Reporte Final (Semana 2)

ICI2241-1 – Análisis y Diseño de Algoritmos

José Lara Arce

▼ Problema descrito

Optimización de la Distribución de Antenas de Telecomunicaciones

Imagina que trabajas para una empresa de telecomunicaciones que está planeando instalar antenas de telefonía móvil en una determinada área urbana. El objetivo es maximizar la cobertura de la señal mientras se minimiza el costo total de instalación de las antenas. El área se representa como un plano 2D, y cada antena tiene un rango de cobertura circular alrededor de su ubicación.

El problema se reduce a encontrar las posiciones óptimas para ubicar las antenas de manera que se cubra la mayor cantidad posible de área y se minimice la cantidad de antenas necesarias.

▼ Algoritmo

▼ Introducción

El Algoritmo de Optimización por Enjambre de Partículas (PSO) es un método de optimización inspirado en el comportamiento social de los animales, como los enjambres de aves o cardúmenes de

peces. En PSO, se utiliza una población de "partículas" que se mueven a través de un espacio de búsqueda en busca de soluciones óptimas.

▼ Entradas, salidas y restricciones

1. Entradas:

- **Área Geográfica**: Se proporciona un área geográfica representada como un plano 2D donde se planea instalar antenas de telecomunicaciones.
- **Número de Antenas**: El número máximo de antenas que se pueden instalar.
- **Radio de Cobertura**: El radio máximo de cobertura de una antena, que define el área en la que puede proporcionar señal.
- **Función Objetivo**: Una función que combina la cobertura de la señal y el costo total de instalación de las antenas. Puede tener diferentes ponderaciones para cobertura y costo según los requisitos.

2. Restricciones:

- **Límites de Posición**: Las antenas deben ubicarse dentro de los límites del área geográfica.
- **Número Máximo de Antenas**: El número total de antenas instaladas no puede exceder el número máximo especificado.
- **Superposición de Cobertura**: Las áreas de cobertura de diferentes antenas no deben superponerse para evitar interferencias y duplicaciones innecesarias.

3. Salidas:

- **Posiciones Óptimas de Antenas**: El algoritmo debe devolver las posiciones óptimas para ubicar las antenas, de modo que se maximice la cobertura de señal y se minimice el costo total.
- **Métricas de Evaluación**: Junto con las posiciones de las antenas, es útil proporcionar información sobre la calidad de la solución, como la cobertura total del área y el costo total estimado.

▼ Explicación paso a paso del algoritmo

a. **Inicialización**:

- Cada partícula representa una distribución potencial de antenas en el área 2D.
- Las posiciones iniciales de las partículas se generan aleatoriamente, representando las ubicaciones iniciales de las antenas.
- Se establece la mejor posición personal ("pbest") de cada partícula como su distribución inicial.
- Se encuentra la mejor posición global ("gbest") entre las distribuciones iniciales.

b. **Evaluación de la Función Objetivo**:

- La función objetivo considera tanto la cobertura de la señal como el costo total de instalación de las antenas.
- Se calcula la cobertura total del área, que puede ser una medida de la superposición de las áreas de cobertura de todas las antenas.

- Se evalúa el costo total estimado de instalación de las antenas, que podría depender de factores como la cantidad de antenas y las distancias de instalación.

c. **Actualización de Posición y Velocidad**:

- Las partículas ajustan su velocidad en función de las diferencias entre sus distribuciones actuales, sus "pbest" y el "gbest".
- La velocidad controla cómo las antenas se mueven en el espacio 2D.
- Las nuevas posiciones de las antenas se calculan sumando su velocidad a sus posiciones actuales, asegurando que las antenas se mantengan dentro de los límites del área.

d. **Repetición**:

- Los pasos b y c se repiten durante un número de iteraciones definido previamente.

▼ Ejemplo

Entradas:

- Número Máximo de Antenas: 5
- Área de la Ciudad: 10x10 unidades
- Área de Cobertura de Antena: Radio de 2 unidades
- Función de Costo: *Costo*=Distancia al centro de la ciudad

  Costo = Distancia al centro de la ciudad

Ejemplo de Posiciones Finales (Resultados Aproximados):

Después de ejecutar varias iteraciones del algoritmo, podríamos obtener posiciones finales aproximadas para las antenas, como sigue:

1. Antena 1: Coordenadas (3, 4)
2. Antena 2: Coordenadas (7, 6)
3. Antena 3: Coordenadas (2, 8)
4. Antena 4: Coordenadas (5, 2)
5. Antena 5: Coordenadas (9, 3)

Hay que tener en cuenta que estas posiciones finales son solo un ejemplo ficticio y aproximado. En una implementación real, el algoritmo convergería hacia soluciones que optimicen la cobertura de señal y minimicen el costo total de instalación, pero las posiciones específicas de las antenas pueden variar debido a la naturaleza estocástica del algoritmo.

▼ Cotas

Las cotas asintóticas del Algoritmo de Optimización por Enjambre de Partículas (PSO) pueden ser un tanto complicadas de calcular de manera precisa debido a la naturaleza estocástica y dependiente del problema del algoritmo. Sin embargo, podemos analizar algunas propiedades generales de las operaciones clave involucradas en PSO para estimar las cotas asintóticas.

▼ **Cota superior**

Dado que las operaciones en PSO involucran la evaluación de funciones objetivo, cálculos de velocidad y actualización de posiciones, es difícil proporcionar una cota superior precisa. En general, la cota superior puede ser O(N * T * (f(n) + m * g(m)))

- *N* es el número de partículas.
- *T* es el número de iteraciones.
- *f*(*n*) es la complejidad de la función objetivo.
- *m* es la dimensión del problema.
- *g*(*m*) es la complejidad de cálculos para actualizar partículas.

▼ **Cota inferior**

Determinar una cota inferior precisa para PSO también es complejo debido a su naturaleza estocástica. Por lo general, se asume que la cota inferior es mayor o igual a la cota inferior de la función objetivo. En términos generales, la cota inferior podría ser

$$O(f_{\inf}(n))$$

Esto representa la cota inferior de complejidad temporal basada en la función objetivo:

$$f_{\inf}(n)$$

▼ **Cota ajustada**

Es posible que la cota ajustada sea igual o muy similar a la cota superior en la notación O grande. Esto se debe a que la cota ajustada intenta considerar tanto los peores casos como las mejores estimaciones posibles del rendimiento del algoritmo.

La cota superior O(N * T * (f(n) + m * g(m))) ya tiene en cuenta tanto la función objetivo como las operaciones de actualización de partículas. Debido a la variabilidad y la naturaleza estocástica de PSO, la cota ajustada podría reflejar un rango amplio de posibilidades, pero podría terminar siendo muy similar o igual a la cota superior.

En este caso, para simplificar el análisis y la representación de la complejidad asintótica del algoritmo PSO, es razonable considerar que la cota ajustada y la cota superior son equivalentes. Esto no significa que sean idénticas en todos los contextos, sino que en el caso de PSO, la complejidad puede aproximarse de manera similar desde ambas perspectivas.

▼ **Operaciones clave**

- **Inicialización de partículas**: La inicialización de las posiciones y velocidades de las partículas requiere asignaciones y operaciones de generación aleatoria. Esto generalmente tiene un costo constante *O*(1) por partícula.
- **Evaluación de la función objetivo**: La evaluación de la función objetivo en cada partícula en cada iteración es una operación costosa. Si la función objetivo requiere *O*(*f*(*n*)) operaciones, la evaluación total será *O*(*N*·*T*·*f*(*n*)), donde *N* es el número de partículas y *T* es el número de iteraciones.

- **Actualización de velocidades y posiciones**: La actualización de las velocidades y posiciones de las partículas implica cálculos basados en las posiciones actuales, mejores posiciones históricas y mejores posiciones globales. Estos cálculos están influenciados por la dimensión m del problema y pueden realizarse en $O(m)$ operaciones por partícula. Por lo tanto, para N partículas y T iteraciones, la complejidad sería $O(N \cdot T \cdot m)$.

Teniendo en cuenta estas operaciones clave, la complejidad asintótica del algoritmo PSO puede ser estimada utilizando la notación O grande como mencionamos previamente.

▼ Propiedad invariante

En el contexto general del algoritmo PSO, una propiedad invariante podría estar relacionada con la convergencia del enjambre hacia una solución óptima:

La propiedad invariante establece que a lo largo de las iteraciones del algoritmo PSO, la dispersión de las partículas en el espacio de búsqueda se reduce de manera gradual, lo que indica su convergencia hacia soluciones óptimas.

▼ Demostración

1. **Caso Base - Antes de iniciar el algoritmo**:
   Antes de iniciar las iteraciones del algoritmo PSO, las partículas se posicionan en ubicaciones aleatorias en el espacio de búsqueda. Dado que estas posiciones iniciales son seleccionadas de manera completamente aleatoria, es natural esperar que las partículas se encuentren dispersas en diversas áreas del espacio de búsqueda. En este punto, la dispersión de las partículas es alta debido a la variabilidad inherente en las posiciones iniciales. Por lo tanto, la propiedad invariante, que describe la reducción gradual de esta dispersión, ya está satisfecha antes de que el algoritmo comience su proceso iterativo. Este hecho es fundamental para el análisis, ya que establece una base desde la cual las partículas pueden comenzar a explorar y converger hacia regiones más prometedoras a medida que avanza el algoritmo.

2. **Mantenimiento invariante - En cada iteración**:

   Supongamos que al final de la iteración '***i'***, la propiedad invariante se cumple, lo que significa que la dispersión de las partículas ha disminuido en comparación con el inicio de la iteración '***i'***. Durante la siguiente iteración, '***i + 1'***, las partículas se someten a un proceso de actualización utilizando información local (la mejor posición personal) y global (la mejor posición global en todo el enjambre). Estas actualizaciones son esenciales para el mecanismo de búsqueda y convergencia del algoritmo PSO.

   Las partículas ajustan su trayectoria y velocidad basándose en estas mejores posiciones conocidas. Esta influencia combinada de información local y global actúa como un guía hacia áreas del espacio de búsqueda que han demostrado contener soluciones prometedoras. A medida que las partículas avanzan en esta iteración, su concentración en estas regiones más prometedoras provoca una disminución adicional de la dispersión en comparación con el estado al inicio de la iteración.

   Esta progresión iterativa, donde las partículas se ajustan y reajustan en función de información local y global, refleja la convergencia gradual del enjambre hacia soluciones óptimas o cercanas a óptimas. Por lo tanto, al finalizar la iteración '***i + 1'***, la propiedad invariante, que enfatiza la

disminución continua de la dispersión, permanece inalterada, fortaleciendo la confianza en que el algoritmo PSO está avanzando en la dirección correcta hacia soluciones de mayor calidad.

3. **Condición de terminación - al finalizar la ejecución**:
   Después de un número suficiente de iteraciones, las partículas habrán tenido la oportunidad de explorar y converger hacia regiones prometedoras del espacio de búsqueda.

   Si esta propiedad se mantiene constantemente en todas las iteraciones y al final de la ejecución del algoritmo, podemos concluir con confianza lo siguiente:

   1. **Logro del Objetivo de Optimización**: La convergencia gradual de las partículas hacia regiones prometedoras demuestra que el algoritmo está cumpliendo su objetivo fundamental de encontrar soluciones óptimas o cercanas a óptimas en el espacio de búsqueda.
   2. **Comportamiento Consistente y Coherente**: La propiedad invariante brinda una forma concreta de evaluar si el algoritmo sigue una estrategia lógica y coherente en su búsqueda de soluciones. La continua reducción de la dispersión respalda la idea de que el algoritmo está tomando decisiones inteligentes en cada iteración para moverse hacia soluciones más prometedoras.
   3. **Generación de Resultados Confiables**: Al demostrar que la propiedad invariante se mantiene a lo largo del tiempo, establecemos que el algoritmo PSO es confiable y consistente en su comportamiento. Esto sugiere que el algoritmo es robusto en términos de producir resultados consistentemente cercanos a soluciones óptimas.

La demostración se basa en la lógica de cómo las partículas se actualizan en cada iteración, lo que resulta en una reducción gradual de la dispersión y una convergencia hacia soluciones óptimas.

▼ Verificación de utilidad

La propiedad invariante establecida, que se refiere a la reducción gradual de la dispersión de las partículas en el espacio de búsqueda, es útil para demostrar la correctitud del algoritmo PSO en la optimización de problemas.

La utilidad de esta propiedad invariante radica en su relación directa con el objetivo del algoritmo PSO. El propósito fundamental del algoritmo es encontrar soluciones óptimas o cercanas a óptimas en el espacio de búsqueda. A través de la demostración de que la dispersión de las partículas se reduce en cada iteración, se establece que las partículas están convergiendo hacia regiones prometedoras. Esto sugiere que el algoritmo PSO está logrando su objetivo al moverse hacia soluciones potencialmente óptimas en el espacio de búsqueda.

La propiedad invariante es más que una simple observación sobre los datos; es un indicador clave de la mejora continua del algoritmo a medida que avanza en las iteraciones. Esta propiedad está estrechamente relacionada con la correctitud del algoritmo, ya que demuestra que el proceso de actualización de partículas en PSO está guiando de manera efectiva a las partículas hacia soluciones más prometedoras.

▼ Comprobación de correctitud

Utilizando la propiedad invariante que establece la reducción gradual de la dispersión de las partículas en el espacio de búsqueda, podemos concluir si el algoritmo PSO es correcto y resuelve el problema de optimización según las especificaciones.

La propiedad invariante nos dice que en cada iteración del algoritmo, la dispersión de las partículas se reduce o se mantiene sin cambios. Esto implica que las partículas están convergiendo gradualmente hacia regiones prometedoras del espacio de búsqueda. Si esta propiedad se mantiene durante todas las iteraciones y al final de la ejecución del algoritmo, podemos afirmar con confianza lo siguiente:

1. El algoritmo está logrando su objetivo de optimización: La convergencia de las partículas hacia regiones prometedoras indica que el algoritmo está identificando y moviéndose hacia soluciones óptimas o cercanas a óptimas en el espacio de búsqueda.
2. El algoritmo es correcto en términos de comportamiento: La propiedad invariante nos proporciona una forma concreta de evaluar si el algoritmo se comporta según lo esperado. Si la propiedad se cumple en todas las iteraciones y al final, esto demuestra que el algoritmo sigue una estrategia coherente y efectiva de mejora de soluciones.
3. El algoritmo está generando resultados confiables: Al demostrar que la propiedad invariante se mantiene, también demostramos que el algoritmo no introduce errores y produce resultados consistentes y confiables en términos de convergencia hacia soluciones prometedoras.

▼ Desempeño del algoritmo de Optimización por Enjambre de Partículas (PSO) en la búsqueda del mínimo de una función de grado máximo par

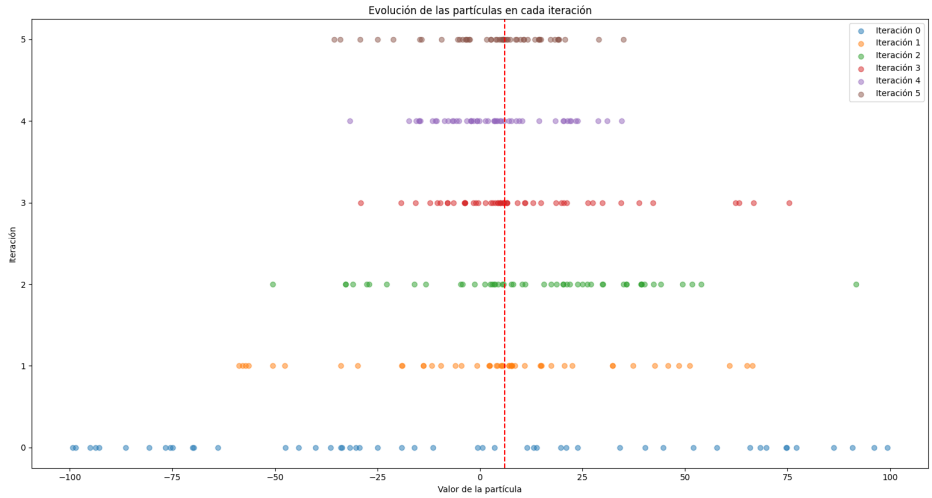
Sea una función de ejemplo:

$$f(x) = x^2 - 12x + 35$$

Abordaremos la optimización de la función cuadrática, en la que el valor mínimo se encuentra en x = 6. Utilizando el algoritmo de Optimización por Enjambre de Partículas (PSO). El objetivo es encontrar el valor mínimo de esta función, lo que nos permitirá evaluar la efectividad del algoritmo en un contexto más específico. Al aplicar el algoritmo PSO a esta función, podremos observar cómo las partículas convergen hacia el mínimo global de la función. Esto nos permitirá entender cómo el algoritmo aborda problemas con una función objetivo conocida y cómo se adapta para encontrar soluciones óptimas.

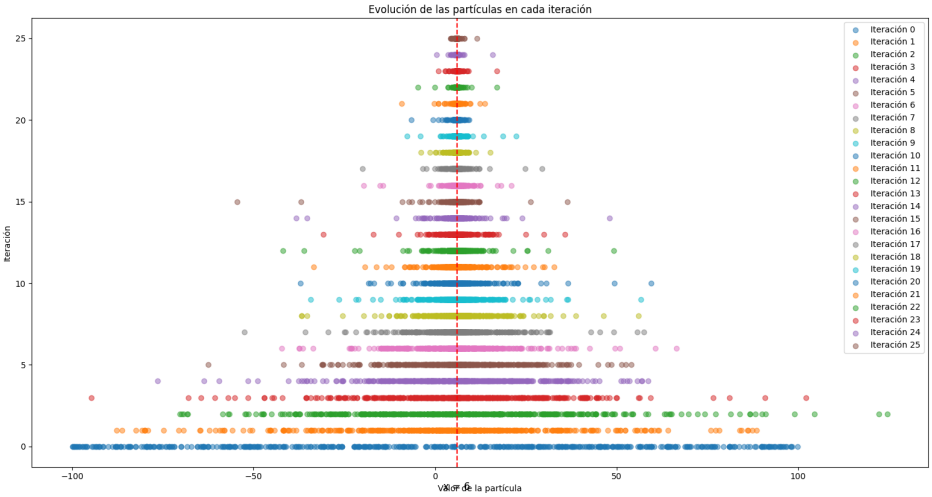
En esta fase de experimentación, nos sumergiremos en la exploración de cómo se comporta el algoritmo PSO a medida que ajustamos dos factores clave: la cantidad de partículas y la cantidad de iteraciones. Nuestro punto de partida será un escenario en el que generamos aleatoriamente 50 partículas con coordenadas en el rango de x=-100 a x=100. Además, ejecutaremos el algoritmo durante un total de 5 iteraciones en este punto inicial. El objetivo es observar cómo evolucionan las partículas en este corto período de tiempo y con una cantidad limitada de partículas e iteraciones. Posteriormente, ampliaremos nuestras investigaciones al aumentar tanto el número de partículas como la cantidad de iteraciones, con el fin de explorar cómo estos ajustes impactan en el rendimiento y la convergencia del algoritmo. Este enfoque nos proporcionará una visión más profunda de la dinámica del algoritmo PSO y cómo su eficacia puede variar en función de estos parámetros ajustables.

▼ Caso 1: 50 partículas y 5 iteraciones



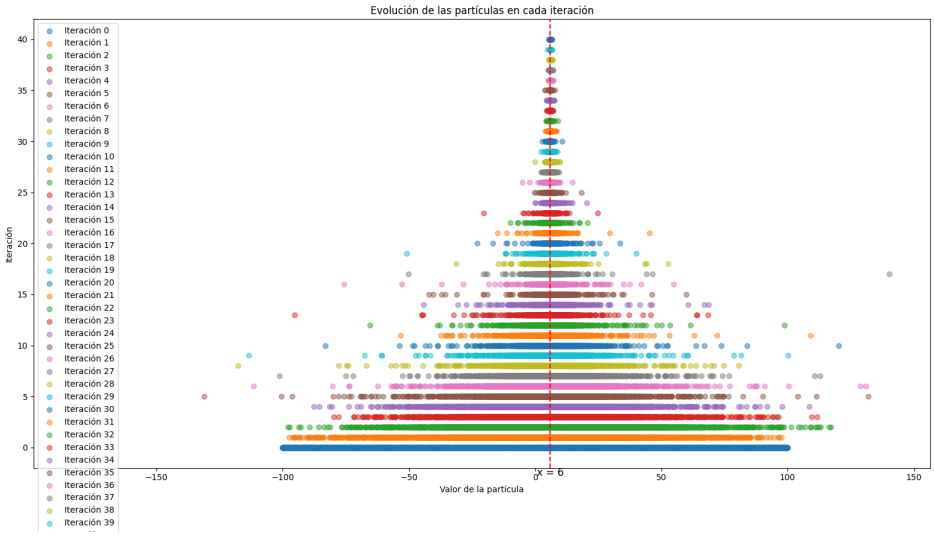
En el caso de prueba inicial con 50 partículas y 5 iteraciones, podemos observar una tendencia interesante en la gráfica de evolución de las partículas. A medida que avanzan las iteraciones, es posible apreciar una ligera disminución en la dispersión de las partículas en el espacio de búsqueda. Si bien aún no se aprecia una convergencia clara hacia el valor mínimo de la función objetivo en x=6, se evidencia un inicio de movimiento hacia una región más cercana a ese punto. Esta disminución de la dispersión sugiere que, incluso con un número limitado de partículas y un corto número de iteraciones, el algoritmo PSO está logrando influir en la dirección de las partículas, dirigiéndolas gradualmente hacia zonas más prometedoras en la búsqueda del mínimo global. Sin embargo, dada la naturaleza aleatoria del proceso y la limitada cantidad de iteraciones, todavía existe variabilidad en la trayectoria de las partículas.

▼ Caso 2: 500 partículas y 25 iteraciones



En el segundo caso de prueba, donde se emplearon 500 partículas y 25 iteraciones, se observa una evolución más pronunciada en la gráfica de las partículas. A medida que aumenta la cantidad de partículas y el número de iteraciones, se puede apreciar una tendencia más definida hacia la convergencia al valor mínimo de la función objetivo en x=6. Aunque la dispersión de las partículas se ha reducido significativamente en comparación con el caso anterior, aún se puede observar una cierta amplitud en la distribución. Esto sugiere que, a pesar de la convergencia hacia la región del mínimo global, algunas partículas todavía están explorando y ajustándose en torno a la solución óptima. La trayectoria hacia el valor mínimo de x=6 se torna más coherente y se destaca una mejora notable en la aproximación de las partículas hacia la solución deseada, demostrando cómo el algoritmo PSO está siendo más eficaz a medida que se aumenta tanto el número de partículas como de iteraciones.

▼ Caso 3: 5000 partículas y 40 iteraciones



En el tercer caso de prueba, se ha empleado un número significativamente mayor de partículas (5000) y se ha incrementado el número de iteraciones a 40. Al analizar la gráfica resultante, es evidente una evolución mucho más marcada y rápida hacia la convergencia al valor mínimo de la función objetivo en x=6. A partir de alrededor de la iteración 30, la dispersión de las partículas disminuye de manera drástica y se inicia un proceso de convergencia masiva. Esta tendencia se vuelve más notable conforme avanza el número de iteraciones, y hacia el final, prácticamente todas las partículas parecen estar agrupándose en torno al mismo punto, ilustrando una convergencia exitosa hacia el mínimo global. La trayectoria de las partículas se vuelve más uniforme y cohesionada, lo que sugiere que el algoritmo ha logrado que las partículas se ajusten cada vez más cerca del valor mínimo óptimo (x = 6). Este caso ejemplifica cómo aumentar tanto la cantidad de partículas como el número de iteraciones puede acelerar y mejorar significativamente la convergencia del algoritmo PSO hacia la solución deseada.

▼ Código

```
def optimizacionEnjambreParticulas(numParticulas, numIteraciones):
    particulas = np.random.uniform(-100, 100, size=numParticulas)
```

```
    velocidades = np.random.uniform(-1, 1, size=numParticulas)

    mejoresPosicionesPersonales = particulas.copy()
    mejorPosicionGlobal = particulas[np.argmin(funcionObjetivo(particulas))]

    evolucion_particulas = [particulas.copy()]

    for _ in range(numIteraciones):
        for i in range(numParticulas):
            actualizacionCognitiva = np.random.random() * (mejoresPosicionesPersonales[i] - particulas[i])
            actualizacionSocial = np.random.random() * (mejorPosicionGlobal - particulas[i])
            velocidades[i] = 0.5 * velocidades[i] + 1.5 * actualizacionCognitiva + 1.5 * actualizacionSocial
            particulas[i] = particulas[i] + velocidades[i]

            if funcionObjetivo(particulas[i]) < funcionObjetivo(mejoresPosicionesPersonales[i]):
                mejoresPosicionesPersonales[i] = particulas[i]
                if funcionObjetivo(particulas[i]) < funcionObjetivo(mejorPosicionGlobal):
                    mejorPosicionGlobal = particulas[i]
        evolucion_particulas.append(particulas.copy())

    return evolucion_particulas, mejorPosicionGlobal

# Parámetros de entrada
numParticulas = 5000
numIteraciones = 40
```

▼ Mejoras

Se mejoraron las explicaciones acerca de:

- **Caso base**
- **Mantenimiento invariante**
- **Condición de terminación - al finalizar la ejecución:**