



11

---

# EJEMPLO DE PSO Y AG

# PSO Y GA APLICADOS A LA OPTIMIZACIÓN DE ENERGÍA EÓLICA

- ▶ **Objetivo:** El objetivo principal del artículo es encontrar la mejor manera de organizar los aerogeneradores (turbinas eólicas) en un parque eólico para maximizar la energía total producida y, al mismo tiempo, minimizar el número de aerogeneradores utilizados (y, por lo tanto, el costo).

**GA and PSO Applied to Wind Energy Optimization**

Martín Bilbao<sup>1</sup>, Enrique Alba<sup>2</sup>

<sup>1</sup> Universidad Nacional de la Patagonia Austral, Caleta Olivia, Argentina  
mbilbao@uva.edu.ar

<sup>2</sup> Universidad de Málaga, España  
eal@icca.uma.es

**Abstract.** In this article we analyze two kinds of metaheuristic algorithms applied to wind farm optimization. The basic idea is to utilize CHC (a sort of GA) and GPSO (a sort of PSO) algorithms to obtain an acceptable configuration of wind turbines in the wind farm that maximizes the total output energy and minimizes the number of wind turbines used. The energy produced depends of the farm geometry, wind conditions and the terrain where it is settled. In this work we will analyze three study farm scenarios with different wind speeds and we will apply both algorithms to analyze the performance of the algorithms and the behavior of the computed wind farm designs.

**Keywords:** CHC, Geometric Particle Swarm Optimization, Optimization, Wind Energy, Metaheuristics.

**1 Introduction**

Nowadays, using renewable energies is an increasing area of research and development in all the world, because they are important alternatives to generate free and clean power. The raise of this energy is clear in Europe and America, being a strategic part of development for many countries like Argentina and Spain. A capital interest resides in combining a maximum of energy generation at the same time as reducing the total cost of the wind farm. A farm is a set of wind turbines, every one being costly, whose position is a strategic decision in order to maximize the produced energy. One of the most important aspects of wind farm design is to obtain an optimal location of the wind turbines, because they receive lower wind speed and less energy captures if e.g. they are located behind each other. This effect is called *the wake effect* [1]. The wake effect can be reduced by optimizing the geometry of the wind farm. Then, obtaining a maximum annual profit means taking into account the number of wind turbines and their proper positioning simultaneously. Therefore, an effective algorithm is necessary to get an optimal solution by using a mathematical model of the wind farm as close as possible to a real world complex problem.

Simulated Annealing and Distributed Genetic Algorithms have been used in the past to solved this kind of problem [2][3]. In this work we use other techniques that have provided in the past good solutions in problems like RND (Radio Network Design) that share some points in common to our work [4] and Geometric Particle Swarm Optimization (GPSO) [5] will be applied and analyzed here, showing also that they can provide new state of the art solutions to optimal wind farm design applications.

The rest of the article is structured as follows: Section 2 we will explain the wake model, the power model and the cost model used. Section 3 will detail CHC and GPSO the proposed algorithms. In section 4 we will show the experimental studies and discuss on the results obtained and in Section 5 the conclusions and future work.

# PSO Y GA APLICADOS A LA OPTIMIZACIÓN DE ENERGÍA EÓLICA

## ► Metaheurísticas:

✓ CHC (un tipo de Algoritmo Genético)

✓ GPSO (Optimización Geométrica por Enjambre de Partículas).

GA and PSO Applied to Wind Energy Optimization

Martín Bilbao<sup>1</sup>, Enrique Alba<sup>2</sup>

<sup>1</sup> Universidad Nacional de la Patagonia Austral, Caleta Olivia, Argentina  
mbilbao@uva.edu.ar

<sup>2</sup> Universidad de Málaga, España  
eat@icca.uma.es

**Abstract.** In this article we analyze two kinds of metaheuristic algorithms applied to wind farm optimization. The basic idea is to utilize CHC (a sort of GA) and GPSO (a sort of PSO) algorithms to obtain an acceptable configuration of wind turbines in the wind farm that maximizes the total output energy and minimizes the number of wind turbines used. The energy produced depends of the farm geometry, wind conditions and the terrain where it is settled. In this work we will analyze three study farm scenarios with different wind speeds and we will apply both algorithms to analyze the performance of the algorithms and the behavior of the computed wind farm designs.

**Keywords:** CHC, Geometric Particle Swarm Optimization, Optimization, Wind Energy, Metaheuristics.

### 1 Introduction

Nowadays, using renewable energies is an increasing area of research and development in all the world, because they are important alternatives to generate free and clean power. The raise of this energy is clear in Europe and America, being a strategic part of development for many countries like Argentina and Spain. A capital interest resides in combining a maximum of energy generation at the same time as reducing the total cost of the wind farm. A farm is a set of wind turbines, every one being costly, whose position is a strategic decision in order to maximize the produced energy. One of the most important aspects of wind farm design is to obtain an optimal location of the wind turbines, because they receive lower wind speed and less energy captures if e.g. they are located behind each other. This effect is called *the wake effect* [1]. The wake effect can be reduced by optimizing the geometry of the wind farm. Then, obtaining a maximum annual profit means taking into account the number of wind turbines and their proper positioning simultaneously. Therefore, an effective algorithm is necessary to get an optimal solution by using a mathematical model of the wind farm as close as possible to a real world complex problem.

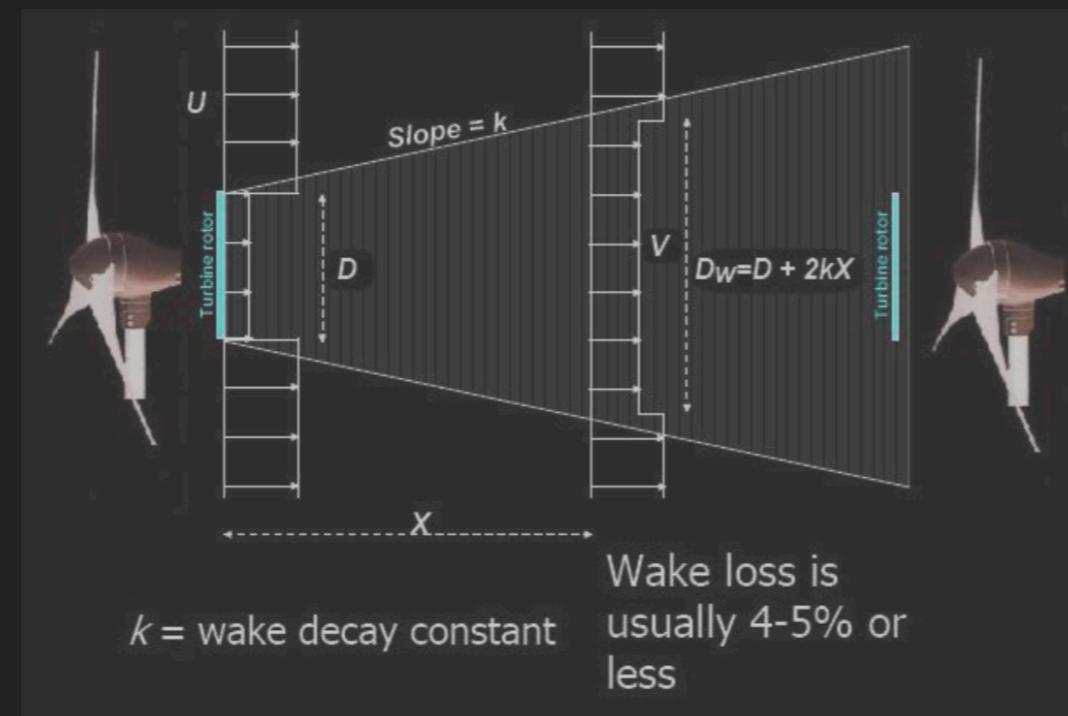
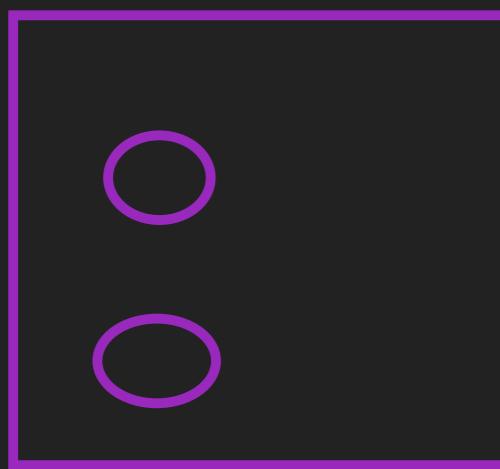
Simulated Annealing and Distributed Genetic Algorithms have been used in the past to solved this kind of problem [2][3]. In this work we use other techniques that have provided in the past good solutions in problems like RND (Radio Network Design) that share some points in common to our work [4] and Geometric Particle Swarm Optimization (GPSO) [5] will be applied and analyzed here, showing also that they can provide new state of the art solutions to optimal wind farm design applications.

The rest of the article is structured as follows: Section 2 we will explain the wake model, the power model and the cost model used. Section 3 will detail CHC and GPSO the proposed algorithms. In section 4 we will show the experimental studies and discuss on the results obtained and in Section 5 the conclusions and future work.



# ¿POR QUÉ ES IMPORTANTE ESTE PROBLEMA?

- ▶ **Demanda de Energía Renovable:** La energía eólica es limpia y cada vez más importante.
- ▶ **Costo:** Los aerogeneradores son caros. Cuanto menos aerogeneradores se usen más efectiva será ls inversión.
- ▶ **Efecto Estela (Wake Effect):** Cuando el viento pasa a través de un aerogenerador, crea una "estela" detrás de él (un área de aire más lento y turbulento). Los aerogeneradores colocados en la estela de otros producirán menos energía. Por lo que agrupar los aerogeneradores muy juntos no es óptimo.
- ▶ **Complejidad:** Encontrar la disposición óptima es manualmente muy difícil debido a la gran cantidad de configuraciones posibles.



# ¿CÓMO SE MODELA EL PARQUE EÓLICO?



## MODELO DEL EFECTO ESTELA

streamdown

- ▶ Se calcula el "déficit de velocidad" (cuánto se reduce la velocidad del viento aguas abajo de un aerogenerador).

$$dV = U_0 - U_t = U_0 \frac{1 - \sqrt{1 - C_t}}{\left(\frac{1+2kX}{D}\right)^2}$$

- ▶ donde:

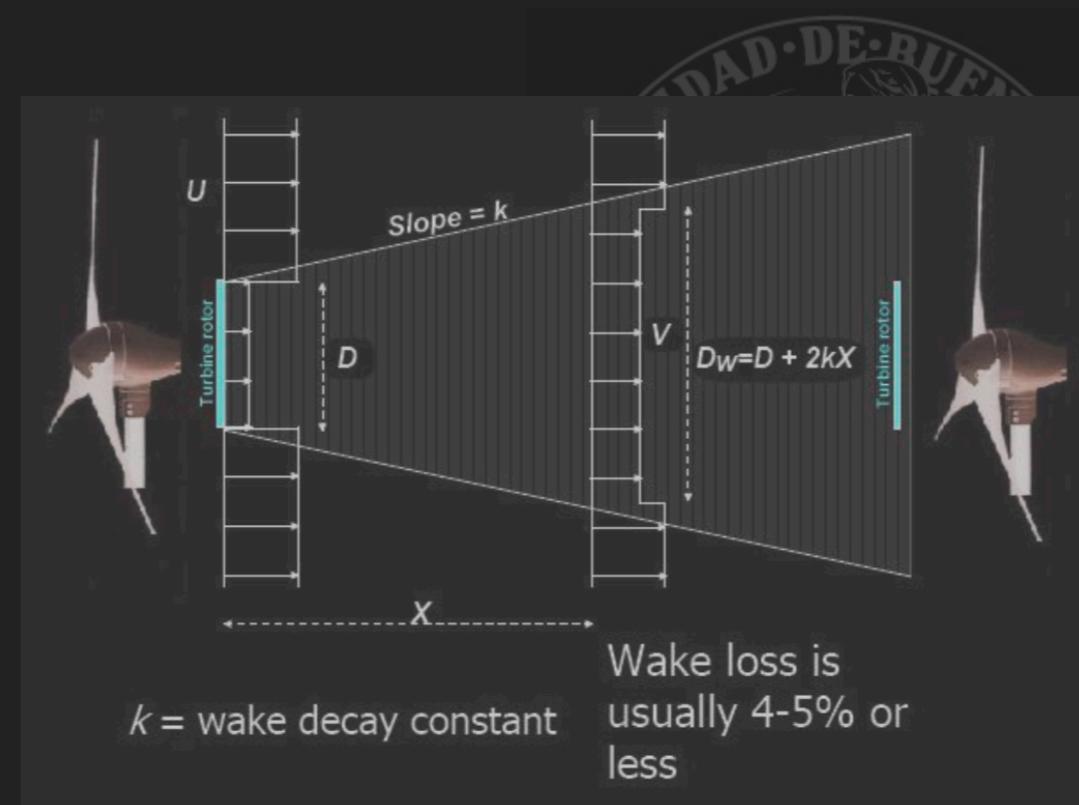
$U_0$ : velocidad inicial del flujo libre

$U_t$ : velocidad en la estela a una distancia  $X$

$C_t$ : coeficiente de empuje del aerogenerador (mide la "resistencia" o "freno aerodinámico" que el rotor opone al paso del viento)

$D$ : diámetro de la turbina

$k$ : constante de decaimiento de la estela



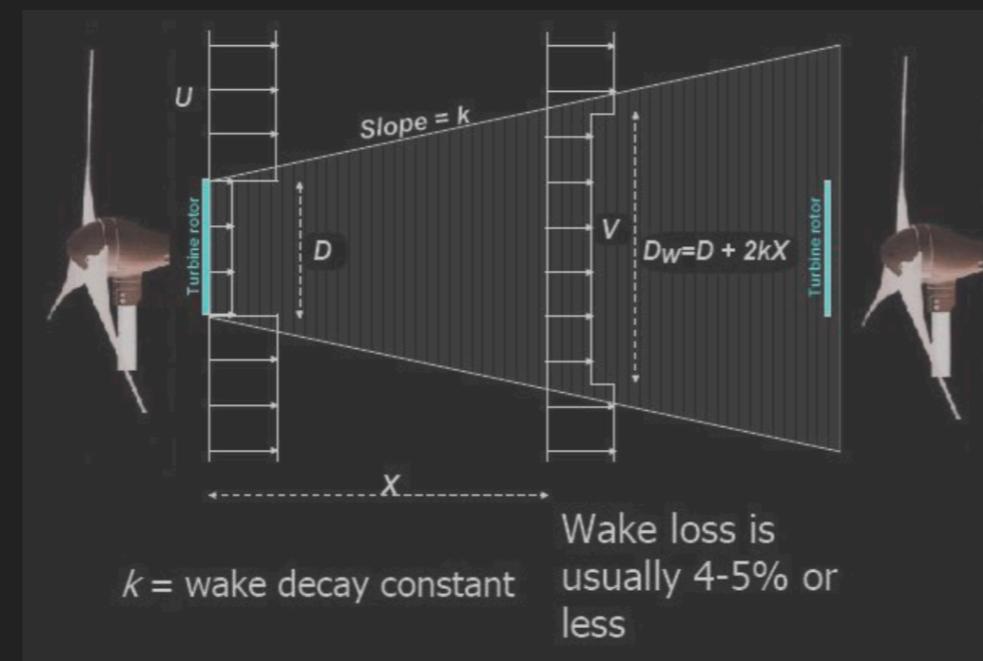
## MODELO DEL EFECTO ESTELA

- ▶  $U_t$ : mide la velocidad del viento en un punto  $x$  pero teniendo en cuenta el efecto combinado de múltiples turbinas que se encuentran aguas arriba (delante) de ese punto  $x$ .

$$U_t = U_0 \left[ 1 - \sqrt{\sum_{i=1}^N \left(1 - \frac{U_i}{U_0}\right)^2} \right]$$

- ▶ donde:

$U_i$ : representa el déficit de velocidad ( $U_0 - U_{t\_individual\_i}$ ) que sería causado por cada turbina individual  $i$  aguas arriba, si esa turbina  $i$  estuviera actuando sola sobre el flujo libre  $U_0$  y estuviéramos midiendo el efecto en el punto  $x$ .



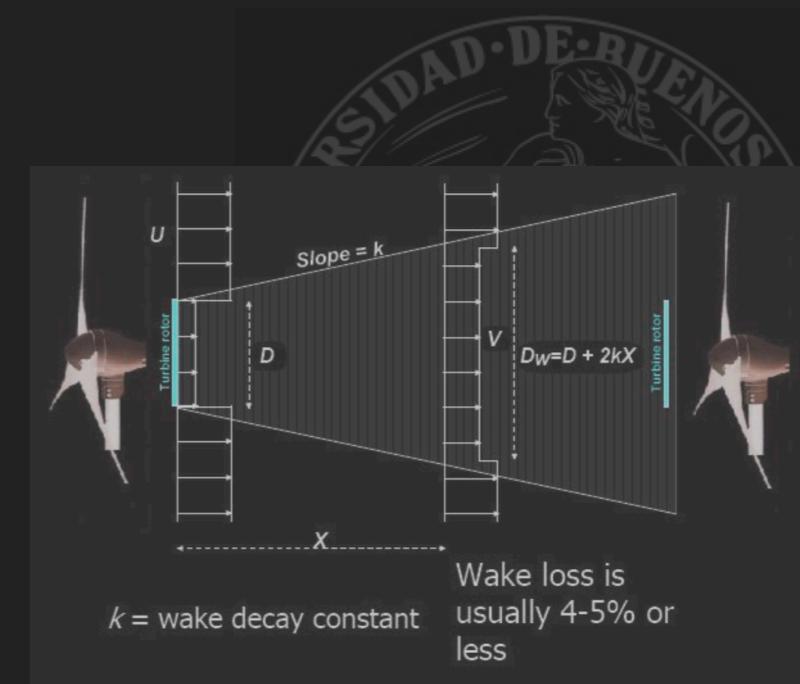
## MODELO DE POTENCIA

- ▶ Define cuánta potencia produce un solo aerogenerador en función de la velocidad del viento ( $U_x$ ) que experimenta (afectada por el modelo de estela).

**cut-off**

- ▶ La curva de potencia es una función por tramos:

$$P_i(kW) = \begin{cases} 0 & \text{for } U_x < 3m/s, \\ 0.3U_x^3 & \text{for } 3m/s \leq U_x \leq 13m/s, \\ 750 & \text{for } 13m/s \leq U_x \leq 25m/s, \\ 0 & \text{for } 25m/s < U_x \end{cases}$$



- ▶ La potencia total del parque ( $P_{tot}$ ) es la suma de la potencia de todos los aerogeneradores individuales.

$$P_{tot} = \sum_{i=1}^N P_i,$$

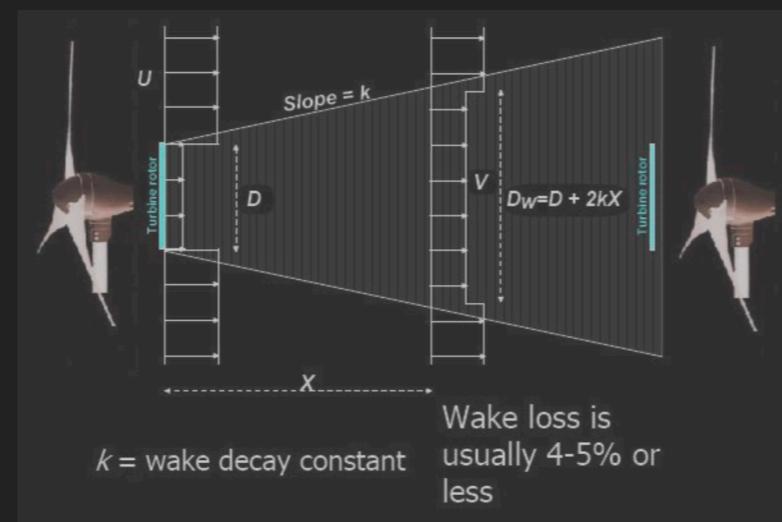
## MODELO DE COSTO

- El costo total anual depende únicamente del número de aerogeneradores ( $N$ ).

$$cost_{tot} = cost_{gy}N(2/3 + 1/3e^{-0.00174N^2})$$

- donde:

$cost_{gy}$  es el costo por aerogenerador al año (400.000 € en este estudio).

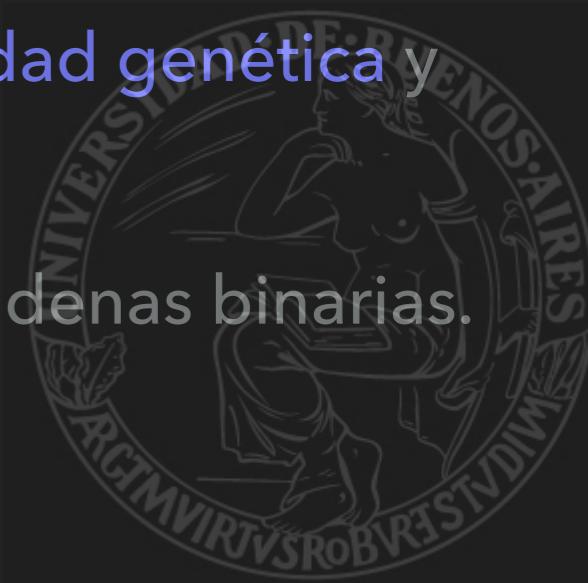


# ALGORITMOS DE OPTIMIZACIÓN



# ALGORITMO CHC (ESHELMAN, 1991)

- ▶ CHC significa: Cross-generational elitist selection, Heterogeneous recombination, and Cataclysmic mutation.
- ▶ Es una variante de AG diseñada para mantener una alta diversidad genética y evitar la convergencia prematura.
- ▶ Utiliza una representación de las soluciones codificada como cadenas binarias.



# ALGORITMO CHC (ESHELMAN, 1991)

- ▶ **Operador de cruce:** Utiliza el cruce HUX (Cruce Uniforme Heterogéneo) (Eshelman, 1991) para crear nuevas soluciones. HUX copia los bits coincidentes de los padres y luego divide los bits diferentes equitativamente entre los descendientes para mantener la diversidad.

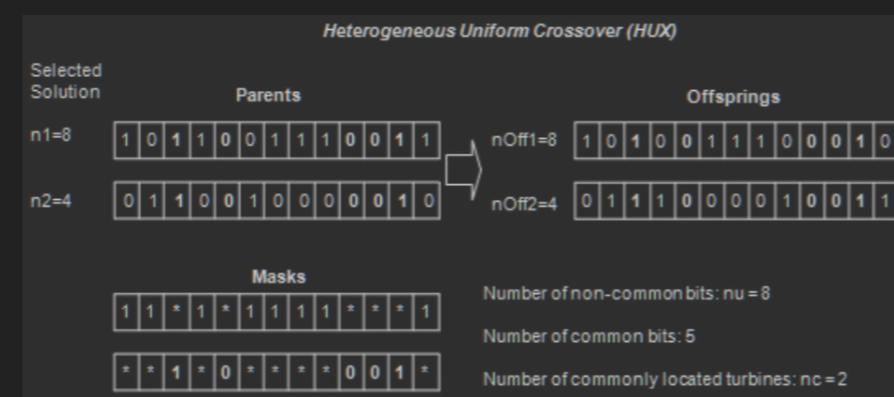
Padre 1: 1 0 1 1 0 1 0 1

Padre 2: 1 1 0 1 1 1 0 0

- ▶ Comparando bit a bit: las posiciones diferentes son 2, 3, 5, 8 → Total diferencias: 4
- ▶ Al contabilizar 4 se elige la mitad de genes aleatoriamente ( es decir 2): genes 3 y 8
- ▶ Se intercambian genes 3 y 8 entre los padres: Padre 1: 1 0 1 1 0 1 0 1      Padre 2: 1 1 0 1 1 1 0 0

Hijo 1: igual a Padre 1, excepto en posiciones 3 y 8 → 1 0 0 1 0 1 0 0

Hijo 2: igual a Padre 2, excepto en posiciones 3 y 8 → 1 1 1 1 1 1 0 1



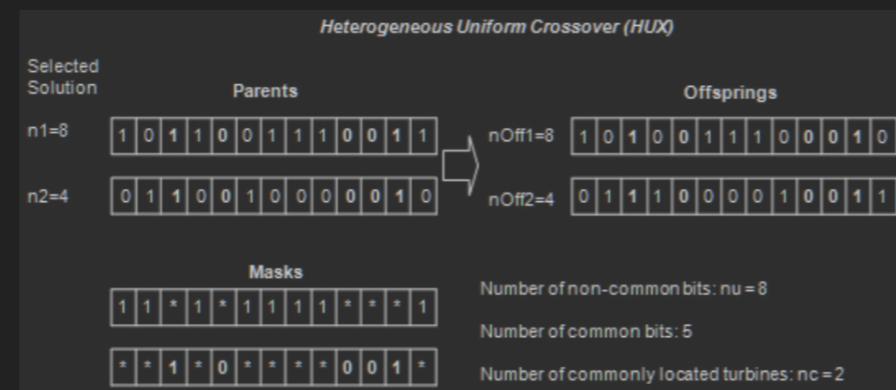
# ALGORITMO CHC (ESHELMAN, 1991)

- ▶ **Prevención de Incesto:** El cruce solo ocurre si los padres son lo suficientemente diferentes (la **distancia de Hamming** supera un umbral). Este umbral disminuye si la población se estanca.
- ▶ La **distancia de Hamming** es una métrica que mide cuántos elementos difieren entre dos cadenas del mismo tamaño.
- ▶ Dada dos cadenas A y B de igual longitud, la distancia de Hamming es el número de posiciones en las que A y B tienen diferentes valores.

Cadena A: 1 0 1 1 0 1

Cadena B: 1 1 0 1 1 1

- ▶ Comparando posición por posición: **Distancia de Hamming = 3**



# ALGORITMO CHC (ESHELMAN, 1991)

- ▶ **Reinicio Cataclísmico:** Si la población converge demasiado (sin mejoras durante un tiempo), todas las soluciones, excepto la mejor, se mutan intensamente (inversión del 50% de los bits) para reintroducir diversidad.
- ▶ **Selección Elitista:** Solo las mejores soluciones (padres + descendencia) sobreviven a la siguiente generación.

## Algorithm 1 CHC

```
1:  $t \leftarrow 0$ ; /* evaluation */  
2:  $initialize(Pa, Distance)$  /*Initialize the population and the distances */  
3: while not stop criterion( $t, Pa$ ) do  
4:    $Parents \leftarrow selected(Pa)$ ; /* Selected parent */  
5:    $Offspring \leftarrow HUX(Parents)$  /* Crossover HUX */  
6:    $evaluate(Pa, Offspring)$  /*evaluate Offspring*/  
7:    $Pa \leftarrow elitism(Offspring, Pa)$   
8:   if  $Pa$  no change then  
9:      $distance \leftarrow distance - 1$ ;  
10:    if  $distance == 0$  then  
11:       $reset(Pa)$   
12:       $initialize(distance)$   
13:    end if  
14:  end if  
15:   $t \leftarrow t + 1$  /* One more generation */  
16: end while  
17: Return: best solution found.
```



# OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS GEOMÉTRICA (GPSO)

- ▶ Es una variante de PSO diseñada para espacios de búsqueda combinatorios (no continuos).
- ▶ En lugar de las actualizaciones tradicionales de velocidad y posición de PSO, utiliza una operación de cruce basada en máscaras llamada 3PMBCX (Cruce Basado en Máscaras de Tres Padres).
- ▶ Ejemplo, si tenemos una partícula con:

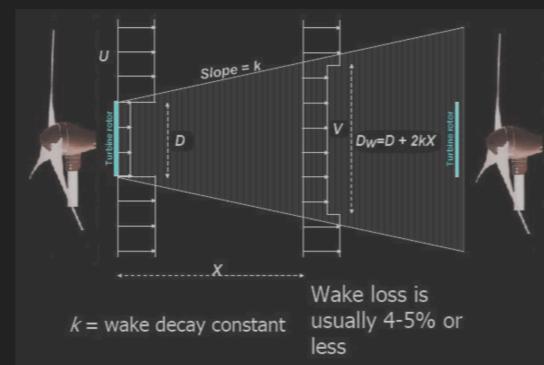
Posición actual (Padre 1): 1 0 1 0 1 1

pBest (Padre 2): 0 1 0 1 1 0

gBest (usado como máscara, Padre 3): 1 1 0 0 1 0

- ▶ Si un gen de la mascara es 0 se toma el gen del padre 1, si es 1 se toma ese gen del padre 2

Nueva posición: 0 1 1 0 1 1



# OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS GEOMÉTRICA (GPSO)

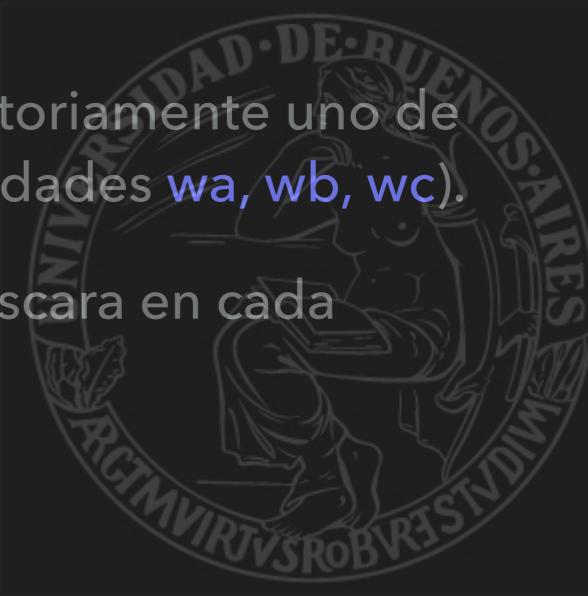
► Para cada partícula (solución  $x_i$ ):

- ✓ Se consideran tres "padres": su posición actual ( $x_i$ ), su mejor posición personal histórica ( $h_i$ ) y la mejor posición global (social) encontrada ( $g_i$ ).
- ✓ Se genera una máscara de cruce. Cada posición en la máscara elige aleatoriamente uno de los tres padres (**a**, **b** o **c**, correspondientes a  $x_i$ ,  $g_i$ ,  $h_i$  con ciertas probabilidades  $w_a$ ,  $w_b$ ,  $w_c$ ).
- ✓ La nueva partícula se forma tomando el bit del parente indicado por la máscara en cada posición.

► También se aplica una pequeña mutación.

## Algorithm 2 GPSO

```
1:  $S \leftarrow InitializeSwarm();$  /* Initialize Swarm */  
2: while not stop criteria do  
3:   for each particle  $x_i$  of  $S$  do  
4:     evaluate( $x_i$ )  
5:     if fitness( $x_i$ )  $\geq$  fitness( $h_i$ ) then  
6:        $h_i \leftarrow x_i;$   
7:     end if  
8:     if fitness( $h_i$ )  $\geq$  fitness( $g_i$ ) then  
9:        $g_i \leftarrow h_i;$   
10:    end if  
11:   end for  
12:   for each particle  $x_i$  of  $S$  do  
13:      $x_i \leftarrow 3PMBCX((x_i, w_a), (g_i, w_b), (h_i, w_c))$   
14:     mutation( $x_i$ )  
15:   end for  
16: end while  
17: Return: best solution found.
```



# INSTANCIACIÓN DE LOS ALGORITMOS



# FUNCIÓN OBJETIVO

- ▶ Se busca maximizar el beneficio (profit).

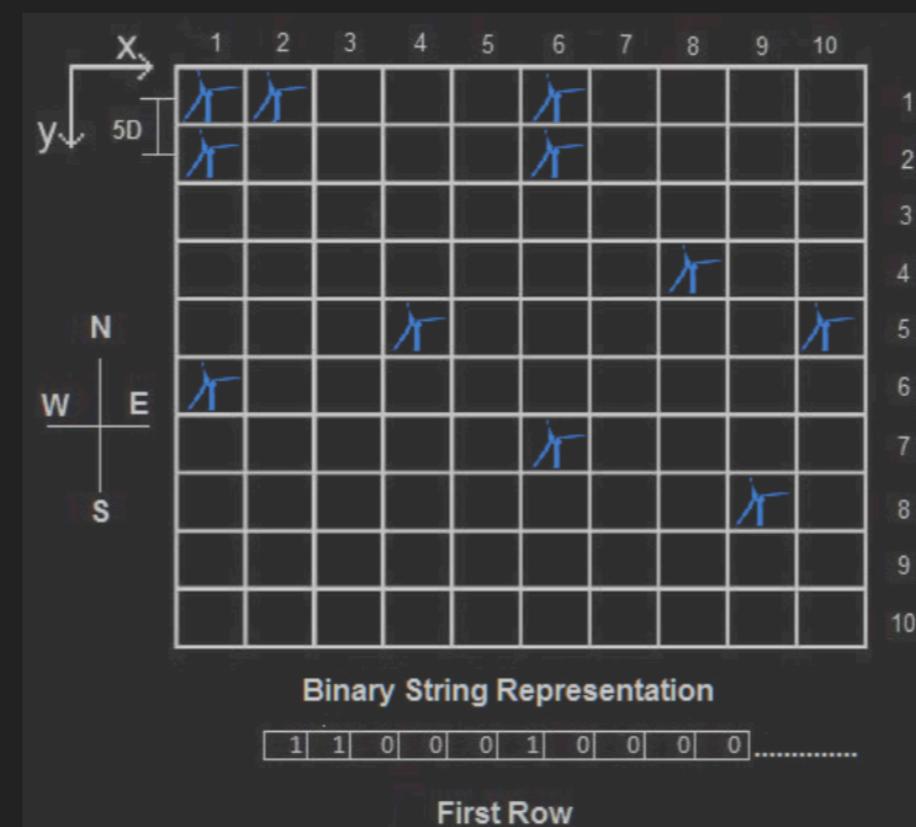
$$\text{profit} = \left[ st - \left( \frac{\text{cost}_{\text{tot}}}{P_{\text{tot}}} \right) \right] P_{\text{tot}}$$

- ▶ **st**: precio de venta de la energía (0.1 €/kWh).
- ▶ **P<sub>tot</sub>**: producción total de energía (del modelo de potencia) en kWh.
- ▶ **cost<sub>tot</sub>**: costo total (del modelo de costo).
- ▶ El número de aerogeneradores (**N**) es desconocido y es parte de lo que los algoritmos intentan optimizar.



# REPRESENTACIÓN DE LA UBICACIÓN DE LOS AEROGENERADORES

- ▶ El área del parque eólico es una cuadrícula de 10x10 celdas.
- ▶ Cada celda puede tener un aerogenerador (1) o no (0).
- ▶ Esto se representa mediante una cadena binaria de 100 bits (10x10).
- ▶ Cada celda mide  $5D \times 5D$  (donde D es el diámetro del rotor, 44m), es decir, 220m x 220m. Este espaciado ayuda a satisfacer reglas generales comunes para la colocación de aerogeneradores.

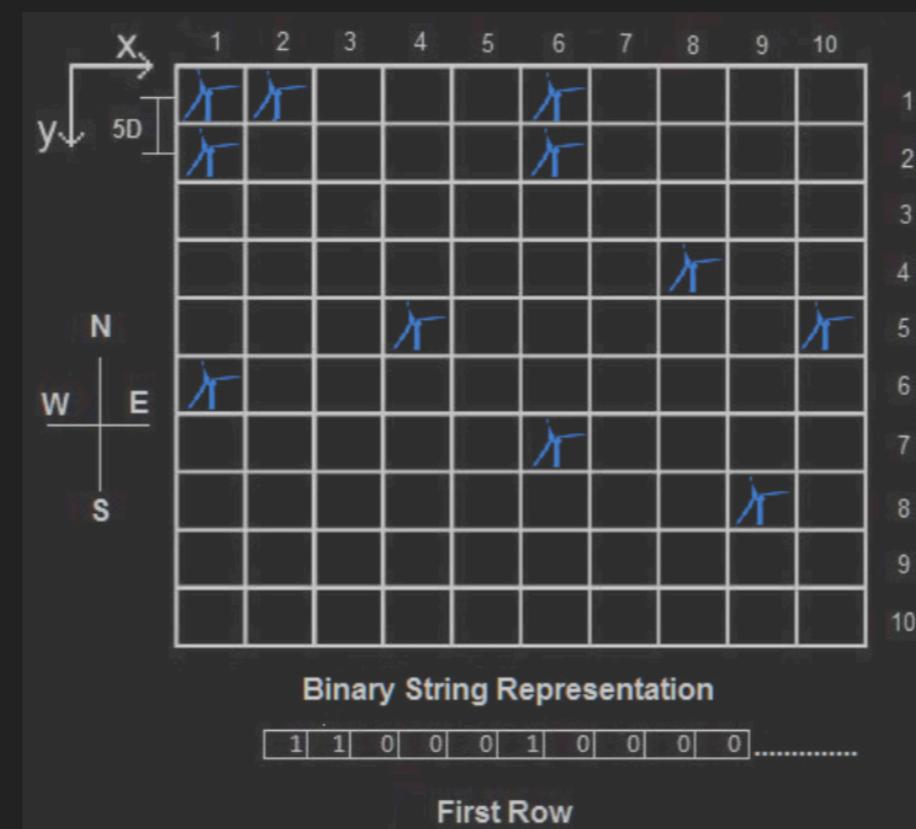


# ESTUDIO EXPERIMENTAL



# ESTUDIO EXPERIMENTAL

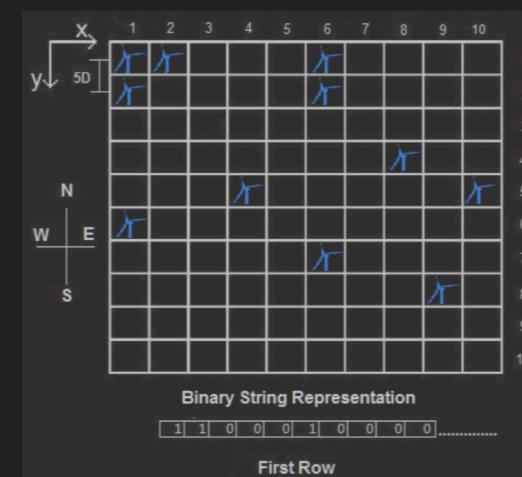
- ▶ Se probaron ambos algoritmos en tres escenarios, asumiendo viento uniforme del Norte:
  - ✓ Escenario (a): Velocidad del viento de 12 m/s
  - ✓ Escenario (b): Velocidad del viento de 18 m/s
  - ✓ Escenario (c): Velocidad del viento de 25 m/s



# ESTUDIO EXPERIMENTAL

► Para cada escenario, ejecutaron tanto CHC como GPSO 30 veces y compararon:

- ✓ Aptitud (beneficio) promedio
- ✓ Producción de potencia
- ✓ Número de aerogeneradores
- ✓ Eficiencia (potencia real / potencia teórica máxima sin pérdidas por estela)
- ✓ Tiempo de ejecución
- ✓ Número de evaluaciones para encontrar la mejor solución
- ✓ Utilizan la prueba de Kruskal-Wallis para la significancia estadística.



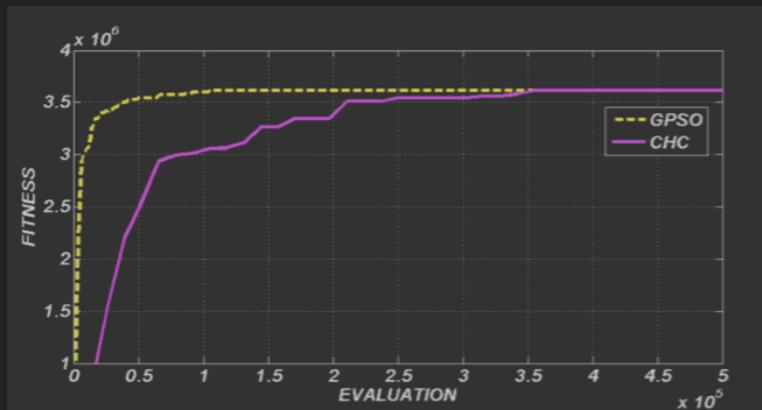
## RESULTADOS CLAVE



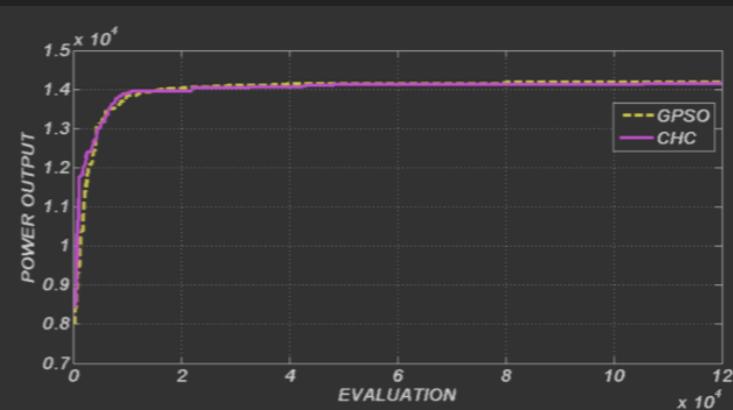
# RESULTADOS CLAVE

## ▶ Escenario (a) - 12 m/s (Viento Moderado):

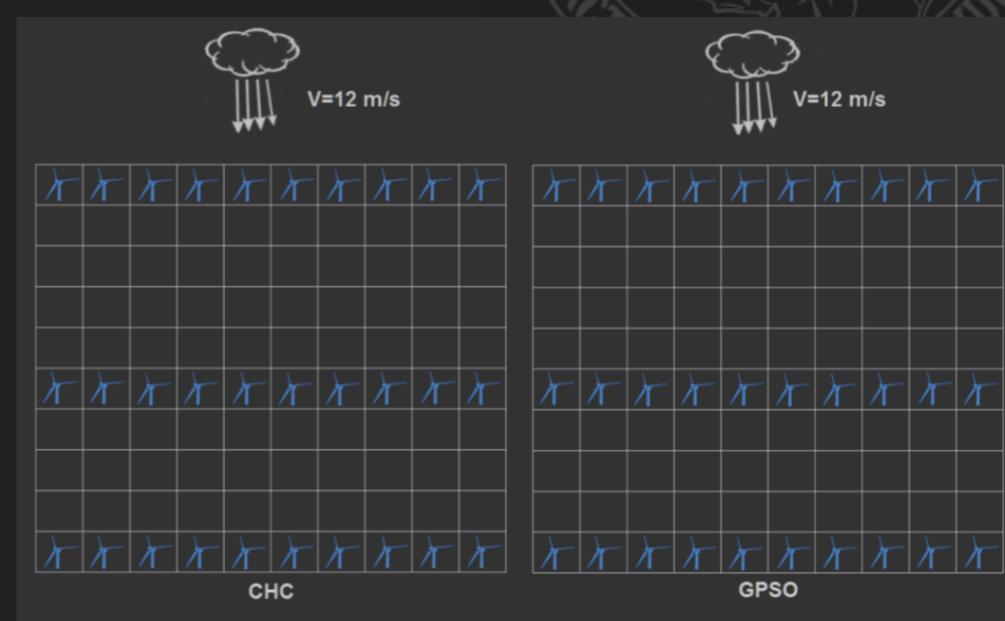
- ✓ CHC obtuvo una aptitud promedio y eficiencia ligeramente mejores.
- ✓ GPSO fue más rápido (menor tiempo de ejecución y menos evaluaciones).
- ✓ Ambos convergieron a un diseño de 30 aerogeneradores dispuestos en tres filas densas perpendiculares al viento.
- ✓ CHC fue estadísticamente más preciso.



(a) Better fitness values



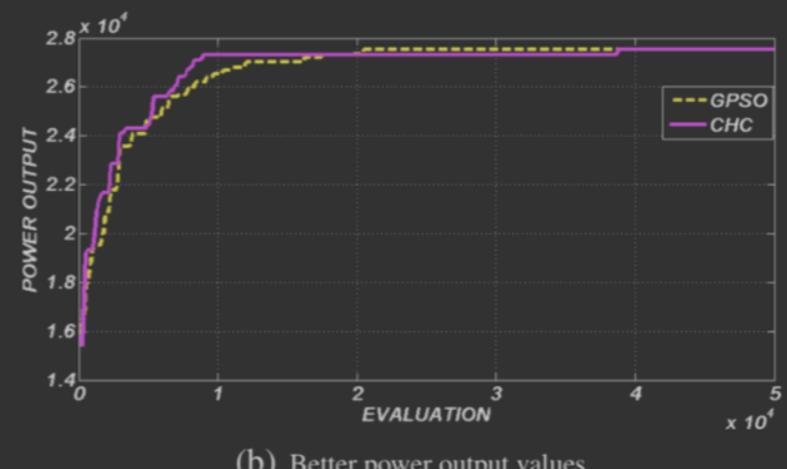
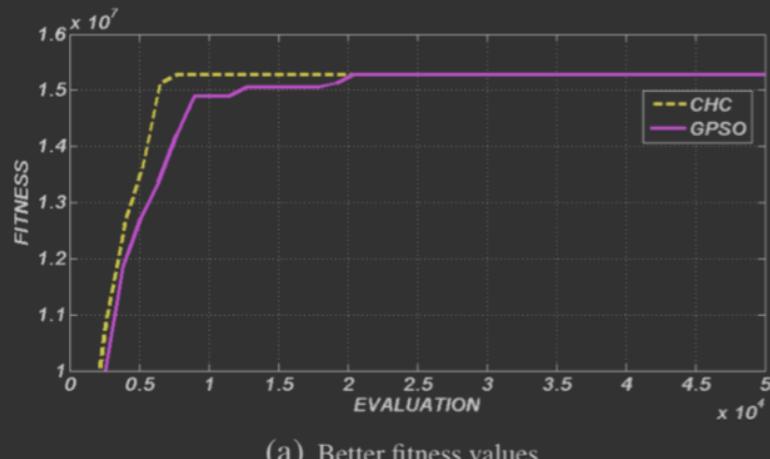
(b) Better power output values



# RESULTADOS CLAVE

► Escenario (b) - 18 m/s (Viento Fuerte):

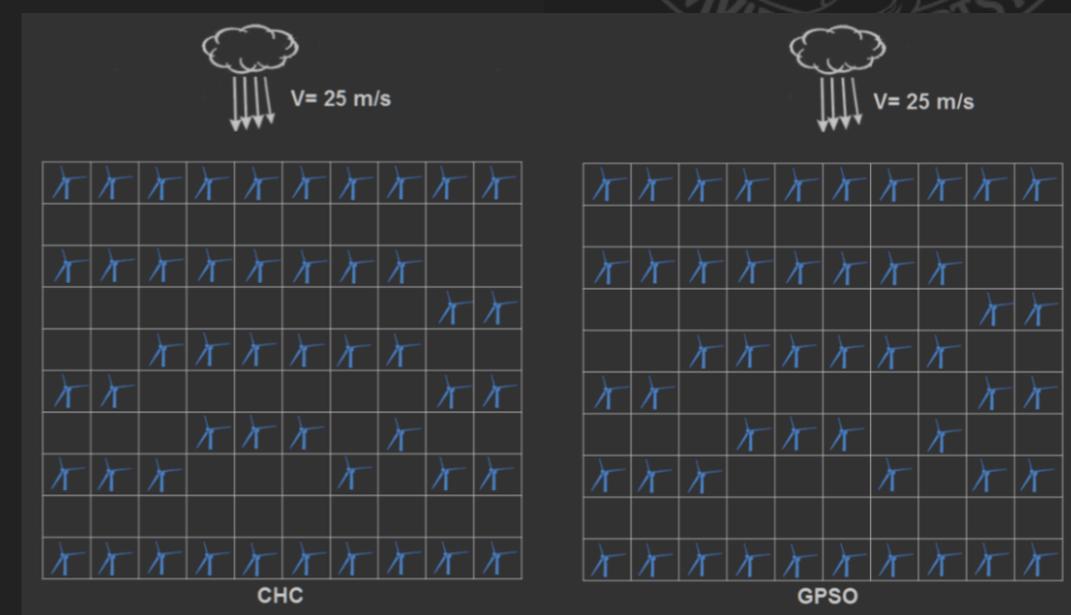
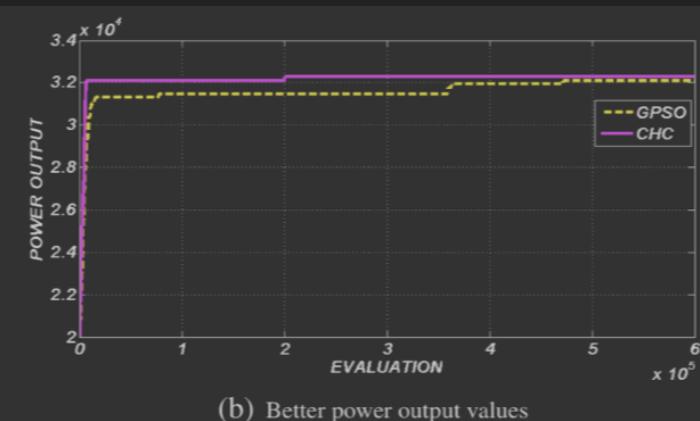
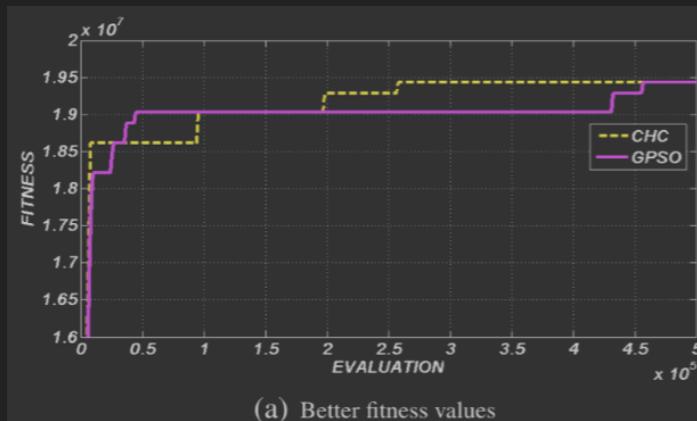
- ✓ CHC y GPSO obtuvieron la misma aptitud promedio, potencia y eficiencia.
- ✓ CHC fue significativamente más rápido en términos de tiempo de ejecución y evaluaciones.
- ✓ Ambos convergieron a un diseño de 40 aerogeneradores, típicamente en dos filas centrales densas.



# RESULTADOS CLAVE

► Escenario (c) - 25 m/s (Viento Muy Fuerte - cerca del corte):

- ✓ CHC nuevamente obtuvo mejores valores en la mayoría de las métricas (aptitud, eficiencia) y fue significativamente más rápido.
- ✓ Ambos convergieron a un diseño de 50 aerogeneradores, típicamente en tres filas centrales densas.

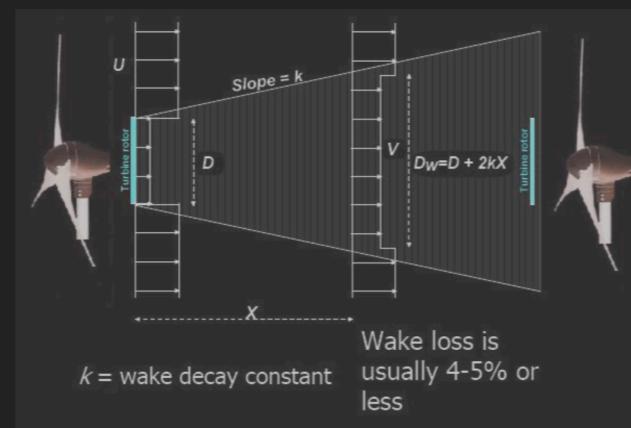


## OBSERVACIONES GENERALES DE LOS RESULTADOS



# OBSERVACIONES GENERALES DE LOS RESULTADOS

- ▶ Ambos algoritmos son competitivos.
- ▶ CHC a menudo logra una calidad de solución ligeramente mejor (aptitud, eficiencia).
- ▶ CHC fue a menudo más rápido para encontrar la mejor solución, especialmente en los escenarios (b) y (c).
- ▶ El número óptimo de aerogeneradores aumentó con la velocidad del viento ( $30 \rightarrow 40 \rightarrow 50$ ). Esto tiene sentido, ya que velocidades de viento más altas significan más energía potencial para capturar, lo que hace viables más aerogeneradores a pesar del costo y los efectos de estela.
- ▶ Los diseños óptimos generalmente implicaron aerogeneradores dispuestos en filas densas perpendiculares a la dirección del viento, una estrategia común para maximizar la exposición de las primeras filas mientras se reconocen los efectos de estela para las filas subsiguientes.

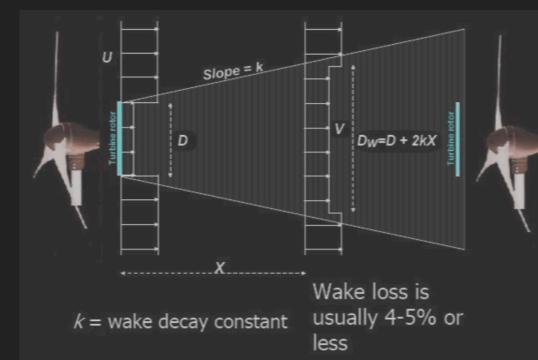


## CONCLUSIONES Y TRABAJO FUTURO



# CONCLUSIONES Y TRABAJO FUTURO

- ▶ Tanto **CHC** como **GPSO** son efectivos para este problema de optimización de parques eólicos.
- ▶ **CHC** generalmente mostró un rendimiento general ligeramente mejor en los escenarios probados.
- ▶ La función de costo permite diferentes soluciones (número de aerogeneradores), mientras que la función de potencia conduce a trayectorias evolutivas similares para el diseño.
- ▶ **Trabajo Futuro:**
  - ✓ Considerar modelos de parque **más realistas** (por ejemplo, efectos del terreno, impacto estético).
  - ✓ Estudiar la **escalabilidad** para parques eólicos más grandes y diferentes parámetros de los aerogeneradores.
  - ✓ Resolverlo como un problema **multiobjetivo** (por ejemplo, optimizando directamente el costo frente a la energía producida, en lugar de combinarlos en una única función de beneficio).



## REFERENCIAS BIBLIOGRÁFICAS Y WEB (I)

- ▶ Bilbao, M., & Alba, E. (2009, October). GA and PSO applied to wind energy optimization. In XV Congreso Argentino de Ciencias de la Computación (Vol. 10).
- ▶ Eshelman, L.J., & Schaffer, J.D. (1993). Real-Coded Genetic Algorithms and Interval-Schemata. In Foundations of Genetic Algorithms 2 (FOGA-2), pp. 187-202. Morgan Kaufmann Publishers.
- ▶ Eshelman, L. J. (1991). The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In Foundations of genetic algorithms (Vol. 1, pp. 265-283). Elsevier.

