



27-4-2025

Memoria Técnica: Optimización Evolutiva de Redes de Sensores

Grado de Ingeniería Informática e
Inteligencia Artificial

JAVIER REVUELTA FERNÁNDEZ

Computación Evolutiva – 2ºA



~ ÍNDICE ~

Explicación del Código:	1
Problemas que han surgido:	5
<i>Posibles mejoras a futuro:</i>	6
Resultados:	7
<i>Evolución de la mejor fitness:</i>	7
<i>Evolución de la fitness media:</i>	8
<i>Evolución de la superficie media cubierta:</i>	9
<i>Evolución de la potencia media:</i>	10
<i>Distribución espacial de sensores:</i>	11
Conclusión y Discusión de Resultados:	12
<i>Conclusión Final:</i>	12

Explicación del Código:

En esta sección se describe detalladamente el flujo de ejecución del programa, los módulos que lo componen y su funcionamiento interno.

Antes de indagar a fondo en el código, hagamos un resumen general del flujo que sigue el programa cuando este es ejecutado:

1. Se inicializan los parámetros generales (número de sensores, dimensiones del área, número de frecuencias disponibles, parámetros evolutivos, etc).
2. Se crea la población inicial de individuos de forma aleatoria.
3. Se evalúa cada individuo, calculando su cobertura, consumo energético e interferencias que resulta en la fitness.
4. Se inicia el bucle de evolución:
 - a. En cada generación, se seleccionan por torneo dos individuos para reproducirse.
 - b. Se realiza cruce y mutación para generar una nueva población.
 - c. Se evalúa la nueva población y se registran sus estadísticas (mejor fitness, media de fitness, superficie media cubierta, potencia media).
 - d. Se guarda la información en archivos CSV.
5. El proceso evolutivo continúa hasta que se alcanza el número máximo de generaciones o se detecta convergencia, es decir, la mejora entre generaciones es mínima.

A partir de ahora, entraremos mucho más en detalle en cada sección del algoritmo consiguiendo desgranar por completo el código para que quede completamente comprensible para cualquier tipo de persona.

La ejecución del programa comienza desactivando el buffer de salida (setvbuf) para garantizar que toda la información que se imprima en consola (por pantalla) se muestre de inmediato. Esto es esencial para ir monitorizando el avance del algoritmo generación a generación en tiempo real. A continuación, se inicializa el generador de números aleatorios utilizando "srand(time(NULL))", de modo que en cada ejecución se generen poblaciones distintas y se exploren configuraciones nuevas de sensores, haciendo que las simulaciones sean siempre diferentes.

Seguidamente se prepara el guardado de resultados. Se define una variable global que deberemos actualizar manualmente para indicar el número de simulación, establecer el nombre del archivo y evitar sobreescrituras. A partir del número de simulación se generan dos nombres de archivos CSV: uno para los datos de evolución global de la población (simulacionX_datos.csv) y otro para almacenar las configuraciones individuales de los sensores en las generaciones clave (simulacionX_sensores.csv). Estos archivos se abren en modo escritura y, acto seguido, se inicializan con sus respectivas cabeceras, donde se definen los campos de cada archivo. El primero guardará generación, mejor fitness, media de fitness, superficie cubierta media y potencia media, mientras que el segundo contendrá toda la información de cada sensor individual (posición, potencia, frecuencia) tanto en la generación inicial como en la final.

Tras preparar la estructura de archivos, el programa inicializa las estructuras de datos fundamentales. Se crean dos arrays de individuos: uno para la población actual y otro para la nueva población generada en cada iteración evolutiva. También se inicializa un historial de medias de fitness que servirá para detectar convergencia más adelante, y se reserva memoria para una matriz de correlaciones entre frecuencias. Esta matriz, inicializada mediante la función “inicializar_matriz_correlacion”, define cómo se afectan dos sensores si utilizan frecuencias cercanas: sensores en la misma frecuencia interfieren totalmente (valor 1.0), sensores en frecuencias adyacentes interfieren parcialmente (valor 0.5), sensores a dos posiciones de distancia interfieren poco (0.25) y para frecuencias más lejanas no hay interferencia (0.0). Esta matriz fija la relación de interferencias entre todos los pares de canales.

Una vez listas las estructuras de datos, se pasa a la creación de la población inicial. Mediante un bucle se inicializan aleatoriamente los individuos. Cada individuo se compone de 1000 sensores, cuyos valores de posición (x,y), potencia y frecuencia se asignan de manera aleatoria en los rangos permitidos. Esta inicialización la realiza la función “inicializar_individuo”, que garantiza una distribución inicial variada de sensores sobre el espacio. Acto seguido, se evalúa cada individuo utilizando la función “evaluar”, que es uno de los pilares fundamentales del algoritmo, ya que a través de ella se mide la calidad de cada configuración de sensores.

El primer aspecto que se calcula es el consumo energético total del individuo. Este valor se obtiene sumando las potencias de transmisión de todos los sensores. Es importante entender que la potencia de cada sensor representa su energía de emisión, por tanto, un mayor valor de potencia significa un mayor consumo energético, lo cual, en un sistema real, implica menor duración de las baterías y necesidad de mantenimiento más frecuente. Esta suma de potencias se realiza de forma sencilla recorriendo todos los sensores del individuo.

A continuación, se calcula la interferencia total del individuo. Para calcular la interferencia, el algoritmo primero evalúa la distancia entre los dos sensores mediante la función “calcular_interferencia”. Si dos sensores están separados por más de un umbral predefinido (10 unidades en este caso), entonces se considera que no hay interferencia entre ellos. Esta lógica refleja el comportamiento real de las señales: sensores muy alejados simplemente no interfieren entre sí de manera apreciable. Solo si la distancia es inferior al umbral, o lo que de manera práctica sería equivalente a que dos sensores presenten solapamiento, se consulta la matriz de correlaciones de frecuencias que establece cuánta interferencia se genera según la cercanía de las frecuencias empleadas. De este modo, no solo la distancia espacial entre sensores es relevante, sino también la asignación de frecuencias, lo que introduce una capa adicional de complejidad en la optimización.

El tercer componente evaluado es la superficie cubierta por el conjunto de sensores. Para calcular este valor, utilizamos un método de análisis por rejilla. El área total de despliegue se divide en una cuadrícula de 200 x 200 celdas, lo que equivale a 40000 celdas individuales. Cada sensor cubre un área circular en torno a su posición, cuyo radio depende directamente de su potencia de emisión. Para determinar cuántas celdas cubre un sensor, el algoritmo recorre todas las celdas y mide la distancia desde el centro de la celda hasta el sensor. Si esta distancia es menor o igual al radio del sensor, se marca la celda como cubierta. Al final, sumando todas las celdas marcadas por cualquier sensor, se obtiene la superficie total cubierta por el individuo. Este método de evaluación exhaustivo garantiza que se capturen los solapamientos y vacíos en la cobertura de

manera realista, aunque supone un coste computacional muy elevado, ya que implica evaluar potencialmente millones de distancias por cada individuo evaluado.

Una vez calculados estos tres valores, se combina todo en un único valor de fitness. La fórmula de fitness asigna un peso positivo a la superficie cubierta, mientras que penaliza el consumo energético y la interferencia multiplicándolos por factores de penalización (λ_1 y λ_2 respectivamente). El fitness obtenido se normaliza dividiéndolo entre el número total de celdas para facilitar la comparación entre simulaciones y la comprensión de los resultados. De este modo, un fitness alto implica una cobertura amplia con un coste energético razonable y niveles de interferencia bajos, lo que representa la configuración deseable de la red.

Es fundamental destacar que la forma en la que se calculan la interferencia y la cobertura ha sido clave en el desarrollo del algoritmo, y ha determinado muchas decisiones posteriores. El hecho de no contar interferencias entre sensores alejados y el modelado detallado de la cobertura mediante rejilla han permitido que el algoritmo se comporte de forma física realista, aunque a costa de un elevado tiempo de cómputo, como se discutirá más adelante.

Además, durante esta inicialización también se guarda en el archivo `simulacionX_sensores.csv` toda la información de los sensores correspondientes a la generación cero, de modo que posteriormente podamos comparar la distribución espacial inicial con la distribución tras la evolución.

Con la población inicial completamente creada y evaluada, el programa entra en el núcleo de la simulación: el bucle de evolución.

En cada generación, para cada individuo de la nueva población se seleccionan dos padres mediante selección por torneo. En este método de selección, se escogen aleatoriamente tres individuos de la población actual y se selecciona como padre aquel que tenga el mejor fitness de entre los tres. De este modo se favorece la supervivencia de las mejores configuraciones, pero sin eliminar del todo la diversidad genética. Una vez seleccionados los padres, se genera un hijo combinando aleatoriamente sensores de ambos progenitores. Esta operación de recombinación se realiza en la función “`cruzar_individuos`”, donde para cada sensor se elige aleatoriamente si se copia del padre 1 o del padre 2. Posteriormente, el hijo sufre una mutación, aplicada mediante la función “`mutar_individuo`”, donde cada sensor tiene una probabilidad de alterar ligeramente su posición, su potencia de transmisión o su frecuencia de trabajo, dentro de márgenes controlados. Este proceso introduce pequeñas variaciones aleatorias que permiten explorar nuevos espacios de soluciones en generaciones futuras.

Después de crear y mutar la nueva población completa, cada individuo se evalúa de nuevo recalculando cobertura, energía e interferencia con las configuraciones recién generadas. Tras la evaluación, se reemplaza la población antigua por la nueva, asegurando que el proceso evolutivo avanza generación a generación.

En cada generación también se calculan y registran estadísticas clave: el mejor fitness individual alcanzado en la población, la media del fitness de todos los individuos, la media de superficie cubierta y la media de potencia por sensor. Todos estos valores se almacenan en el archivo `simulacionX_datos.csv`, permitiendo un posterior análisis exhaustivo de la evolución del sistema. Paralelamente, estos datos se muestran por pantalla en cada generación, permitiendo observar en tiempo real cómo evoluciona la calidad de la población.

Respecto al criterio de parada o convergencia, se utiliza un control basado en la fitness media de las generaciones. Se mantiene un historial de la fitness media de las últimas 5 generaciones y, si la mejora entre la media más antigua y la más reciente es inferior a un umbral preestablecido por nosotros (en este caso de 0.001), se considera que el algoritmo ha convergido y se detiene antes de alcanzar el número máximo de generaciones. Este enfoque asegura que no se desperdicien ciclos de computación si la evolución ha llegado a una meseta de rendimiento.

Finalmente, tras finalizar la evolución, se guarda también en `simulacionX_sensores.csv` toda la información de los sensores correspondientes a la última generación, permitiendo contrastar visualmente la evolución espacial de la red de sensores desde su configuración inicial hasta la optimizada.

Con esta estructura modular, el algoritmo proporciona una solución completa y eficiente al problema planteado, permitiendo además un análisis pormenorizado de su evolución a través de los datos exportados.

Problemas que han surgido:

Durante el desarrollo e implementación del algoritmo evolutivo, surgieron múltiples desafíos que exigieron ajustes tanto en la configuración de los parámetros como en el propio diseño de algunas funciones. A continuación, se describen detalladamente los principales problemas encontrados, así como las soluciones aplicadas en cada caso.

1. Tamaño de la rejilla y rango de potencias:

Inicialmente, la rejilla de análisis utilizada para calcular la cobertura era demasiado pequeña (100x100) y un ancho de 10 unidades lo que implicaba celdas de gran tamaño. Esto hacía que los sensores cubrieran de manera irreal áreas demasiado extensas o, si las potencias eran bajas, apenas cubrieran nada. La correspondencia física entre el tamaño de celda y el radio de los sensores no era coherente.

Para abordar este problema se incrementó el número de celdas de la rejilla a 200x200 así como el ancho de cada celda, reduciendo así el tamaño de cada celda y permitiendo un análisis espacial mucho más preciso. Además, se ajustó el rango de potencias iniciales de los sensores (de un rango mayor como 1 a 10 unidades, a uno entre 1.5 y 4.0 unidades) para que el radio de cobertura de cada sensor fuese proporcional al tamaño de las celdas, obteniendo así coberturas de aproximadamente un 90–92% del área total, valores totalmente razonables para un despliegue eficiente.

2. Cálculo de interferencias poco realista:

En una versión inicial, el cálculo de interferencias consideraba todos los pares de sensores, independientemente de la distancia que los separara. Esto generaba valores de interferencia muy elevados e irreales, ya que sensores muy lejanos (sin solapamiento real) seguían contribuyendo negativamente.

En este caso, se introdujo un umbral de interferencia y solo se consideran las interferencias entre sensores cuya distancia es inferior a este umbral. Esto simula de manera mucho más realista el comportamiento físico de este tipo de señales.

3. Ajuste de los factores de penalización (λ_{E} y λ_{I}):

Una vez corregido el cálculo de la interferencia, los valores numéricos de esta disminuyeron drásticamente (de valores de ~100000 a ~3000). Sin embargo, los factores de penalización seguían pensados para las magnitudes antiguas, por lo que su impacto sobre el fitness era inapropiado.

Debimos reescalar los factores de penalización a valores de $\lambda_{\text{E}} = 0.05$ para el consumo energético y $\lambda_{\text{I}} = 1.2$ para la interferencia. Este ajuste refleja la importancia relativa de cada parámetro: el consumo energético afecta a largo plazo (se penaliza moderadamente) mientras que la interferencia puede romper inmediatamente las comunicaciones (se penaliza fuertemente).

4. Criterio de parada por convergencia defectuoso:

Inicialmente, el criterio de parada del algoritmo por convergencia se basaba en analizar si la mejor fitness de la población apenas mejoraba durante varias generaciones. Sin embargo, este criterio era insuficiente, ya que una sola solución buena podía mantenerse constante mientras la calidad media de la población seguía mejorando.

En consecuencia, se modificó el criterio de convergencia para basarlo en la media de fitness de toda la población de cada generación. Ahora, el algoritmo solo se detiene cuando la media de fitness apenas mejora a lo largo de varias generaciones consecutivas. Esto garantiza que la mejora poblacional global se aprovecha hasta estabilizarse verdaderamente.

Posibles mejoras a futuro:

Aunque el algoritmo actual ofrece resultados sólidos y una evolución realista, existen algunas áreas donde podrían aplicarse mejoras para aumentar su eficiencia o calidad:

- **Optimización del cálculo de superficie / Reducción del tiempo de ejecución:** Actualmente, el cuello de botella del programa es la función “calcular_superficie”, debido a que evalúa el radio de cada sensor contra las 40000 celdas del área. Esto genera aproximadamente 40 millones de operaciones por individuo, por generación. Con 20 individuos y 50 generaciones, se alcanzan cifras del orden de miles de millones de comprobaciones.
- **Mutaciones dinámicas:** Se podría implementar un esquema de mutaciones dinámicas, donde las mutaciones fueran más agresivas al principio (favoreciendo la exploración del espacio de soluciones) y más suaves conforme avanza la evolución (favoreciendo la explotación y refinamiento final).
- **Adaptación de la selección por torneo:** Actualmente el torneo es de tamaño fijo ($k = 3$). Se podría explorar la idea de usar un torneo pequeño al inicio para favorecer la diversidad y aumentar progresivamente su tamaño para seleccionar individuos más elitistas conforme se busca convergencia.

Resultados:

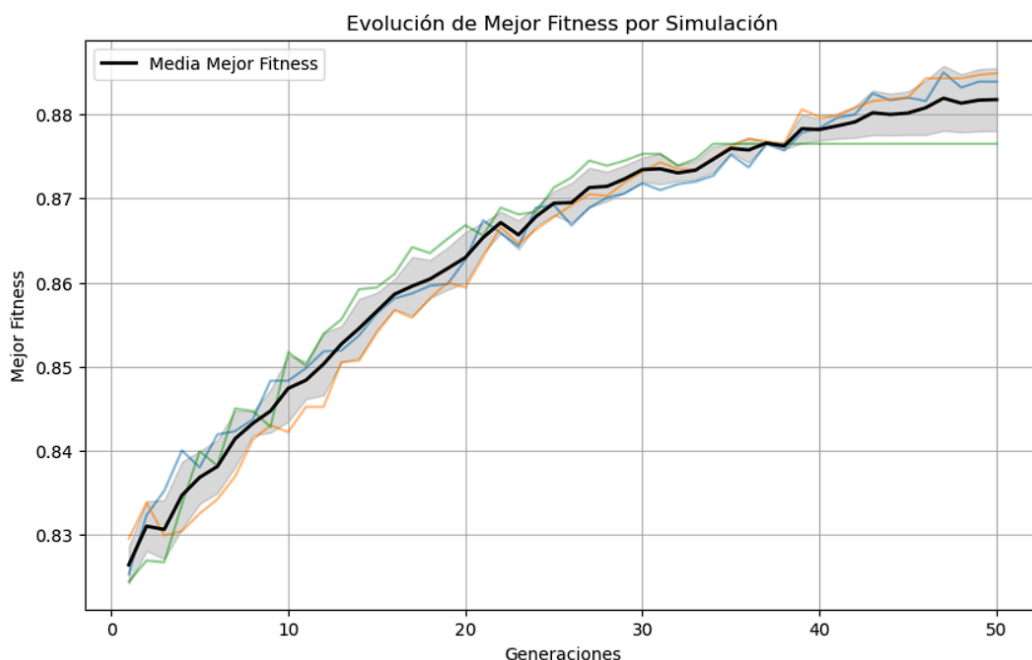
En esta sección se presentan y analizan los resultados obtenidos a partir de las distintas simulaciones realizadas con el algoritmo evolutivo desarrollado. Se mostrarán las gráficas de evolución de la fitness, tanto de la mejor fitness como de la fitness media de la población, así como la evolución de otros parámetros relevantes como la superficie cubierta y la potencia media de los sensores. Además, se ilustrará la distribución espacial de los sensores antes y después del proceso evolutivo, permitiendo visualizar de manera intuitiva la mejora alcanzada. Cada resultado irá acompañado de su interpretación y análisis detallado para facilitar la comprensión del comportamiento del sistema.

Evolución de la mejor fitness:

En la primera gráfica se representa la evolución de la mejor fitness alcanzada en cada generación a lo largo de las distintas simulaciones independientes. Cada línea de color fino representa la mejor fitness obtenida en una simulación individual. Además, la línea negra gruesa muestra la media de la mejor fitness en cada generación, calculada a partir de todas las simulaciones. La zona sombreada alrededor de esta línea indica la variabilidad entre simulaciones, en concreto, una banda de una desviación estándar por encima y por debajo de la media.

Podemos observar que, aunque hay ligeras diferencias entre simulaciones debido a la naturaleza aleatoria del algoritmo evolutivo, el comportamiento global es muy coherente. Todas las simulaciones siguen un patrón similar: al principio el fitness mejora rápidamente, y conforme avanzan las generaciones la mejora se ralentiza, siguiendo una típica curva logarítmica de evolución.

Es importante destacar que, en caso de que alguna simulación haya convergido antes de alcanzar las 50 generaciones (como es el caso de dos de ellas en la 37 y 47 respectivamente), los últimos valores se han rellenado repetidamente con el valor de convergencia, para mantener la longitud uniforme en todas las simulaciones. Esto se debe tener en cuenta a la hora de interpretar el tramo final de las curvas.



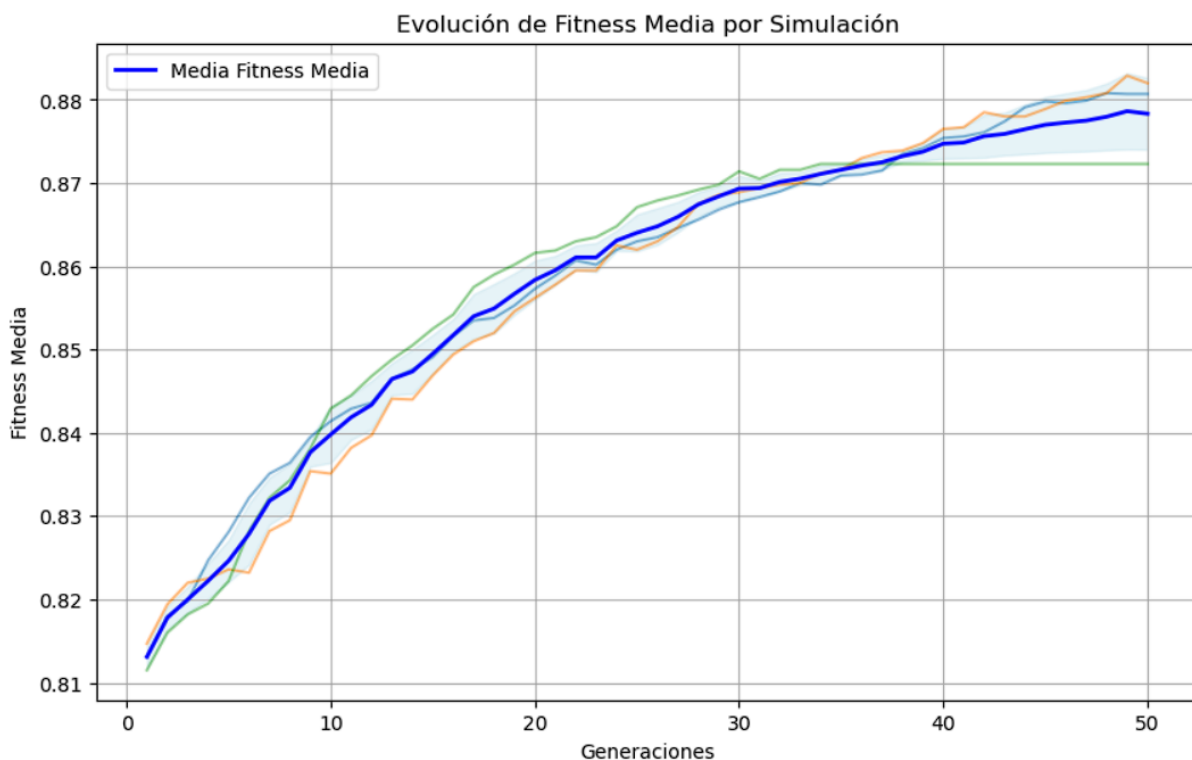
Evolución de la fitness media:

La segunda gráfica muestra la evolución de la fitness media de toda la población en cada generación, para cada simulación. De nuevo, cada línea de color representa una simulación individual, la línea azul gruesa representa la media de las fitness medias en cada generación, y la zona sombreada azul claro indica la dispersión de las simulaciones.

A diferencia de la evolución de la mejor fitness, la curva de fitness media presenta una progresión mucho más suave y regular. Esto es esperado, ya que la media de fitness es menos sensible a mutaciones puntuales o mejoras aisladas, reflejando la tendencia global de toda la población, y amortiguando los efectos de pequeños picos o caídas que sí afectan al mejor individuo.

Esta gráfica es especialmente útil porque refleja la mejora global de la población, no solo de los mejores individuos, lo que confirma la eficacia del proceso evolutivo implementado..

Al igual que en el caso anterior, a partir de la generación en la que se detecta convergencia, los valores se han mantenido constantes para completar las 50 generaciones.



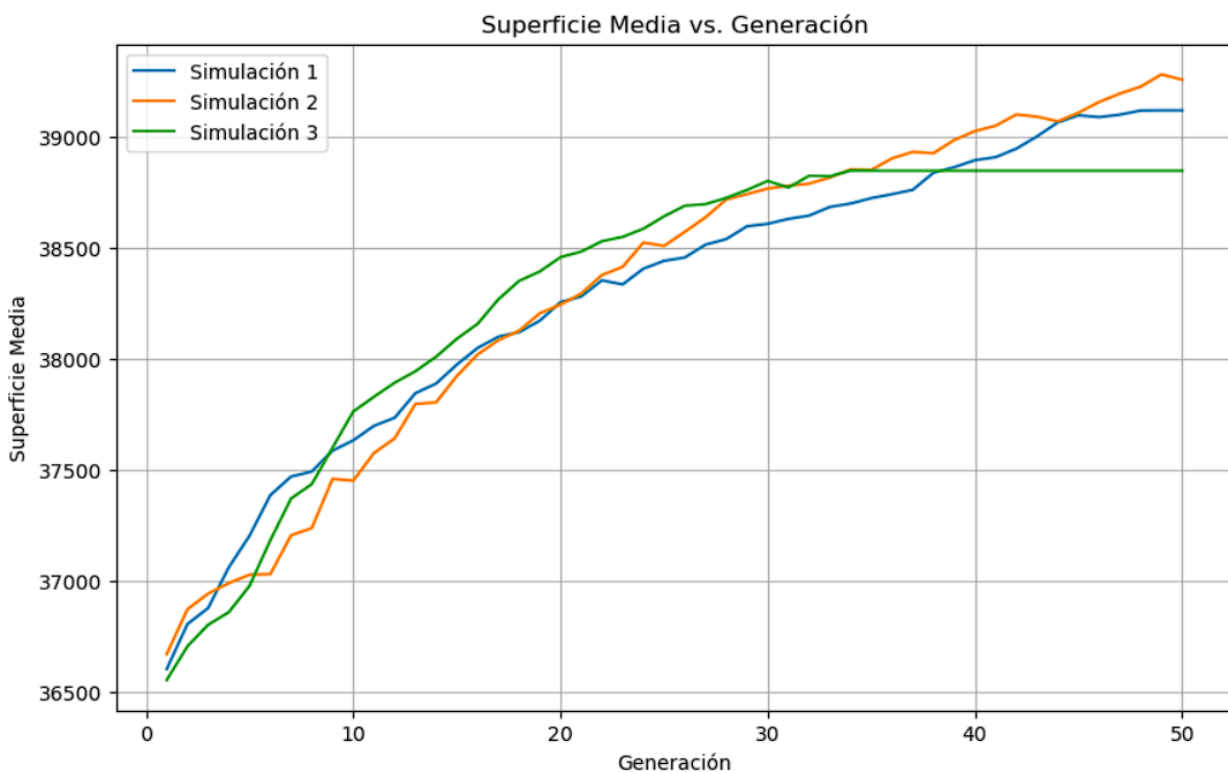
Evolución de la superficie media cubierta:

Esta gráfica representa la evolución de la superficie cubierta (en número de celdas) a lo largo de las generaciones.

Vemos que en todas las simulaciones la superficie cubierta aumenta generación tras generación, confirmando que el proceso evolutivo está encontrando configuraciones más eficientes en cuanto a cobertura.

El patrón esperado es un aumento progresivo de la superficie cubierta, hasta alcanzar un máximo cercano al límite físico impuesto por las dimensiones del área y el rango de potencias de los sensores. Observaremos más adelante cómo las simulaciones buscan mejorar no solo la cobertura inicial, sino también minimizar solapamientos innecesarios, reorganizando los sensores de manera más eficiente.

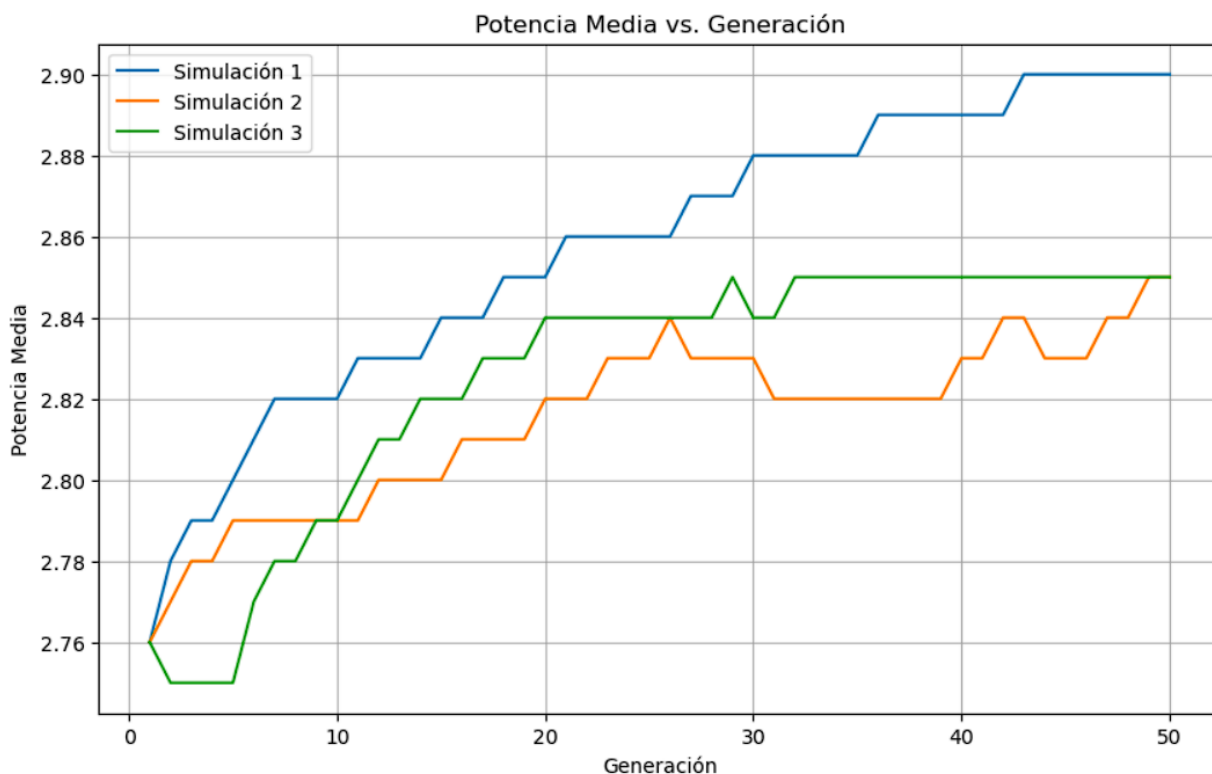
Se parte de coberturas iniciales en torno a 36.500–37.000 celdas cubiertas y se alcanza una cobertura final próxima a las 39.000–39.200 celdas de un total de 40.000 posibles, lo que representa una cobertura superior al 97% del área disponible. Además, la evolución es bastante progresiva y estable, sin caídas bruscas, indicando que las nuevas configuraciones son generalmente mejores o al menos no peores que las anteriores.



Evolución de la potencia media:

En esta gráfica observamos la evolución de la potencia media de los sensores en la población a lo largo del tiempo. Se aprecia un ligero pero constante aumento de la potencia media durante el proceso evolutivo.

Este incremento está en línea con lo esperado: para conseguir una mejor cobertura y reducir la interferencia, el algoritmo evolutivo tiende a seleccionar sensores que emiten con algo más de potencia, dentro de los límites permitidos. Sin embargo, gracias al ajuste de los coeficientes λ_{\square} y λ_{\square} , el incremento de potencia no es excesivo, y se controla de forma razonada para no comprometer la eficiencia energética.



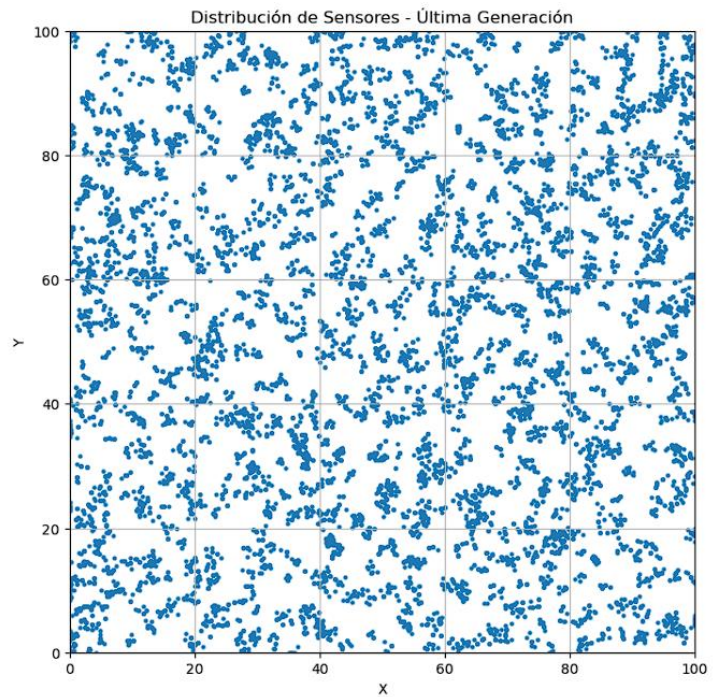
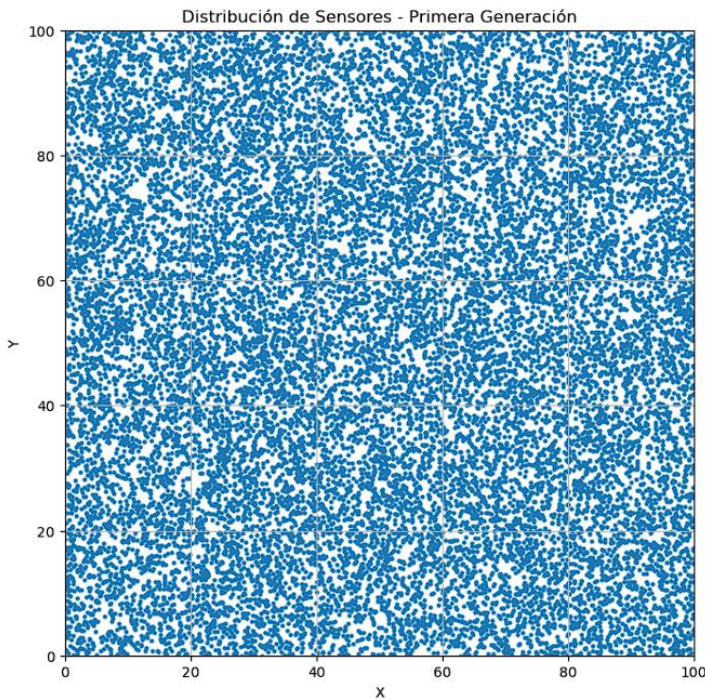
Distribución espacial de sensores:

Finalmente, se presentan las distribuciones espaciales de los sensores tanto en la primera generación como en la última generación de la evolución.

En la distribución de la primera generación (distribución inicial aleatoria) se aprecia un patrón completamente aleatorio, como era de esperar, dado que los sensores fueron colocados de manera uniforme y sin optimización alguna. Hay numerosas zonas donde los sensores están muy juntos y otras zonas donde se observan vacíos considerables, lo cual implica solapamientos innecesarios y zonas desatendidas.

En la última generación, después del proceso evolutivo, la distribución de sensores es visiblemente más estructurada. Aunque sigue existiendo aleatoriedad natural, se observa una tendencia a repartir los sensores de forma más homogénea, maximizando el área cubierta y reduciendo las zonas de solapamiento excesivo.

Esto confirma que el algoritmo no solo ha mejorado métricas globales como fitness o superficie cubierta, sino también la propia estructura espacial de la red de sensores.



Conclusión y Discusión de Resultados:

En este proyecto se ha abordado el diseño, implementación y análisis de un algoritmo evolutivo para optimizar la distribución de una red de sensores en un área determinada, considerando múltiples factores relevantes como la cobertura, el consumo energético y la interferencia entre sensores.

Tras el diseño inicial del modelo, se ajustaron cuidadosamente los parámetros de evaluación y evolución, como los coeficientes de penalización energética e interferencial (λ_{\square} y λ_{\square}), el tamaño de población, el rango de potencias, y la granularidad de la rejilla para evaluar la cobertura. Estas decisiones, tomadas de forma razonada y contrastada mediante múltiples pruebas, han permitido alcanzar un comportamiento del algoritmo coherente y estable.

La evolución de la disposición de los sensores ha seguido un patrón muy característico de los procesos evolutivos naturales: una rápida mejora inicial en las primeras generaciones, seguida de una estabilización progresiva hacia una meseta óptima. El fitness inicial en torno a 0.82 ha ido mejorando generación tras generación hasta estabilizarse aproximadamente entre 0.88–0.885, con coberturas superiores al 97% del área total, y un consumo energético controlado.

Respecto al papel de los operadores evolutivos, se ha observado que las mutaciones y el cruce han sido fundamentales para la mejora del sistema. El cruce ha permitido combinar características de distintos buenos individuos para crear soluciones potencialmente mejores, mientras que las mutaciones han introducido la variabilidad necesaria para evitar estancamientos y seguir explorando nuevas configuraciones del espacio de búsqueda. Aunque las mutaciones fueron constantes durante toda la evolución, una mutación más agresiva al principio podría acelerar aún más la convergencia, como se plantea en las posibles mejoras futuras.

En cuanto a la selección, el método de selección por torneo utilizado ha demostrado ser el más eficiente en este caso frente a alternativas clásicas como la selección por ruleta. A diferencia de esta que puede dar demasiado peso a individuos muy buenos desde el principio y provocar una pérdida temprana de diversidad genética, el torneo introduce una presión selectiva más controlada, equilibrando adecuadamente la explotación de los mejores individuos con la exploración de otros potencialmente útiles. Gracias a este método, hemos conseguido una evolución estable, sin caídas abruptas de calidad ni estancamientos prematuros como reflejan las gráficas de evolución de la fitness. Además, una posible mejora futura sería adaptar dinámicamente el tamaño del torneo: comenzar con torneos pequeños para favorecer más exploración en fases iniciales y aumentarlo progresivamente para reforzar la explotación en fases finales, acelerando así la convergencia hacia soluciones óptimas.

Analizando la cantidad de generaciones necesarias para lograr una solución estable, hemos comprobado que la convergencia se alcanza de forma consistente alrededor de la generación 45-47 en todas las simulaciones realizadas. A partir de ese punto, las mejoras en fitness, cobertura o energía son mínimas, indicando que el algoritmo ha encontrado configuraciones cercanas al óptimo local.

Conclusión Final:

Por todo esto, podemos afirmar que el algoritmo evolutivo desarrollado cumple de forma sobresaliente con los objetivos planteados, optimizando la distribución de sensores en

condiciones realistas y logrando un equilibrio efectivo entre cobertura, eficiencia energética y calidad de red. El modelo evoluciona de manera consistente, mejora progresivamente generación tras generación, y alcanza una convergencia lógica y natural, todo ello respaldado por un proceso de diseño riguroso y un análisis detallado de resultados.

En definitiva, el sistema implementado demuestra ser una herramienta eficaz y robusta para abordar problemas complejos de optimización multiobjetivo mediante técnicas evolutivas.