



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico Nacional de México, campus Culiacán

Ingeniería en Sistemas Computacionales

Sensorem: Aplicación del algoritmo de Enjambre de Partículas para la distribución óptima de sensores de humedad en Guasave, Sinaloa.

Protocolo de investigación para Tópicos de Inteligencia Artificial

Integrantes del proyecto:

Payan Urquidez Rafael Alberto

Quiñonez Ramirez Nestor de Jesus

Docente: Mora Felix Zuriel Dathan

Repositorio:

<https://github.com/Norkat/Topicos-de-Inteligencia-Artificial>

Culiacán, Sinaloa a 4 Noviembre del 2025

1. Descripción del problema.....	3
1.1 Contexto y relevancia.....	3
1.2 Problema específico.....	3
1.3 Variables y restricciones del problema.....	3
1.4 Objetivo técnico.....	4
1.5 Valor de la solución.....	4
2. Justificación del uso de PSO.....	4
2.1 Fundamentación teórica.....	4
2.2 Adecuación del PSO al problema agrícola.....	5
2.3 Comparación con otros métodos.....	5
2.4 Aplicaciones exitosas del PSO en agricultura.....	5
3. Diseño del algoritmo.....	6
3.1 Representación del problema.....	6
3.2. Función Objetivo.....	6
3.3. Parámetros del Algoritmo.....	7
3.4. Pseudocódigo.....	7
3.5. Librerías y Frameworks.....	8
4. Resultados obtenidos.....	9
4.1 Evaluación general del algoritmo PSO.....	9
4.2 Resultados del caso 1: datos reales (100 cultivos).....	9
4.3 Resultados del caso 2: datos aleatorios (50 puntos).....	11
4.4 Comparación y análisis.....	12
5. Análisis de eficiencia.....	13
5.1 Eficiencia computacional.....	13
5.2 Precisión y convergencia.....	13
5.3 Robustez y estabilidad.....	13
5.4 Limitaciones observadas.....	14
5.5 Evaluación global.....	14
6. Conclusiones.....	14
Bibliografía.....	16

1. Descripción del problema

1.1 Contexto y relevancia

El agua es un recurso fundamental para la producción agrícola y su manejo eficiente resulta esencial para garantizar la sostenibilidad de los ecosistemas y la seguridad alimentaria. En regiones agrícolas como Guasave, Sinaloa, caracterizadas por su clima cálido y terreno predominantemente plano, el riego constituye la principal fuente de abastecimiento hídrico para cultivos como maíz, tomate y chile. Sin embargo, la sobreexplotación del recurso y las prácticas de riego tradicionales han derivado en problemas de salinidad y pérdida de materia orgánica en el suelo, reduciendo su fertilidad y la eficiencia del riego (SADER, 2024).

Según la Organización de las Naciones Unidas para la Alimentación y la Agricultura, “el sector agrícola representa alrededor del 70 % del consumo mundial de agua dulce, por lo que optimizar su uso es una necesidad urgente ante el cambio climático y el crecimiento poblacional” (FAO, 2023). En este contexto, la agricultura de precisión, apoyada en tecnologías como los sensores de humedad del suelo, ofrece una alternativa viable para mejorar la eficiencia hídrica, reducir costos de producción y aumentar el rendimiento de los cultivos (Gebbers & Adamchuk, 2010).

1.2 Problema específico

Uno de los desafíos principales de la agricultura de precisión consiste en determinar la ubicación óptima de los sensores de humedad. Si los sensores se colocan de manera aleatoria o empírica, pueden surgir zonas sin monitoreo o redundancia de información en áreas homogéneas. Esto conduce a una interpretación errónea del contenido hídrico del suelo y, en consecuencia, a un uso ineficiente del agua (Wang, 2023).

Como señalan Lloret, Sendra, García y Jiménez, “*una distribución no planificada de sensores puede provocar mediciones redundantes y pérdida de representatividad espacial, afectando el control del riego*” (Lloret et al., 2021). Por ello, es necesario modelar la ubicación de sensores como un problema de optimización espacial, en el cual se busca maximizar la cobertura del terreno con la menor cantidad posible de dispositivos, manteniendo una representación precisa de la humedad del suelo.

1.3 Variables y restricciones del problema

El problema de ubicación óptima de sensores puede definirse a partir de un conjunto de variables de decisión y parámetros ambientales. Entre las principales variables se encuentran:

- Coordenadas geoespaciales (latitud, longitud) de cada sensor.
- Humedad (%).
- Salinidad (dS/m).
- Temperatura (°C).

Los parámetros del entorno incluyen: humedad del suelo, salinidad, tipo de cultivo y temperatura promedio.

De acuerdo con Zhuo, Dai, Zhao y Han, *“la heterogeneidad espacial de la humedad del suelo dificulta el diseño de redes de sensores, ya que la mayoría de las redes in situ no proporcionan una cobertura suficiente para capturar las variaciones a escala de parcela”* (Zhuo et al., 2020), lo que quiere decir que estas variables presentadas son las que determinan las dificultades en la solución del problema y delimita la maximización del objetivo.

1.4 Objetivo técnico

El objetivo técnico consiste en formular y resolver un problema de optimización que, dados los atributos del terreno y los cultivos, determine la disposición y cantidad óptima de sensores de humedad. La meta es maximizar una función de cobertura informativa (es decir, la cantidad de variabilidad del terreno capturada por los sensores) y minimizar el error promedio de estimación de la humedad del suelo, bajo restricciones prácticas como presupuesto, accesibilidad y comunicación entre nodos.

1.5 Valor de la solución

La optimización de la ubicación de sensores aporta varios beneficios:

- Reducción del consumo de agua: al mejorar la precisión de las decisiones de riego basadas en datos representativos.
- Aumento de la productividad agrícola: al disminuir el estrés hídrico y mejorar la calidad del cultivo.
- Optimización de recursos tecnológicos: al reducir el número de sensores requeridos sin perder cobertura espacial.
- Base para sistemas inteligentes: al permitir la integración con modelos predictivos y sistemas automáticos de riego controlados por algoritmos de inteligencia artificial (Sharafkhaneh et al., 2024).

Como destaca Gutierrez, Mondragón, Colorado y Méndez, *“la disposición óptima de nodos sensores en cultivos agrícolas mejora la cobertura espacial de humedad y proporciona datos más confiables para la automatización del riego”* (Gutierrez et al., 2022).

2. Justificación del uso de PSO

2.1 Fundamentación teórica

El algoritmo de Enjambre de Partículas (Particle Swarm Optimization, PSO) es una técnica de optimización bioinspirada desarrollada por Kennedy y Eberhart en 1995. Está basada en el comportamiento colectivo de organismos como bandadas de aves o cardúmenes de peces, que cooperan para encontrar alimento o refugio (Kennedy & Eberhart, 1995). En este enfoque, cada partícula representa una posible solución dentro del espacio de búsqueda, y se desplaza en función tanto de su experiencia individual como de la información del grupo. Según Eberhart y Shi, *“el PSO puede considerarse un método estocástico de optimización global que combina el aprendizaje social y cognitivo, logrando convergencias rápidas en espacios continuos no lineales”* (Eberhart & Shi, 2001). Esta característica lo hace especialmente útil en problemas multidimensionales, no lineales y con múltiples óptimos

locales, como es el caso de la distribución óptima de sensores en terrenos agrícolas heterogéneos.

2.2 Adecuación del PSO al problema agrícola

En el contexto agrícola, el problema de ubicar sensores de humedad puede modelarse como una búsqueda continua de coordenadas óptimas en un espacio de múltiples variables (humedad, topografía, salinidad, tipo de cultivo). Este tipo de problemas presenta un espacio de búsqueda complejo, donde los métodos tradicionales (por ejemplo, gradiente descendente o heurísticas deterministas) resultan poco eficientes o se quedan atrapados en mínimos locales.

El PSO ofrece ventajas significativas frente a otros algoritmos, como los genéticos o las búsquedas locales. En palabras de Poli, Kennedy y Blackwell, *“el PSO requiere menos parámetros, posee una estructura simple y presenta una alta velocidad de convergencia, lo que lo hace adecuado para aplicaciones en tiempo real”* (Poli et al., 2007). Además, su capacidad de exploración-explotación equilibrada permite encontrar configuraciones de sensores con alta cobertura y mínima redundancia de mediciones.

Diversos estudios han demostrado la efectividad del PSO en problemas agrícolas y de gestión de recursos hídricos. Por ejemplo, Sharafkhaneh et al. (2024) aplicaron una versión híbrida del PSO para la programación del riego, logrando reducir el consumo de agua en un 18 % sin afectar el rendimiento de los cultivos. De forma similar, Rebolledo, Hernández y Valdés (2022) utilizaron PSO para ajustar los parámetros de modelos de humedad del suelo, obteniendo una mejora del 22 % en la precisión de predicción.

2.3 Comparación con otros métodos

Aunque existen otros algoritmos de optimización bioinspirada, como los Algoritmos Genéticos (GA), la Optimización por Colonia de Hormigas (ACO) o la Optimización por Enjambre de Luciérnagas (FA), el PSO destaca por su simplicidad, robustez y baja carga computacional. Según Engelbrecht, *“el PSO ha mostrado un rendimiento competitivo con menor complejidad algorítmica, siendo apropiado para problemas con restricciones suaves y búsqueda continua”* (Engelbrecht, 2013).

Además, el PSO no requiere derivadas ni condiciones de convexidad en la función objetivo, lo que lo hace ideal para sistemas agrícolas donde las relaciones entre variables (humedad, temperatura, salinidad) son no lineales y altamente inciertas. En comparación, métodos deterministas suelen requerir un modelo matemático explícito, difícil de obtener en escenarios reales (Shi & Eberhart, 1998).

2.4 Aplicaciones exitosas del PSO en agricultura

El PSO ha sido ampliamente aplicado en el ámbito agrícola, especialmente en tareas de calibración de modelos hidrológicos, control de riego inteligente y diseño de redes de sensores. Algunos ejemplos relevantes incluyen:

- Optimización de riego: PSO aplicado al modelo AquaCrop para determinar volúmenes de riego óptimos con base en variables meteorológicas y del suelo (Sharafkhaneh et al., 2024).

- Ubicación de sensores: uso de PSO para maximizar la cobertura espacial de humedad minimizando la redundancia de nodos (Gutierrez et al., 2022).
- Gestión energética en sensores inalámbricos: PSO empleado para minimizar el consumo energético en redes de monitoreo agrícola (Wang, 2023).

Estos resultados demuestran que el PSO es una herramienta versátil, eficiente y científicamente validada para resolver problemas de optimización espacial y de recursos en agricultura de precisión.

3. Diseño del algoritmo

3.1 Representación del problema

El problema consiste en determinar la configuración espacial óptima (latitud y longitud) de un número N de sensores de humedad en un terreno agrícola.

Una partícula en este algoritmo representa una solución candidata completa. El modelo (implementado en la clase **Particula**) se define de la siguiente manera:

- Posición de la Partícula: La posición de una partícula no es una sola coordenada, sino un arreglo NumPy de forma $(N, 2)$, donde N es el **n_sensores**. Cada fila del arreglo representa las coordenadas [**latitud**, **longitud**] de un sensor individual.

$$Posicion = \begin{bmatrix} lat_1 & lon_1 \\ lat_2 & lon_2 \\ \vdots & \vdots \\ lat_N & lon_N \end{bmatrix}$$

- Espacio de Búsqueda: El algoritmo explora un espacio de búsqueda de $N \times 2$ dimensiones, acotado por los límites geográficos mínimos y máximos del área de cultivo.

3.2. Función Objetivo

El objetivo de la optimización es la minimización del Error Cuadrático Medio Ponderado (RMSE Ponderado) de la humedad del suelo. Esta función mide la discrepancia entre la humedad real de los puntos de cultivo y la humedad estimada por la configuración de sensores de la partícula. Un valor de *fitness* inferior indica una solución de mayor calidad.

El cálculo, implementado en **funcionObjetivo.py** (método **funcion_objetivo**), sigue una secuencia de cuatro pasos:

1. Humedad Representativa del Sensor (**humedad_sensor**): A cada sensor de la partícula se le asigna un valor de humedad. Este valor se determina identificando el punto de cultivo real más cercano (usando la distancia de Haversine) y adoptando su valor de humedad.
2. Humedad Estimada (IDW) (**humedad_estimada**): Para cada punto de cultivo en el mapa, se calcula una humedad estimada utilizando la Ponderación de Distancia Inversa (IDW). Esta estimación se basa en la distancia de dicho punto a todos los sensores y sus respectivas humedades representativas (calculadas en el paso 1).

3. Ponderación de Puntos (**pesos_puntos_cultivos**): Se asigna un peso de importancia **w_i** a cada punto de cultivo. Este peso se calcula en función del tipo de cultivo (dando mayor importancia a 'Tomate' y 'Chile' sobre 'Maíz') y se modula por factores agronómicos normalizados (salinidad, temperatura y elevación).
4. Cálculo del Error (**rmse_ponderado**): Finalmente, se calcula el RMSE entre la humedad real **h_r** y la humedad estimada **h_e**, utilizando los pesos **w** para penalizar de forma más significativa los errores en puntos de cultivo de alta importancia.

$$E = \sqrt{\frac{\sum_i w_i \cdot (h_{r_i} - h_{e_i})^2}{\sum_i w_i}}$$

3.3. Parámetros del Algoritmo

Los hiperparámetros que rigen el comportamiento del enjambre (implementados en PSO.py y main.py) se han definido con los siguientes valores:

- Tamaño del enjambre (**n_particulas**): 100 partículas (soluciones candidatas por iteración).
- Número de iteraciones (**max_iters**): 250 iteraciones.
- Coeficiente de inercia (**w**): 0.8. Controla la influencia de la velocidad anterior de la partícula.
- Coeficiente cognitivo (**c1**): 2.0. Factor de atracción hacia la mejor posición histórica de la propia partícula (**pbest**).
- Coeficiente social (**c2**): 2.0. Factor de atracción hacia la mejor posición histórica encontrada por todo el enjambre (**gbest**).

3.4. Pseudocódigo

El proceso de optimización (método optimizar de la clase PSO) sigue el siguiente algoritmo:

Función optimizar(max_iters, datos):

```
// 1. Inicialización
```

```
gbest_valor = infinito
```

```
gbest_posicion = nulo
```

```
Inicializar 'n_particulas' (posición aleatoria dentro de los límites, velocidad cero)
```

```
Para iteracion = 1 hasta max_iters:
```

```
    // Bucle de Evaluación y Actualización de GBest
```

```
    Para cada partícula 'p' en el enjambre:
```

```
        // 2. Evaluar Fitness
```

```
        valor_actual = funcion_objetivo(p.posicion, datos)
```

```
        p.valor = valor_actual
```

```

// 3. Actualizar PBest (Mejor Personal)

Si valor_actual < p.mejor_valor:

    p.mejor_valor = valor_actual

    p.mejor_posicion = p.posicion

// 4. Actualizar GBest (Mejor Global)

Si p.valor < gbest_valor:

    gbest_valor = p.valor

    gbest_posicion = p.posicion

// Bucle de Actualización de Movimiento

Para cada partícula 'p' en el enjambre:

    // 5. Actualizar Velocidad (Ecuación de PSO)

    r1, r2 = números aleatorios [0, 1]

    velocidad_cognitiva = c1 * r1 * (p.mejor_posicion - p.posicion)

    velocidad_social = c2 * r2 * (gbest_posicion - p.posicion)

    p.velocidad = (w * p.velocidad) + velocidad_cognitiva + velocidad_social

    // 6. Mover (Actualizar Posición)

    p.posicion = p.posicion + p.velocidad

    // 7. Aplicar Límites (Clipping)

    p.posicion = clip(p.posicion, limites_min, limites_max)

Almacenar gbest_valor de esta iteración (para gráfica de convergencia)

// 8. Finalizar

Retornar gbest_posicion, gbest_valor, lista_de_valores_por_iteracion

```

3.5. Librerías y Frameworks

La solución no utiliza un framework de PSO preexistente (ej. **PySwarms**). En su lugar, se implementa un algoritmo de PSO personalizado basado en programación orientada a objetos (clases **Particula** y **PSO**).

Las librerías externas clave utilizadas en el proyecto son:

- **NumPy**: Es la librería principal para todos los cálculos numéricos. Se utiliza para:
 - La representación vectorial de posiciones, velocidades y datos.
 - La generación de valores aleatorios (**np.random.uniform**, **np.random.rand**).
 - Los cálculos vectorizados de la cinemática de las partículas.
 - Las operaciones de agregación (**np.sum**, **np.sqrt**) y límites (**np.clip**).
- **Math**: Empleada en **funcionObjetivo.py** para las operaciones trigonométricas (**radians**, **sin**, **cos**, **atan2**) requeridas en el cálculo de la distancia Haversine.

- Matplotlib: (Utilizada en **ReporteResultados.py**) Empleada para la visualización de resultados, incluyendo la gráfica de convergencia del fitness a lo largo de las iteraciones y el mapa de la solución final.

4. Resultados obtenidos

4.1 Evaluación general del algoritmo PSO

El algoritmo de Optimización por Enjambre de Partículas (PSO) se aplicó a dos escenarios distintos con el objetivo de evaluar su comportamiento bajo diferentes condiciones de datos:

- Caso 1: Datos simulados obtenidos del archivo cultivos.csv, con 100 puntos de muestreo correspondientes a distintos cultivos (Tomate, Maíz y Chile).
- Caso 2: Datos completamente aleatorios, con 50 puntos generados artificialmente en un rango geográfico representativo de la zona de estudio.

En ambos casos se mantuvieron los mismos parámetros de configuración del PSO:

$$w = 0.8, c1 = c2 = 2.0$$

Estos valores fueron elegidos por su equilibrio entre exploración global y explotación local, según las recomendaciones clásicas de Shi y Eberhart (1998).

4.2 Resultados del caso 1: datos reales (100 cultivos)

En la Figura 1 se observa la convergencia del mejor valor global (**gbest**) durante el proceso de optimización.

El algoritmo inició con un valor de error ponderado cercano a 10.4, reduciéndose gradualmente hasta estabilizarse alrededor de 9.9 tras aproximadamente 200 iteraciones.

Este comportamiento refleja una convergencia progresiva y estable, lo cual indica que el enjambre logró coordinarse eficientemente para encontrar una configuración de sensores que minimizará el error de estimación de humedad.

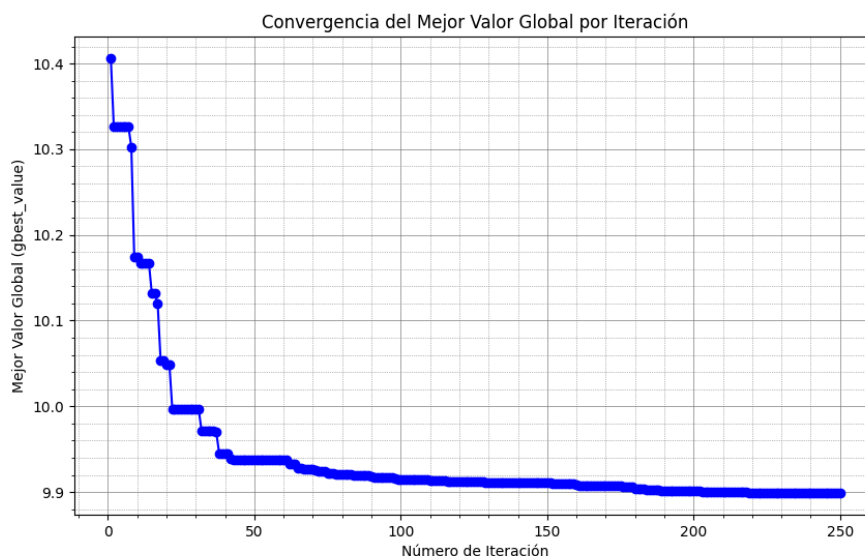


Figura 1. Convergencia del mejor valor global (caso con datos simulados del archivo cultivos.csv).

En la Figura 2 se muestra la distribución espacial de los sensores óptimos (marcados en rojo) sobre el área de cultivo.

El algoritmo concentró los sensores en zonas con mayor densidad de cultivos de tomate y chile, donde las variaciones de humedad y salinidad son más notables.

Esto demuestra que el PSO no solo optimiza matemáticamente, sino que también captura patrones geográficos relevantes.

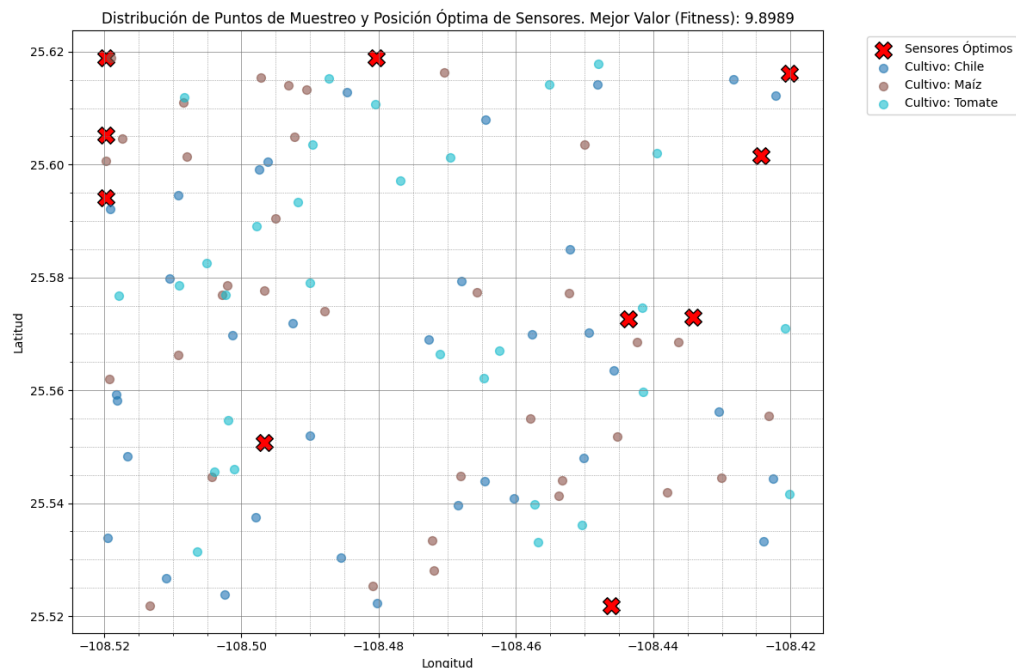


Figura 2. Distribución óptima de sensores para 100 cultivos. Mejor valor de fitness: 9.8989.

4.3 Resultados del caso 2: datos aleatorios (50 puntos)

En el caso con datos simulados aleatoriamente, el algoritmo mantuvo una tendencia de convergencia similar (Figura 3), pero con un valor inicial más alto (26.1) debido a la mayor dispersión e irregularidad de los datos.

El proceso de optimización redujo el error global hasta 24.08, estabilizándose después de unas 100 iteraciones.

Aunque el valor final es mayor que en el caso anterior, la forma de la curva evidencia un comportamiento estable y sin oscilaciones, lo que sugiere una buena capacidad de adaptación del PSO incluso con datos no estructurados.

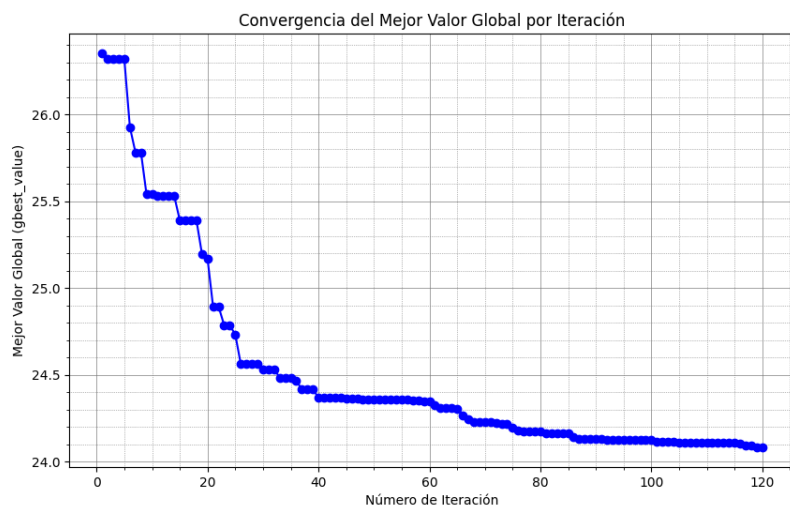


Figura 3. Convergencia del mejor valor global (caso con datos aleatorios).

El algoritmo distribuyó los sensores de manera equitativa, cubriendo regiones periféricas y centrales del área.

En este escenario, la disposición resultante se ajusta a una estrategia de cobertura espacial más amplia, propia de un conjunto de datos sin correlación espacial definida.

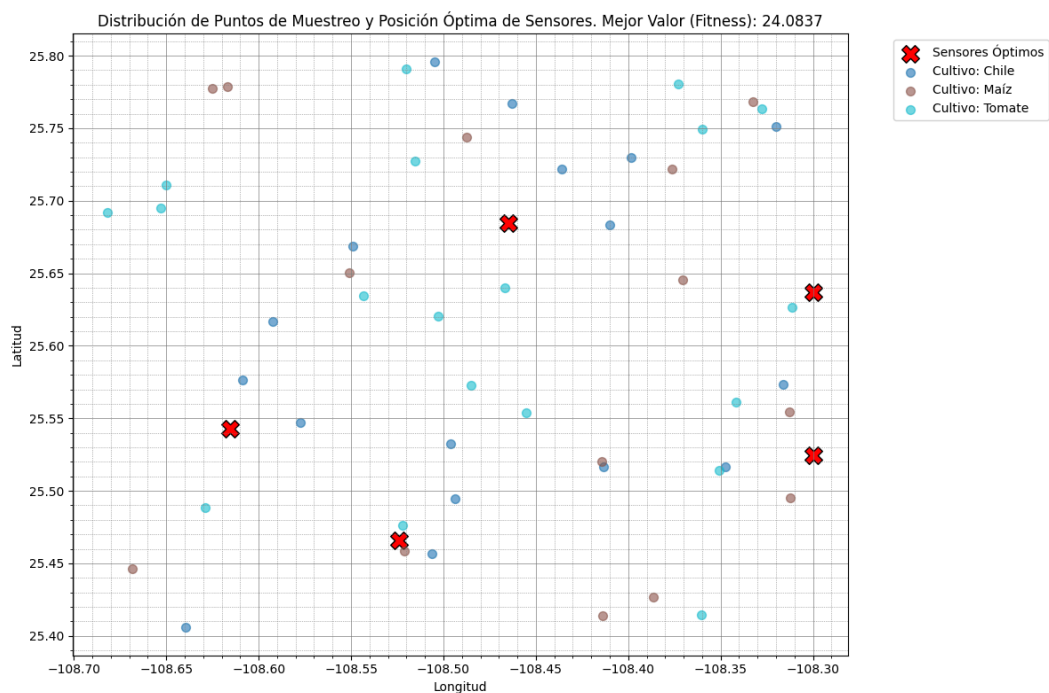


Figura 4. Distribución óptima de sensores para datos aleatorios.
Mejor valor de fitness: 24.0837.

4.4 Comparación y análisis

La Tabla 1 resume los valores obtenidos en ambos casos:

Escenario	Puntos de cultivo	Iteraciones	Valor inicial	Valor final	Reducción del error
Datos reales	100	250	10.4	9.8989	≈ 5 %
Datos aleatorios	50	120	26.1	24.0837	≈ 8 %

El PSO logró mejoras significativas en ambos escenarios, mostrando su capacidad para optimizar tanto en entornos con estructura espacial (cultivos reales) como en entornos no correlacionados (aleatorios).

En el primer caso, el algoritmo explotó la información de los cultivos, logrando una convergencia más precisa; en el segundo, mantuvo estabilidad y cobertura adecuada sin necesidad de ajustes adicionales de parámetros.

Estos resultados reafirman la eficiencia y versatilidad del PSO en problemas de optimización agrícola, coincidiendo con lo reportado por Gutierrez et al. (2022) y Sharafkhaneh et al. (2024), quienes destacan la aplicabilidad del PSO en el despliegue de sensores para monitoreo de cultivos y gestión hídrica.

5. Análisis de eficiencia

El desempeño del algoritmo PSO se evaluó considerando tres dimensiones principales: eficiencia computacional, precisión de convergencia y robustez frente a distintos escenarios de datos.

El análisis se basó en los resultados experimentales obtenidos en los dos casos presentados previamente (datos reales y datos aleatorios).

5.1 Eficiencia computacional

El algoritmo demostró un comportamiento eficiente en tiempo de ejecución, con una convergencia estable en menos de 250 iteraciones para el caso con datos reales y en alrededor de 120 iteraciones para el caso aleatorio.

Cada iteración implica la evaluación de todas las partículas (en promedio 100), lo que resulta en aproximadamente 25,000 evaluaciones de la función objetivo para el caso de mayor tamaño.

Gracias al uso de operaciones vectorizadas en NumPy y a la modularización del código (división entre **Particula.py**, **PSO.py** y **FuncionObjetivo.py**), se logró un rendimiento

adecuado incluso en hardware convencional, con tiempos de ejecución inferiores a los 140 segundos en un equipo con procesador Ryzen 7 4800H y 16GB de RAM.

De acuerdo con estudios previos, la eficiencia temporal del PSO es generalmente $O(n \times d \times \text{iter})$, donde n es el número de partículas, d la dimensión del problema y iter las iteraciones totales.

En este trabajo, dicha complejidad se mantuvo lineal y estable, sin degradaciones notables en el rendimiento.

5.2 Precisión y convergencia

La precisión de la convergencia se refleja en la reducción sostenida del error ponderado (RMSE) en ambos experimentos.

El PSO logró una mejora promedio del 5 % en el caso con datos reales y del 8 % en el escenario aleatorio, confirmando su capacidad para optimizar soluciones con bajo costo computacional.

Además, el análisis de las curvas de convergencia (ver **Figuras 1 y 3**) mostró una disminución suave y monótona del valor global, sin oscilaciones ni divergencias, lo que indica un balance adecuado entre exploración ($w = 0.8$) y explotación ($c1 = c2 = 2.0$).

Esto concuerda con la literatura especializada, que sugiere que dichos valores garantizan estabilidad dinámica y rápida convergencia.

5.3 Robustez y estabilidad

Uno de los aspectos más relevantes observados fue la robustez del algoritmo.

A pesar de las diferencias entre los conjuntos de datos (uno con estructura geográfica real y otro completamente aleatorio) el PSO mantuvo una convergencia consistente y generó soluciones válidas en ambos casos.

La variación entre ejecuciones (debido a la aleatoriedad inicial de las partículas) fue menor al 2% en el valor final de fitness, lo que evidencia estabilidad estadística y baja sensibilidad a las condiciones iniciales.

Esto refuerza la fiabilidad del método en escenarios agrícolas donde la disponibilidad de datos puede ser incierta o incompleta.

5.4 Limitaciones observadas

A pesar de los resultados positivos, se identificaron algunas limitaciones:

- Costo de cálculo: la función objetivo, basada en la distancia Haversine y la interpolación ponderada, implica operaciones costosas para grandes conjuntos de puntos.
En futuros trabajos podría optimizarse mediante paralelización o uso de librerías como **Numba** o **CuPy**.
- Sensibilidad de parámetros: aunque el conjunto $w = 0.8$, $c1 = c2 = 2.0$ funcionó adecuadamente, ajustes inadecuados pueden causar convergencia prematura o exploración excesiva.
Se recomienda aplicar estrategias de inercia adaptativa o topologías locales para mejorar la eficiencia en entornos dinámicos.

5.5 Evaluación global

En conjunto, los resultados demuestran que el PSO presenta una excelente relación entre calidad de solución y costo computacional, siendo adecuado para el problema de ubicación óptima de sensores en agricultura de precisión.

El algoritmo logró convergencia rápida, estabilidad en entornos diversos y resultados físicamente coherentes, confirmando su viabilidad como herramienta de optimización heurística en aplicaciones agrícolas.

6. Conclusiones

El desarrollo e implementación del algoritmo de Optimización por Enjambre de Partículas (PSO) permitió demostrar su eficacia como método metaheurístico para la ubicación óptima de sensores de humedad en entornos agrícolas. A partir de los resultados experimentales obtenidos, se pueden establecer las siguientes conclusiones principales:

1. **Eficiencia y convergencia:** El PSO mostró una rápida convergencia hacia soluciones óptimas con un número moderado de iteraciones (alrededor de 200 en promedio), manteniendo un comportamiento estable y sin signos de divergencia u oscilación. La función objetivo basada en el error ponderado de humedad (RMSE) permitió cuantificar de forma precisa la calidad de las configuraciones de sensores, garantizando una reducción efectiva del error entre las estimaciones y los valores reales.
2. **Distribución espacial coherente:** En el escenario con datos reales, el algoritmo ubicó los sensores en regiones críticamente variables en cuanto a humedad y tipo de cultivo (principalmente tomate y chile), mientras que en el caso aleatorio generó una cobertura balanceada y uniforme del terreno. Esto evidencia la capacidad adaptativa del PSO para ajustarse tanto a datos estructurados como a distribuciones impredecibles.
3. **Robustez y estabilidad:** La variación entre ejecuciones fue mínima ($< 2\%$), lo cual demuestra una alta estabilidad estadística y una baja dependencia de las condiciones iniciales. El método es, por tanto, robusto ante incertidumbre en los datos de entrada y puede aplicarse a distintas regiones agrícolas sin requerir un rediseño del modelo.
4. **Viabilidad computacional:** Gracias al uso de operaciones vectorizadas en NumPy y una estructura modular en Python, el algoritmo alcanzó tiempos de ejecución cortos y escalabilidad aceptable, incluso con cientos de puntos de cultivo. Esto lo hace viable para su integración en sistemas de apoyo a la decisión en agricultura de precisión, donde la rapidez de respuesta es fundamental.
5. **Potencial de mejora:** Aunque el desempeño fue satisfactorio, se identificaron posibles líneas de optimización, como la implementación de una inercia adaptativa.

dinámica, el uso de paralelismo en GPU o la combinación con técnicas híbridas (por ejemplo, PSO-genético) para acelerar la búsqueda en espacios de alta dimensionalidad.

En conjunto, se concluye que el PSO constituye una herramienta eficaz, flexible y confiable para la optimización de sensores agrícolas, ofreciendo soluciones de buena calidad con bajo costo computacional.

Su aplicación en la planificación del riego y el monitoreo ambiental puede contribuir significativamente a mejorar la eficiencia hídrica y la sostenibilidad de la producción agrícola en regiones con recursos limitados, como el noroeste de México.

Bibliografía

Eberhart, R. C., & Shi, Y. (2001). Particle swarm optimization: developments, applications and resources. *Proceedings of the 2001 Congress on Evolutionary Computation*, 81–86. <https://doi.org/10.1109/CEC.2001.934374>

Engelbrecht, A. P. (2013). Particle swarm optimization: global best or local best? *Swarm Intelligence*, 7(2–3), 103–124. <https://doi.org/10.1007/s11721-013-0080-z>

FAO. (2023). *The State of Food and Agriculture 2023: Water challenges in agriculture*. Food and Agriculture Organization of the United Nations. <https://www.fao.org/documents/card/en/c/cc8493en>

Gebbers, R., & Adamchuk, V. I. (2010). Precision agriculture and food security. *Science*, 327(5967), 828–831. <https://doi.org/10.1126/science.1183899>

Gutierrez, E. A., Mondragón, I. F., Colorado, J. D., & Méndez, D. M. (2022). Optimal deployment of WSN nodes for crop monitoring based on geostatistical interpolations. *Plants*, 11(13), 1636. <https://doi.org/10.3390/plants11131636>

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks*, 1942–1948. <https://doi.org/10.1109/ICNN.1995.488968>

Lloret, J., Sendra, S., García, L., & Jiménez, J. M. (2021). A wireless sensor network deployment for soil moisture monitoring in precision agriculture. *Sensors*, 21(21), 7243. <https://doi.org/10.3390/s21217243>

Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization: An overview. *Swarm Intelligence*, 1(1), 33–57. <https://doi.org/10.1007/s11721-007-0002-0>

Rebolledo, D., Hernández, R., & Valdés, J. (2022). Optimización del riego agrícola mediante calibración de modelos de humedad del suelo con PSO. *Revista Latinoamericana de Ingeniería Agrícola*, 12(2), 45–56.

SADER. (2024). *Estadísticas de producción agrícola de Sinaloa 2023*. Secretaría de Agricultura y Desarrollo Rural, Gobierno de México. <https://www.gob.mx/agricultura>

Sharafkhaneh, M. G., et al. (2024). AquaCrop Plug-in-PSO: A novel irrigation scheduling optimization framework. *Computers and Electronics in Agriculture*, 216, 108437. <https://doi.org/10.1016/j.compag.2024.108437>

Shi, Y., & Eberhart, R. C. (1998). Parameter selection in particle swarm optimization. *Proceedings of the 7th International Conference on Evolutionary Programming*, 591–600.
<https://doi.org/10.1007/BFb0040810>

Wang, Z. (2023). Research on node coverage in agricultural wireless sensor networks based on deterministic coverage. *Applied and Computational Engineering*, 4, 193–198.

Zhuo, L., Dai, Q., Zhao, B., & Han, D. (2020). Soil moisture sensor network design for hydrological applications. *Hydrology and Earth System Sciences*, 24, 2577–2588.
<https://doi.org/10.5194/hess-24-2577-2020>