



**TECNOLÓGICO  
NACIONAL DE MÉXICO**



Temas de IA

Modulo III Algoritmos bio inspirados

05/11/2025

Gomez Gastelum Diego Alejandro  
Meza Manjarrez Marco Josue

# **Proyecto de Inteligencia Artificial: Optimización de Riego con Enjambre de Partículas**

**Competencia:** Resolver problemas en base a técnicas de búsqueda en espacio de estados.

## **1. Descripción del Proyecto**

El objetivo principal es optimizar la ubicación de sensores de humedad en un campo agrícola para maximizar la eficiencia del riego, utilizando una técnica de búsqueda en el espacio de estados: la Optimización por Enjambre de Partículas (PSO).

El script cumple con esto al emplear la biblioteca pswarms para inicializar y ejecutar un optimizador Global Best PSO. La "mejor distribución de sensores" es definida y encontrada a través de la Función Objetivo, que busca un equilibrio entre varios factores ambientales simulados.

Una ubicación eficiente de los sensores es crucial para obtener datos representativos que reflejen la heterogeneidad del terreno, lo cual permite la aplicación de sistemas de riego variable por zonas. Se deben considerar múltiples factores espaciales para la optimización, incluyendo:

- La variabilidad del contenido de humedad en el suelo.
- La calidad y retención de agua del suelo.
- La necesidad de máxima cobertura espacial para evitar la redundancia de datos.

## **2. Justificación de uso del algoritmo PSO**

El algoritmo de Enjambre de Partículas (PSO) es una metaheurística bioinspirada ideal para resolver este problema de optimización debido a las siguientes razones:

**Optimización Multidimensional:** El problema de ubicar  $N=10$  sensores requiere optimizar 20 dimensiones (una coordenada  $x$  y una  $y$  para cada sensor). PSO es altamente efectivo en la exploración de grandes espacios de búsqueda de alta dimensionalidad.

**Problema Multi-Objetivo:** La calidad de una solución (la ubicación de los 10 sensores) se define por múltiples factores (varianza de humedad, calidad del suelo, distancia entre sensores) que a menudo están en conflicto. PSO maneja bien la evaluación de una función de aptitud compleja que combina estos criterios.

**Robustez y Simplicidad:** PSO requiere pocos hiperparámetros de ajuste ( **$c1$ ,  $c2$ ,  $w$** ) en comparación con otros algoritmos genéticos, lo que facilita su implementación y adaptación al problema agrícola.

### 3. Diseño del Algoritmo

El diseño se estructura en tres componentes principales: el entorno simulado, la representación de la solución y la función de aptitud.

Modelo del Entorno (Clase CampoAgricola)

Se simula un campo de unidades, con matrices de datos para:

- Humedad (self.humedad): Valores aleatorios entre 0.2 y 0.9.
- Calidad del Suelo (self.suelo): Índice de 0.5 a 1.0.
- Topografía (self.topografia): Elevación simulada con distribución normal.

Representación de la Solución

- Número de Sensores (): 10.
- Dimensiones (): 20 ().
- Límites: El espacio de búsqueda está acotado por los límites del campo . Cada partícula es un vector de 20 posiciones, representando las coordenadas .

Función de Aptitud (Función Objetivo)

La función evaluar\_configuracion calcula el *Score* que el PSO intenta maximizar (al minimizar el negativo de este score):

Score = (Humedad total + Varianza de Humedad + Suelo Total) – Penalizacion

### 4. Herramientas

NumPy (`import numpy as np`)

Es la librería fundamental para la computación numérica y el manejo eficiente de grandes arreglos de datos.

PySwarms (`import pyswarms as ps`)

Es la librería especializada que implementa el algoritmo de Optimización por Enjambre de Partículas (PSO).

Matplotlib (`import matplotlib.pyplot as plt`)

Es la librería estándar de Python usada para la creación de gráficos y visualizaciones estáticas, animadas e interactivas.

## 5. Resultados Obtenidos

La ejecución del algoritmo PSO arroja las siguientes salidas clave:

**Mejor Costo Encontrado (Score Máximo):** Se obtiene el valor mínimo del negativo del Score. Un costo de, por ejemplo, -17.56 implica un score máximo de \$17.56\$.

**Coordenadas Óptimas:** Se obtiene el vector de 20 posiciones que corresponde a la mejor distribución.

El resultado final se presenta visualmente, donde las coordenadas óptimas se superponen a los mapas del campo. Esto evidencia que el algoritmo ha ubicado los 10 sensores estratégicamente en áreas que:

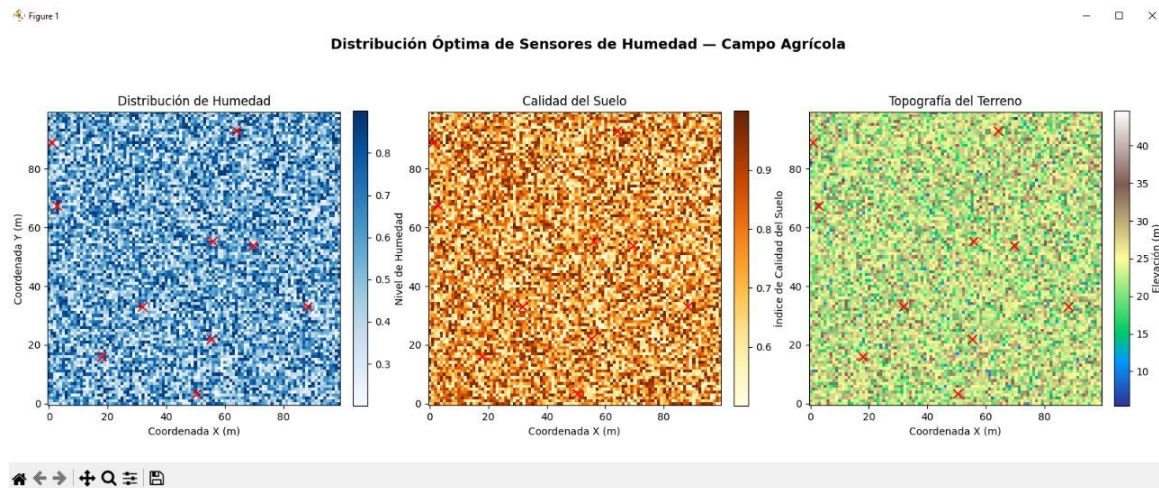
Están lejos unas de otras (buena cobertura).

Corresponden a diferentes niveles de humedad.

Están en zonas de alta calidad del suelo

## 6. Análisis de Eficiencia

La eficiencia del algoritmo se juzga por su capacidad para converger a una solución óptima en un número razonable de iteraciones (150 en la implementación) y por la calidad de la solución (el Score obtenido).



```

===== RESULTADOS =====
Mejor costo encontrado: -14.7535
Coordenadas óptimas de sensores (x, y):
[[64.10595776 93.03272893]
 [31.78777953 33.04111909]
 [69.65614501 53.92958638]
 [ 2.76560609 67.39773319]
 [17.96273721 16.09505276]
 [50.4403399  3.487935  ]
 [55.28067399 21.94554391]
 [88.18672672 33.19974452]
 [ 0.95710978 89.13760988]
 [55.82338497 55.3227488  ]]

```

Figure 1

### Distribución Óptima de Sensores — Campo Guasave, Sinaloa

