecuación 6.

$$A_{i,j} = r_i^2 \cos^{-1} \left(\frac{d_{i,j}^2 + r_i^2 - r_j^2}{2d_{i,j}r_i} \right) + r_j^2 \cos^{-1} \left(\frac{d_{i,j}^2 + r_j^2 - r_i^2}{2d_{i,j}r_j} \right) - \frac{1}{2} \sqrt{(-d_{i,j} + r_i + r_j)(d_{i,j} - r_i + r_j)(-d_{i,j} + r_i - r_j)(d_{i,j} + r_i + r_j)} \quad (6)$$

En esta ecuación, $d_{i,j}$ corresponde a la distancia entre los centros de ambas turbinas, mientras que r_i y r_j son los radios de las estelas de las turbinas i y j , respectivamente. La Figura 2 simplifica la explicación de la Ecuación 6, en donde la turbina j es parcialmente eclipsada por la turbina i mediante el efecto estela. Finalmente, cabe mencionar que esta intersección de ambas turbinas está dada por $A_{i,j}$ que a su vez corresponde al diámetro del eclipse.

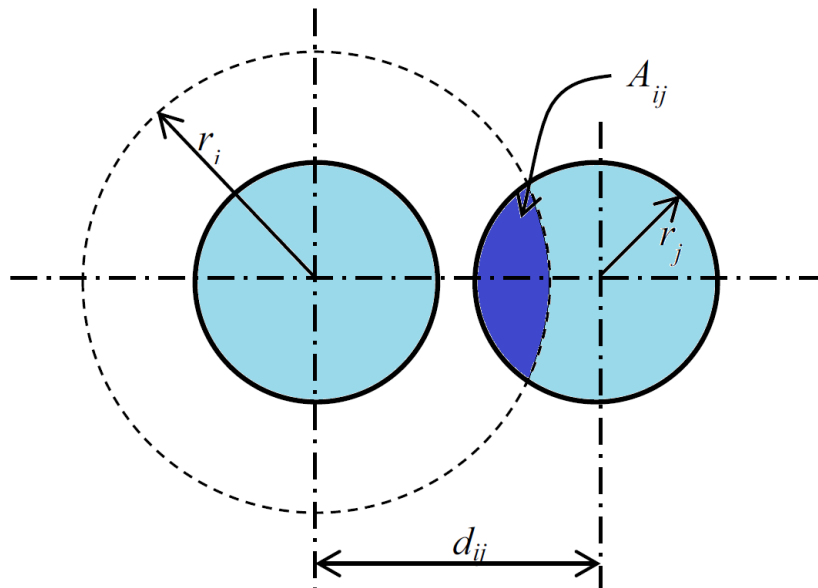


Figura 2. Intersección entre dos turbinas.

3. Algoritmo DE

El algoritmo DE fue propuesto en 1997 por los investigadores Storn y Price con el propósito de optimizar funciones generales. No obstante, el diseño simple y buena eficiencia del mismo hicieron que este algoritmo rápidamente se convirtiera en uno de los pilares del cómputo evolutivo [10]. En términos sencillos, el DE se compone de tres partes fundamentales, la mutación, el cruzamiento y la selección. Sin embargo, como cualquier otro algoritmo metaheurístico, el DE comienza con una etapa de inicialización en donde se establecen los parámetros y creación de la población inicial de soluciones X_i^G , en donde i establece la solución actual dentro de un conjunto de soluciones desde 1 hasta N , mientras que G corresponde a la señalización de la población general. La definición de esta población inicial se da mediante la siguiente expresión:

$$X^G = L_B + (U_B - L_B) \times rand(d, N) \quad (7)$$

donde, L_B y U_B corresponden a los límites inferior y superior de cada individuo, respectivamente, mientras que d establece las dimensiones o variables de cada individuo. Una vez inicializada y evaluada toda la población inicial en la respectiva función objetivo, el DE da paso a su primera etapa específica antes mencionada que es la mutación. Existen una enorme cantidad de métodos de mutación para el DE, esto debido a que dicha etapa está directamente relacionada con la exploración del algoritmo, por lo que su constante mejoramiento ha sido uno de los pilares de las variantes del DE [11]. Sin embargo, en este trabajo se analizará la mutación original del DE propuesta por sus respectivos autores y conocida como DE/rand/1. Esta mutación establece la selección aleatoria de tres individuos $(X_{r_1}^G, X_{r_2}^G, X_{r_3}^G)$ dentro de la población actual para posteriormente introducirlos a la ecuación de mutación siguiente:

$$v_i^{G+1} = X_{r_1}^G + F \cdot (X_{r_2}^G - X_{r_3}^G) \quad (8)$$

donde, v_i^{G+1} es el vector mutado resultante de la mutación de los tres individuos seleccionados aleatoriamente, mientras que F es un factor de escalamiento constante comprendido entre 0.5 y 0.8. Una vez conseguido el vector mutado, este se usa para conseguir el vector cruzado en la segunda etapa del DE que es precisamente el cruzamiento. Al igual que en la mutación, existen múltiples métodos de cruzamiento para el DE, esto debido a que está directamente relacionado con la explotación del algoritmo. No obstante, en este trabajo se explicará el cruzamiento binomial, que es el propuesto originalmente para el DE. En este cruzamiento primero se establece una tasa de cruce constante C_r generalmente definida entre 0.1 y 0.3. Posteriormente, el vector cruzado u_i^{G+1} se va creando iterativamente para cada una de sus dimensiones considerando al vector mutado v_i^{G+1} y al individuo actual de la población X_i^G . Para definir cuál de los individuos será utilizado en la respectiva dimensión se utiliza la siguiente ecuación:

$$u_i^{G+1} = u_d^G \begin{cases} v_{i,d}^G \rightarrow d = d_{rand} \\ v_{i,d}^G \rightarrow rand(0,1) \leq C_r \\ X_{i,d}^G \rightarrow Otro\ caso \end{cases} \quad (9)$$

donde, u_d^G es la dimensión actual del vector cruzado u_i^{G+1} , mientras que $v_{i,d}^G$ y $X_{i,d}^G$ corresponden a las dimensiones actuales del vector mutado y el individuo actual, respectivamente. Por otra parte, d_{rand} es un número aleatorio definido entre 1 y d , y el cual debe cambiar en cada iteración. Este vector cruzado corresponde a un nuevo individuo, el cual pasa a la siguiente y última etapa del DE que es la selección. A diferencia de los dos operadores anteriores, la selección es una etapa que no ha sido tan explorada en la literatura, correspondiendo únicamente al 2% de las variantes del DE [12]. En este caso, se utiliza la selección original del algoritmo, la cual es conocida como selección elitista y consiste únicamente en evaluar a u_i^{G+1} en la función objetivo y comparar su valor fitness con el de X_i^G . Al obtener un vencedor, ya sea en términos de minimización o maximización, dicho individuo ocupará el espacio de X_i^G dentro de la población actual. Finalmente, todo este proceso se repite iterativamente hasta completar un cierto número de generaciones o accesos a función, entendiendo a una generación como el hecho de que todos los individuos de la población actual hayan participado en el proceso de la Ecuación 9, mientras que los accesos a función son básicamente el número de evaluaciones en la función objetivo.

4. Optimización del WFLO

Existen múltiples formas de interpretar la distribución de turbinas en un parque eólico en el WFLO, esto incluye la distribución en un campo con restricciones territoriales que complejizan aún más el modelo. No obstante, la mayoría de estos parques son colocados en la costa para aprovechar los vientos marinos, lo cual involucra no considerar restricciones territoriales, que es justo el modelo que se usará en este trabajo [13]. Con base en ello, la función objetivo a optimizar considerará dos puntos de equilibrio, el costo económico de instalación del parque, así como la potencia energética total suministrada por el mismo, buscando así un equilibrio entre ambas partes. Es decir, la función objetivo $f(x)$ buscará la distribución que permita obtener la mayor eficiencia energética (P_t) del parque al menor costo de instalación posible. Dicha función se aprecia a continuación:

$$f(x) = \frac{Costo}{P_t} \quad (10)$$

donde P_t corresponde a la energía total entregada por el parque y previamente descrita en la Ecuación 5, mientras que $Costo$ establece precisamente el costo económico de la instalación del parque y se puede describir de la siguiente forma:

$$Costo = N_T \left(\frac{2}{3} + \frac{1}{3} e^{-0.00174 \times N_T^2} \right) \quad (11)$$

donde N_T denota el número total de turbinas en el parque. Por otra parte, un parque eólico real usualmente requiere una extensión territorial de 2000 m^2 , por lo que para efectos de este experimento computacional se dimensionará dicha extensión a una cuadrícula de 10×10 espacios que simularán una extensión territorial de 200 m^2 cada uno y en donde solo se podrá albergar una turbina eólica, lo cual a su vez permite que el parque entero tenga una disponibilidad máxima para 100 turbinas. Esto se hace con la intención de dar más realismo a la simulación, pues se debe considerar que el diámetro que ocupan las hélices de la turbina es de 5 veces el tamaño del diámetro del rotor, tal como sucede en la mayoría de turbinas reales.

Para comprender las especificaciones de las turbinas a considerar en este trabajo, se presenta la Tabla 1, en donde lo más destacable es la velocidad del viento, la cual será de 12 m/s , que tiende a ser la media en este tipo de instalaciones. Finalmente, para el diseño del parque eólico que como se mencionó anteriormente consistirá de una cuadrícula de 100 espacios, su distribución es la presentada en la Figura 3, donde se puede notar que la distribución es de izquierda a derecha y de abajo hacia arriba. La distribución mostrada en dicha figura es solo representativa por lo que no corresponde a un resultado en particular, mostrando en cada celda del hipotético parque eólico la posición y numeración de cada turbina.

Tabla 1. Características de la turbina eólica.

Característica	Valor
Coefficiente de empuje (C_T)	0.88
Rugosidad de la superficie (h_0)	0.3
Altura del poste (h)	60 m
Diámetro del rotor (D_0)	40 m
Eficiencia de la turbina (η)	40 %
Densidad del aire (ρ_{air})	1.225 kg/m^3
Velocidad del viento (V_0)	12 m/s
Direcciones del viento ($^\circ$)	10, 20, 30, ..., 360

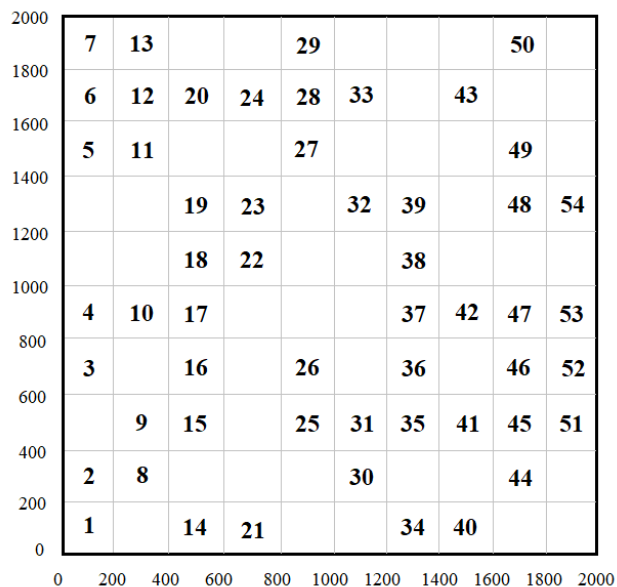


Figura 3. Distribución de las turbinas en el parque eólico.

5. Resultados computacionales

En esta sección, se presentan los resultados obtenidos por el DE y diversas variantes en la optimización del WFLO explicado en las secciones anteriores. Tal como se mencionó en la introducción, el DE se compone de tres etapas fundamentales que son la mutación, el cruzamiento y la selección, así como la inicialización que tienen la gran mayoría de algoritmos metaheurísticos. Con base en eso, los algoritmos variantes seleccionados para competir contra el DE son dos enfocados en nuevos métodos de inicialización, el Evolución diferencial basado en clusterización - Differential Evolution based on Clustering (DEC) [14], y el Evolución diferencial caótico oposicional - Chaotic Oppositional Differential Evolution (CODE) [15]. De igual forma, se proponen otros dos enfocados en nuevas mutaciones, la Mutación basada en población histórica en el evolución diferencial - Historical population based mutation strategy in Differential Evolution (Hip-DE) [16], y el Evolución diferencial con mutación basada en la biología - Differential Evolution with biological based mutation (DE-HeO) [17]; mientras que por el lado del cruzamiento se propone el uso del Evolución diferencial con cruzamiento basado en enlace híbrido - Differential Evolution with Hybrid Linkage Crossover (HLXDE) [18], y el Cruzamiento diferencial epistático basado en grafos cartesianos en ensamble con el evolución diferencial - epistatic Differential Crossover based on Cartesian graph product in ensemble DE (eXEDE) [19]. Finalmente, también se propone el uso de dos variantes del DE enfocadas en un nuevo operador de selección, que son el Evolución diferencial con selección elitista de ruleta - Roulette wheel selection-Elitist-Differential Evolution (ReDE) [20] y el Evolución diferencial con selección basada en configuración paramétrica individual - Differential Evolution with

improved individual-based parameter setting and selection strategy (IDEI) [21]. Todos estos algoritmos compitieron en igualdad de oportunidades, con una población inicial de 100 individuos, 50,000 accesos a función y con 35 ejecuciones independientes.

Respecto a los últimos dos datos mencionados es importante aclarar que existe la posibilidad de considerar como criterio de paro un factor externo a las iteraciones del algoritmo, como lo es un criterio por tiempo de ejecución o por una perturbación externa. Sin embargo, no es lo ideal hacer algo así, debido a que no se le daría a cada algoritmo la oportunidad de demostrar su real desempeño en igualdad de oportunidades. Esto es debido a que algunos algoritmos contienen más acceso a función por iteración que otros, por lo que ejecutar por tiempo o perturbación solo provocaría que algunos métodos logren entregar todo su desempeño y otros no, creando una competencia injusta. Por otra parte, en este tipo de experimentaciones computacionales siempre se consideran 35 ejecuciones independientes debido a que es la cantidad de datos mínimos requeridos por pruebas no paramétricas para ofrecer una estadística correcta.

La parametrización interna de cada algoritmo se basó en la estipulada por sus autores en los respectivos artículos originales, mientras que para el DE se usó un factor de escalamiento de 0.8 y una tasa de cruzamiento de 0.2. Finalmente, cabe mencionar que el desarrollo y ejecución de los algoritmos, así como del WFLO fueron realizados en el entorno de programación de MATLAB en una Notebook (Procesador AMD de 2.7 GHz, 8 GB de RAM) con sistema operativo Windows 10.

Primeramente, los resultados métricos de los algoritmos respecto a la función objetivo de la Ecuación 10 se pueden apreciar en la Tabla 2. Como se puede notar, se consideran cinco métricas, tres relacionadas directamente al valor fitness que son el mejor, el peor y el promedio, siendo este último una ponderación de las 35 ejecuciones independientes. Por otra parte, se consideran también dos métricas no relacionadas con el valor fitness sino con la ejecución general de cada algoritmo, que son la Desviación estándar – Standard Deviation (STD, por sus siglas en inglés) y el tiempo computacional medido en segundos [22]. Respecto al STD, se trata de una métrica que permite analizar la desviación o diferencia general que existe entre cada uno de los 35 resultados de un algoritmo, donde evidentemente lo ideal es tener un STD de cero, lo que implicaría que todos los resultados son exactamente iguales. Mientras que el tiempo computacional se refiere al tiempo requerido por el algoritmo para optimizar el WFLO las 35 veces, por lo que el dato apreciado no es la sumatoria de los 35 tiempos, sino el promedio de ellos.

Con base en lo anterior, se puede ver que, de todos los métodos, el IDEI presenta el mejor desempeño respecto a los resultados mejor y promedio, mientras que el propio DE obtiene dentro de los peores resultados, el mejor. De igual forma, este mismo algoritmo consiguió el mejor tiempo computacional con solo 531.21

segundos de ejecución. No obstante, es lógico que el DE sea el mejor método en este rubro pues todos los demás algoritmos son variantes que agregan etapas al DE original, lo cual inherentemente repercute en un mayor costo computacional respecto al DE. Finalmente, la mejor estabilidad es reportada por el Hip-DE, aunque es importante mencionar que estabilidad no es lo mismo que precisión, pues se puede apreciar que dicho algoritmo es estable pero dentro de resultados alejados del mejor competidor que es el IDEI.

Tabla 2. Resultados métricos de los algoritmos.

Algoritmo	Mejor	Promedio	Peor	STD	Tiempo (s)
DE	0.001455	0.001460	0.001466	1.63E-04	523.21
DEc	0.001459	0.001462	0.001473	2.55E-05	544.58
CODE	0.001452	0.001459	0.001468	3.17E-04	613.04
Hip-DE	0.001463	0.001471	0.001475	1.08E-04	546.11
DE-HeO	0.001459	0.001467	0.001474	1.74E-06	623.05
HLXDE	0.001456	0.001462	0.001469	5.21E-04	601.45
eXEDE	0.001461	0.001470	0.001475	2.06E-05	578.23
ReDE	0.001460	0.001477	0.001483	3.11E-06	591.09
IDEI	0.001451	0.001458	0.001467	1.28E-04	535.37

Otro aspecto crucial a considerar en este tipo de trabajos es la tasa de convergencia de cada algoritmo. La convergencia no es otra cosa más que el ritmo con el que el algoritmo logra ir encontrando mejores resultados a lo largo del proceso iterativo. En ese sentido, lo ideal es encontrar una gráfica que alcance lo más rápido posible el óptimo resultado sin sufrir estancamientos. Para ello, se muestran las convergencias de todos los algoritmos en la Figura 4, donde claramente se puede notar que nuevamente el IDEI (línea azul) corresponde al mejor algoritmo, pues alcanza su óptimo resultado alrededor de los 1000 accesos a función. También se puede notar que el IDEI es seguido por el CODE (línea naranja) y el HLXDE (línea negra), quienes consiguen su mejor resultado en aproximadamente 1100 y 6000 accesos a función, respectivamente. En contraparte, el Hip-DE (línea amarilla) demuestra ser el peor algoritmo en términos de convergencia pues se estanca desde antes de la mitad del proceso iterativo, seguido por el eXEDE (línea roja) y el DEc (línea cian), quienes presentan comportamientos similares.

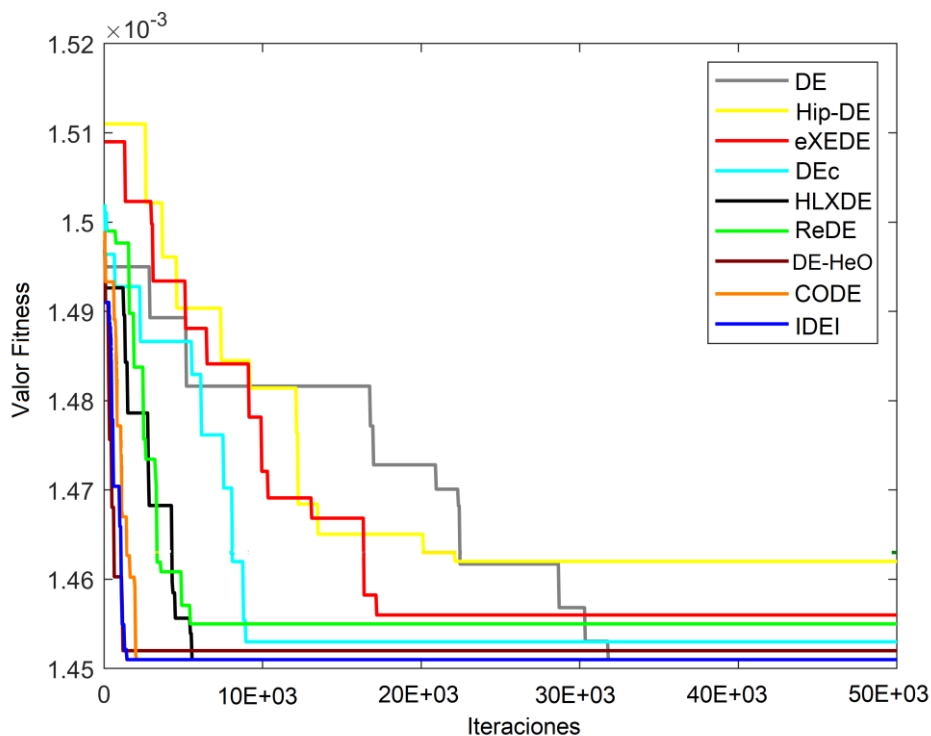


Figura 4. Convergencias de los algoritmos.

Tal como se mencionó anteriormente, el objetivo de la optimización del WFLO es encontrar la distribución óptima de turbinas que permitan reducir los costos de instalación lo más posible, al mismo tiempo que se obtiene el mayor suministro energético. Es por ello que, la Tabla 3 presenta dicha relación para cada uno de los algoritmos considerando el número de turbinas, costo de instalación y energía entregada, esto con base en el mejor resultado obtenido de la Tabla 2. Con base en lo anterior, se puede apreciar que el IDEI propone una distribución de solo 39 turbinas, que corresponde al menor número posible y cuyo costo de instalación representa 29.9216 unidades de costo, mientras que la energía suministrada con esta distribución es de 18,446.80 kW, que evidentemente es la menor de todas.

Por el contrario, el eXEDE propone el mayor número de turbinas con un total de 50, que a su vez representa el costo más alto con 36.0218 unidades de costo, pero con la mayor energía suministrada, 27,336.22 kW. Teniendo en cuenta que ambos algoritmos se ubican en polos opuestos de la distribución óptima se puede calcular que con 44 turbinas se obtiene el balance adecuado. Por ello, al observar la Tabla 3 se puede notar que tanto el DEc como el DE-HeO cumplen con esa condición, con la diferencia de que este último ofrece un mayor suministro energético, por lo que podemos aseverar que, aunque el DE-HeO no se caracteriza por tener las mejores métricas, sí entrega un balance adecuado entre costo y energía.

Tabla 3. Relación costo/energía de los algoritmos.

Algoritmo	Turbinas	Costo de instalación	Energía entregada
DE	41	28.1116	20,156.04 kW
DEc	44	30.0376	22,355.11 kW
CODE	40	27.1892	19,378.05 kW
Hip-DE	48	34.1255	25,176.34 kW
DE-HeO	44	30.1056	22,359.18 kW
HLXDE	43	29.1633	21,889.08 kW
eXEDE	50	36.0218	27,336.22 kW
ReDE	49	35.6614	26,057.09 kW
IDEI	39	26.9216	18,446.80 kW

Con base en la Tabla 3, es importante conocer la distribución propuesta por los algoritmos, que para fines de este trabajo se presentan las distribuciones tanto del IDEI como del DE-HeO en las figuras 5 y 6, respectivamente. Por una parte, el IDEI presenta una distribución de turbinas principalmente colocadas a lo largo del perímetro del parque, con únicamente 8 turbinas en el interior del mismo. Esto es crucial pues cuantas más turbinas haya en el interior del parque, mayor número de turbinas sufrirán del efecto estela explicado anteriormente. Por otra parte, el DE-HeO presenta de igual forma una distribución principalmente perimetral, pero considerando 10 turbinas en el interior del parque. Aunado a esto, el DE-HeO comete el error de quitar una turbina del perímetro donde impacta el viento, lo cual incrementa el efecto estela para las turbinas internas, mientras que el IDEI no comete este error, teniendo todo ese sector cubierto de turbinas que aprovechan todo el viento disponible. Por ende, se puede afirmar que, aunque el DE-HeO ofrece la mayor energía, el IDEI propone la mejor distribución eólica.

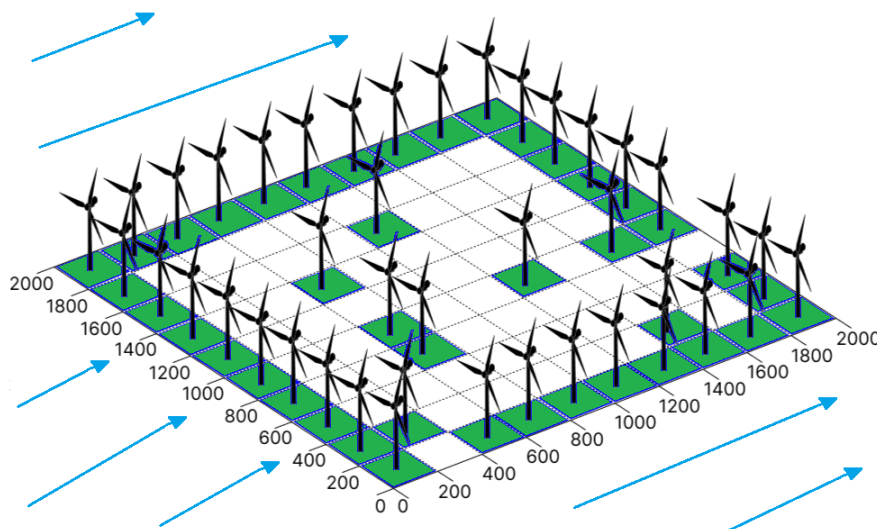


Figura 5. Distribución eólica propuesta por el IDEI.

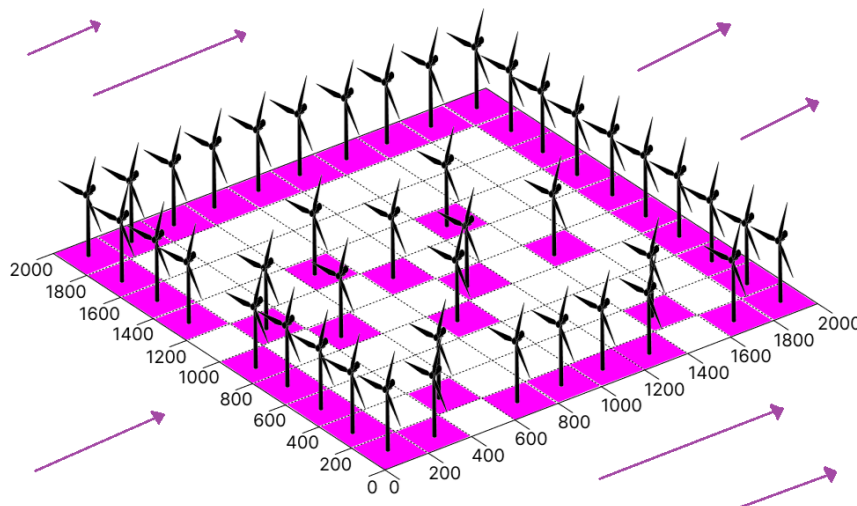


Figura 6. Distribución eólica propuesta por el DE-HeO.

Para concluir la experimentación, es importante analizar el rendimiento operacional de los algoritmos desde una perspectiva estadística sofisticada, por lo que para ello se propone la prueba de Friedman [23]. Esta prueba no paramétrica se caracteriza por comparar a un grupo completo de individuos cuyos resultados sean independientes entre sí, ofreciendo un p -valor para cada uno que determina si existe o no una diferencia significativa contra una hipótesis nula. Para mayor comprensión de este y otras pruebas no paramétricas se invita al lector a consultar las siguientes referencias [23, 24]. Un aspecto importante de la prueba de Friedman es que ofrece un ranking general de los métodos analizados, por lo que se puede saber con precisión cuál es mejor que otro. De igual forma, es importante mencionar que tal como se describió al inicio de la sección, este tipo de pruebas requieren un mínimo de 35 datos independientes para poder realizar la comparación estadística, razón por la cual se realizaron 35 ejecuciones independientes para cada algoritmo. Para este trabajo, dicho análisis se muestra en la Tabla 4, donde se puede apreciar que el IDEI consigue la mejor posición con un p -valor de $3.19E-02$, mientras que sus competidores directos como el CODE o el DE-HeO se ubican en las posiciones 2 y 5 con p -valores de $4.11E-02$ y $5.06E-03$, respectivamente.

Tabla 4. Resultados de la prueba de Friedman.

Algoritmos	p -valores	Ranking
DE	$7.09E-02$	3
DEc	$9.66E-03$	6
CODE	$4.11E-02$	2
Hip-DE	$4.55E-04$	7
DE-HeO	$5.06E-03$	5
HLXDE	$1.41E-03$	4

eXEDE	2.67E-05	9
ReDE	8.10E-04	8
IDEI	3.19E-02	1

6. Conclusiones

En este trabajo se presentó un análisis comparativo entre el algoritmo DE y algunas de sus variantes más contemporáneas para la solución del problema WFLO, el cual es uno de los pilares de la optimización de sistemas de energía en la actualidad. Específicamente, se utilizó un modelo de Jensen sin restricciones territoriales y considerando un máximo de 100 turbinas eólicas. Los resultados obtenidos demostraron la viabilidad del uso de estos algoritmos para la solución de este problema, dado su buen desempeño, estabilidad y costo computacional. No obstante, existe un algoritmo que logró resaltar de entre todos, el IDEI. Esta variante propuesta en 2017 se caracteriza por enfocar su mejora en la etapa de selección del DE. Los resultados posicionaron al IDEI como el mejor en términos de precisión, convergencia y estabilidad no paramétrica. De igual forma, el IDEI fue capaz de entregar una distribución eólica bastante competente que permite ser replicada en modelos más complejos.

Por otra parte, es importante mencionar que ningún algoritmo fue capaz de entregar altas capacidades de energía con menos de 40 turbinas, lo cual indica que su factor evolutivo tiende a beneficiar siempre al número de turbinas en lugar de la potencia entregada. Esto podría solucionarse con la hibridación de un modelo evolutivo con las ventajas de un sistema bio-inspirado. También, como algunos de los trabajos futuros que pueden surgir es la aplicación del IDEI en modelos eólicos como el de Larsen o variantes más sofisticadas del de Jensen. Otro importante aporte es analizar el comportamiento de estos algoritmos desde una perspectiva multiobjetivo para obtener un resultado más cohesivo.

Referencias

1. P. Hou et al., "A review of offshore wind farm layout optimization and electrical system design methods," *J. Mod. Power Syst. Clean Energy*, vol. 7, no. 5, pp. 975–986, 2019. <https://doi.org/10.1007/s40565-019-0550-5>
2. M. Samorani, "The wind farm layout optimization problem," in *Handbook of Wind Power Systems*, Berlin, Heidelberg: Springer, 2014, pp. 21–38. https://doi.org/10.1007/978-3-642-41080-2_2
3. Y. Chen, H. Li, K. Jin, and Q. Song, "Wind farm layout optimization using genetic algorithm with different hub height wind turbines," *Energy Convers. Manag.*, vol. 70, pp. 56–65, 2013. <https://doi.org/10.1016/j.enconman.2013.02.007>
4. M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, "Metaheuristic algorithms: A comprehensive review," in *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*, 2018, pp. 185–231. <https://doi.org/10.1016/B978-0-12-813314-9.00010-4>

5. R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, pp. 341–359, 1997. <https://doi.org/10.1023/A:1008202821328>
6. Y. Yang, S. Tao, H. Li, H. Yang, and Z. Tang, "A multi-local search-based SHADE for wind farm layout optimization," *Electronics*, vol. 13, no. 16, p. 3196, 2024. <https://doi.org/10.3390/electronics13163196>
7. J. J. Thomas et al., "A comparison of eight optimization methods applied to a wind farm layout optimization problem," *Wind Energy Sci.*, vol. 8, no. 5, pp. 865–891, 2023. <https://doi.org/10.5194/wes-8-865-2023>
8. H. Sun, X. Gao, and H. Yang, "A review of full-scale wind-field measurements of the wind-turbine wake effect and a measurement of the wake-interaction effect," *Renew. Sustain. Energy Rev.*, vol. 132, p. 110042, 2020. <https://doi.org/10.1016/j.rser.2020.110042>
9. R. Shakoor, M. Y. Hassan, A. Raheem, and Y. K. Wu, "Wake effect modeling: A review of wind farm layout optimization using Jensen's model," *Renew. Sustain. Energy Rev.*, vol. 58, pp. 1048–1059, 2016. <https://doi.org/10.1016/j.rser.2015.12.229>
10. K. R. Opara and J. Arabas, "Differential Evolution: A survey of theoretical analyses," *Swarm Evol. Comput.*, vol. 44, pp. 546–558, 2019. <https://doi.org/10.1016/j.swevo.2018.06.010>
11. K. Opara and J. Arabas, "Comparison of mutation strategies in differential evolution—a probabilistic perspective," *Swarm Evol. Comput.*, vol. 39, pp. 53–69, 2018. <https://doi.org/10.1016/j.swevo.2017.12.007>
12. Z. Zeng, M. Zhang, T. Chen, and Z. Hong, "A new selection operator for differential evolution algorithm," *Knowl.-Based Syst.*, vol. 226, p. 107150, 2021. <https://doi.org/10.1016/j.knosys.2021.107150>
13. K. Yang, G. Kwak, K. Cho, and J. Huh, "Wind farm layout optimization for wake effect uniformity," *Energy*, vol. 183, pp. 983–995, 2019. <https://doi.org/10.1016/j.energy.2019.07.019>
14. D. Bajer, G. Martinović, and J. Brest, "A population initialization method for evolutionary algorithms based on clustering and Cauchy deviates," *Expert Syst. Appl.*, vol. 60, pp. 294–310, 2016. <https://doi.org/10.1016/j.eswa.2016.05.009>
15. M. F. Ahmad, N. A. M. Isa, W. H. Lim, and K. M. Ang, "Differential evolution with modified initialization scheme using chaotic oppositional based learning strategy," *Alexandria Eng. J.*, vol. 61, no. 12, pp. 11835–11858, 2022. <https://doi.org/10.1016/j.aej.2022.05.028>
16. Z. Meng and C. Yang, "Hip-DE: Historical population-based mutation strategy in differential evolution with parameter adaptive mechanism," *Inf. Sci.*, vol. 562, pp. 44–77, 2021. <https://doi.org/10.1016/j.ins.2021.01.031>
17. S. Prabha and R. Yadav, "Differential evolution with biological-based mutation operator," *Eng. Sci. Technol., Int. J.*, vol. 23, no. 2, pp. 253–263, 2020. <https://doi.org/10.1016/j.jestch.2019.05.012>
18. Y. Cai and J. Wang, "Differential evolution with hybrid linkage crossover," *Inf. Sci.*, vol. 320, pp. 244–287, 2015. <https://doi.org/10.1016/j.ins.2015.05.026>
19. I. Fister, A. Tepeh, and I. Fister Jr., "Epistatic arithmetic crossover based on Cartesian graph product in ensemble differential evolution," *Appl. Math. Comput.*, vol. 283, pp. 181–194, 2016. <https://doi.org/10.1016/j.amc.2016.02.034>
20. V. Ho-Huu, T. Nguyen-Thoi, T. Truong-Khac, L. Le-Anh, and T. Vo-Duy, "An improved differential evolution based on roulette wheel selection for shape and size optimization of truss structures

- with frequency constraints," *Neural Comput. Appl.*, vol. 29, pp. 167–185, 2018. <https://doi.org/10.1007/s00521-016-2426-1>
21. M. Tian, X. Gao, and C. Dai, "Differential evolution with improved individual-based parameter setting and selection strategy," *Appl. Soft Comput.*, vol. 56, pp. 286–297, 2017. <https://doi.org/10.1016/j.asoc.2017.03.010>
 22. E. H. Livingston, "The mean and standard deviation: what does it all mean?" *J. Surg. Res.*, vol. 119, no. 2, pp. 117–123, 2004. <https://doi.org/10.1016/j.jss.2004.02.008>
 23. M. R. Sheldon, M. J. Fillyaw, and W. D. Thompson, "The use and interpretation of the Friedman test in the analysis of ordinal-scale data in repeated measures designs," *Physiother. Res. Int.*, vol. 1, no. 4, pp. 221–228, 1996. <https://doi.org/10.1002/pri.66>
 24. M. W. Fagerland, "t-tests, non-parametric tests, and large studies—a paradox of statistical practice?," *BMC Med. Res. Methodol.*, vol. 12, no. 1, p. 78, 2012. <https://doi.org/10.1186/1471-2288-12-78>