

# Tecnológico Nacional de México Campus Culiacán

Proyecto de Optimización de Riego con  
Enjambre de Partículas

Tópicos de Inteligencia Artificial

**Profesor:**

Zuriel Dathan Mora Felix

**Integrantes:**

Aguilar Recio Jesús Octavio

Echeagaray Aceves Astrid Monserrath



**Repositorio GitHub:**

<https://github.com/OctavioAR/TOPICOS-IA.git>

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Objetivos</b>	<b>2</b>
2.1. Objetivo general . . . . .	2
2.2. Objetivos específicos . . . . .	2
<b>3. Justificación</b>	<b>3</b>
<b>4. Alcance</b>	<b>5</b>
<b>5. Desarrollo</b>	<b>6</b>
5.0.1. Clase Configuración . . . . .	6
5.0.2. Clase Partícula . . . . .	6
5.0.3. Clase Enjambre . . . . .	6
5.0.4. Clase Función Objetivo . . . . .	6
5.0.5. Clase Resultados . . . . .	7
5.0.6. Clase Main . . . . .	7
5.0.7. Resultados . . . . .	7
<b>6. Agenda</b>	<b>9</b>
<b>7. Conclusión</b>	<b>9</b>

# Índice de Imágenes

1. Resultado consola parte 1 . . . . .	8
2. Resultado consola parte 2 . . . . .	8
3. Resultado gráfico . . . . .	8
4. Cronograma de actividades . . . . .	9

## 1. Introducción

La agricultura es una actividad fundamental en el mundo la cuál con el paso del tiempo es más demandada por el incremento exponencial de la población. Por eso mismo enfrenta desafíos como la seguridad alimentaria y el riego, ya que al tener que generar más cultivos en parcelas cada vez más grandes es imposible de gestionar por simples humanos, pero con la ayuda de tecnologías se pueden reducir este tipo de problemas. En este proyecto se plantea una problemática en la región agrícola de Guasave, Sinaloa ya que se necesita gestionar el riego de cultivos utilizando sensores. Para la colocación de dichos sensores se utiliza la lógica del algoritmo bio inspirado de Optimización por Enjambre de Partículas (PSO) ya que simula el comportamiento colectivo de sistemas naturales para resolver problemas complejos. En general en este proyecto se centra en desarrollar un sistema que utilice la lógica del algoritmo PSO para obtener la ubicación optima de los sensores, con el fin de gestionar de manera eficiente los recursos hídricos en una de las región agrícola de Guasave.

## 2. Objetivos

### 2.1. Objetivo general

Desarrollar un sistema para optimizar la ubicación de sensores de humedad en campos agrícolas de la región de Guasave, Sinaloa, utilizando el algoritmo bio-inspirado de Optimización por Enjambre de Partículas (PSO), con el fin de mejorar la eficiencia de riego y la gestión de los recursos hídricos.

### 2.2. Objetivos específicos

- Modelar el problema de ubicación como un espacio de búsqueda de 2 dimensiones (latitud, longitud) dentro de los límites geoespaciales de la zona de estudio.
- Diseñar y definir una función de aptitud o objetivo que evalúe la calidad de una ubicación candidata.
- Implementar el algoritmo PSO para explorar el espacio de búsqueda y encontrar las ubicaciones optimas de los sensores.

### 3. Justificación

El municipio de Guasave, Sinaloa “es conocido como el corazón agrícola de México debido a su alta productividad. Cuenta con cerca de 200,000 hectáreas bajo riego, por lo que es el municipio con la mayor superficie de siembra en este sistema a nivel nacional.” [1]. Por eso mismo al ser una de las principales regiones del país para cuestiones de cultivos, es necesario modernizar los sistemas de monitoreo y riego ya que actualmente se cuenta con carencias tecnológicas enfocadas al cultivo en México. Y a causa de estas limitaciones los agricultores se ven afectados económicamente y también los consumidores, en sí a todos nos perjudica de algún modo este problema.

En la actualidad, la agricultura enfrenta grandes desafíos relacionados con la sostenibilidad y el uso eficiente de los recursos naturales, especialmente el agua. El crecimiento de la población mundial y el cambio climático han intensificado la presión sobre los sistemas agrícolas, demandando prácticas más inteligentes que permitan optimizar los procesos de producción sin comprometer el medio ambiente. De hecho, “utilizar eficientemente los recursos limitados de tierra cultivable sigue siendo un desafío significativo en el desarrollo de las economías rurales” [6]. En este contexto, la incorporación de tecnologías de monitoreo y optimización basadas en inteligencia artificial representa una herramienta clave para mejorar la eficiencia del riego y la gestión de los cultivos.

En el estado de Sinaloa, la agricultura de riego tiene una relevancia económica y social fundamental. Tal como menciona Mancera Gonzáles, “en el estado de Sinaloa, ubicado en el noroeste de México, el 94 % del agua disponible se utiliza para la agricultura, el 5 % para el abastecimiento público y el restante para la industria y la generación de energía eléctrica” [4]. Este volumen, estimado en 9005 hm<sup>3</sup> anuales, refleja la gran demanda de agua que caracteriza a la región y la importancia de implementar estrategias de optimización hídrica mediante tecnologías inteligentes de monitoreo y control.

Una de las principales estrategias para lograr un riego eficiente consiste en la correcta ubicación de sensores de humedad en el suelo, los cuales permiten obtener información precisa sobre las condiciones hídricas del terreno. Sin embargo, determinar la posición óptima de estos sensores constituye un problema complejo, ya que depende de múltiples variables como la topografía del terreno, la distribución de los cultivos y la variabilidad del contenido de humedad del suelo.

Para abordar este tipo de problemas de optimización, se han desarrollado algoritmos bioinspirados que imitan el comportamiento colectivo de sistemas naturales. Estos métodos se enmarcan en la Inteligencia de Enjambre (SI, por sus siglas en inglés), definida como “una rama de la Inteligencia Artificial basada en el comportamiento colectivo inteligente de enjambres sociales en la naturaleza” [3]. Entre ellos destaca la Optimización por Enjambre de Partículas (PSO, por sus siglas en inglés), un algoritmo propuesto originalmente por Kennedy y Eberhart [5], “inspirado en el comportamiento social y biológico de las parvas de aves que buscan alimento” [2]. Este algoritmo es capaz de explorar grandes espacios de búsqueda y encontrar soluciones eficientes mediante la cooperación

entre partículas.

En este proyecto se propone aplicar el algoritmo PSO a la industria agrícola, específicamente en la región de Guasave, Sinaloa, con el propósito de optimizar la colocación de sensores de humedad en campos agrícolas. El uso de PSO es pertinente, ya que se ha demostrado su capacidad para resolver problemas de optimización en la agricultura, como la “planificación de cultivos” [6]. A través de este enfoque, se busca maximizar la cobertura de monitoreo y la diversidad de cultivos atendidos, reduciendo redundancias y mejorando la eficiencia en el uso del agua.

## 4. Alcance

El alcance del presente proyecto se centra en desarrollar un algoritmo bio-inspirado para resolver problemas de optimización, dicho algoritmo es el PSO con el cual podremos ubicar la localización óptima para la colocación de los sensores en los cultivos de la región de Guasave. Por ende, el alcance incluye el desarrollo del algoritmo PSO en el lenguaje de programación Python, además de implementar una función de aptitud (FO) capaz de medir cuales son las mejores ubicaciones de los sensores (partículas) utilizando los datos que se brindaron sobre la distribución de los cultivos, los distintos tipos de porcentaje de humedad, también las características del suelo como la salinidad, etc. Todo se integrará mediante el uso de bibliotecas especializadas para poder manipular estos tipos de datos y poder hacer operaciones matemáticas con ellos. Además, geográficamente el proyecto se limita a la región de Guasave, Sinaloa, abarcando ciertas coordenadas específicas de latitud y longitud. Por motivos del curso y tiempo, utilizamos la lógica básica del algoritmo PSO para resolver el problema, el proyecto se limita a encontrar las coordenadas óptimas para colocar los sensores, la optimización de parámetros, las pruebas de validación y la documentación.

## 5. Desarrollo

Para el desarrollo del sistema se decidió utilizar Python debido a su amplio catálogo de librerías matemáticas y de IA. Se decidió seguir una estructura orientada a objetos para así poder modular el código. El proyecto está conformado por las siguientes clases:

### 5.0.1. Clase Configuración

La clase “configuracion” se encarga de reunir todos los parámetros de configuración del algoritmo PSO. Se establece los valores que controlan el comportamiento del enjambre, como el numero de sensores, partículas, iteraciones, dimensiones, límites geográficos y los parámetros de inercia, coeficiente cognitivo y social. Dicha clase tiene una función llamada “obtener limites()” en la que genera los límites de posición y velocidad para cada dimensión del espacio de búsqueda. Dicha función asegura que todas las soluciones (ubicaciones) generadas se mantengan dentro del área de búsqueda.

### 5.0.2. Clase Partícula

En la clase “particula” crea un objeto partícula con sus atributos y métodos, esta representa una partícula individual dentro del enjambre PSO, cada partícula tiene cinco atributos principales: posición, velocidad, valor, mejor\_posicion y mejor\_valor. Dentro de la clase se encuentra la función “actualizar\_apetitud()” la cual se encarga de comparar la aptitud actual con la mejor, actualizándola solo si se encuentra una mejor solución. Dicha clase preserva el conocimiento histórico acumulado durante la búsqueda.

### 5.0.3. Clase Enjambre

En la clase “enjambre” se encuentra la lógica del algoritmo PSO, teniendo dos funciones principales. La primera “inicializar\_enjambre()” es en donde se crean todas las partículas con posiciones y velocidades aleatorias dentro de los límites, también se evalúa con la FO su aptitud en dicha posición inicial. La segunda función “iteraciones\_PSO” realiza la lógica del algoritmo PSO el cual actualiza las velocidades de cada partícula mediante la combinación de los tres componentes de inercia (mantiene la dirección actual), componente cognitivo (atrae hacia la mejor posición) y componente social (atrae hacia la mejor posición global), después actualizamos la posición de la partícula y volvemos a calcular su aptitud en esa nueva posición.

### 5.0.4. Clase Función Objetivo

En la clase “función\_objetivo” implementa la lógica para medir de alguna forma si la posición en la que se encuentra el sensor es buena o mala. La FO utiliza métricas como la cobertura geográfica junto con variables agrícolas definidas

en el archivo csv. La función procesa las posiciones de todos los sensores, calculando distancias euclidianas hacia cada cultivo mediante operaciones de vectores utilizando las librerías de NumPy para así determinar su cobertura dentro del radio especificado. La segunda métrica es en base a los cultivos cubiertos, evalúa tres dimensiones agrónomas: diversidad de tipos de cultivo, variabilidad de la humedad y rango de elevación. La aptitud final se calcula por medio de una suma ponderada que integra componentes de cobertura (minimizando distancias promedio), penalización por cultivos no cubiertos y calidad agrícola (maximizando diversidad y condiciones óptimas), utilizando pesos que reflejen la importancia de cada variable.

#### **5.0.5. Clase Resultados**

En esta clase “ResultadosGraficos” se encarga de representar gráficamente los resultados obtenidos. Teniendo una función “cobertura\_sensores()” la cual utiliza la librería Matplotlib para crear un diagrama que muestre la distribución de los cultivos en el espacio geográfico y las posiciones optimas de los sensores y los círculos de cobertura que ilustran el área de abarca cada sensor.

#### **5.0.6. Clase Main**

Esta clase “main” ejecuta el flujo de optimización, comenzando por cargar la configuración y validando la existencia del archivo de datos con un bloque de código try-except, después se instancia el enjambre PSO con todos los parámetros que necesita y ejecuta la función optimizar que da inicio al algoritmo PSO, después procesa los resultados obtenidos y generamos la visualización.

#### **5.0.7. Resultados**

El proceso del algoritmo PSO demostró una evolución buena en lo largo de 150 iteraciones, 4 sensores y 50 partículas, donde cada partícula representa una configuración completa de los 4 sensores. Iniciando con una aptitud de 233.83 y terminando con una aptitud de 62.92 demostrando una mejora global del sistema. También se muestra como a lo largo de las iteraciones se cubrieron 83, 54, 68 cultivos y al finalizar se lograron cubrir 98 cultivos de 100. Además, que la diversidad de cultivos se mantuvo constante (maíz, chile y tomate) logrando así cubrir la diversidad de estos. La distribución de los 4 sensores óptimos forma una configuración cuadrangular como se muestra en la figura 3 maximizando así el área que cubrir. Con el radio de cobertura de 0.03 unidades demostró ser efectivo para cubrir un 97 % del total de los cultivos. Los resultados finales muestran la efectividad del enfoque de minimización de costos, donde valores más bajos corresponden directamente a configuraciones con mayor cobertura geográfica y mejor calidad de monitorización agrícola.



```

==== CONFIGURACIÓN DEL PROBLEMA ====
Numero de sensores: 4
Dimensiones del problema: 8
Numero de particulas: 50
Límites geograficos: lat[25.52, 25.62], Lon[-108.52, -108.42]
Radio de cobertura: 0.03
=====
Datos cargados correctamente

INICIANDO OPTIMIZACIÓN DE 4 SENSORES
Iniciando enjambre con 50 particulas...
Enjambre inicializado. Mejor aptitud inicial: 233.8347

Iniciando optimización PSO por 150 iteraciones...
Cubiertos: 83, Aptitud: 213.36
  Nuevo mejor global en iteración 1: 213.3641
  Nuevo mejor global en iteración 2: 204.0818
Cubiertos: 54, Aptitud: 502.93
  Nuevo mejor global en iteración 3: 183.8545
  Nuevo mejor global en iteración 3: 182.7261
Cubiertos: 84, Aptitud: 203.11
Cubiertos: 68, Aptitud: 363.57
Cubiertos: 84, Aptitud: 202.75
  Nuevo mejor global en iteración 5: 162.8626
Cubiertos: 79, Aptitud: 252.52
Cubiertos: 75, Aptitud: 291.44
Cubiertos: 71, Aptitud: 334.78
  Nuevo mejor global en iteración 10: 153.1913
  Nuevo mejor global en iteración 10: 152.6946
Iteración 10: Mejor aptitud = 152.6946, Mejoras = 28

```

Figura 1: Resultado consola parte 1

```

Iteración 140: Mejor aptitud = 62.9289, Mejoras = 2
Cubiertos: 96, Aptitud: 82.93
Cubiertos: 95, Aptitud: 92.71
Cubiertos: 96, Aptitud: 82.45
Cubiertos: 97, Aptitud: 73.05
Cubiertos: 96, Aptitud: 82.49
Iteración 150: Mejor aptitud = 62.9289, Mejoras = 2

Optimización completada.

==== RESULTADOS FINALES ====
Sensor 1: Lat=25.540010, Lon=-108.499266
Sensor 2: Lat=25.594415, Lon=-108.501998
Sensor 3: Lat=25.553095, Lon=-108.444802
Sensor 4: Lat=25.596264, Lon=-108.460302
Aptitud final: 62.9289
Cultivos cubiertos: 98 de 100

```

Figura 2: Resultado consola parte 2

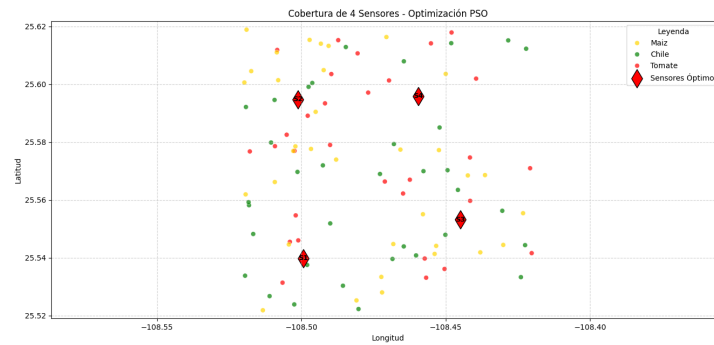


Figura 3: Resultado gráfico

## 6. Agenda

El proyecto constó de 10 días para su elaboración, como se puede apreciar en la figura 4 la cual representa cómo es que se dividieron los días para poder realizar dicho proyecto, siendo los cuadros de color verde los días esperados para su elaboración y rojos los días en que se realizó.



Figura 4: Cronograma de actividades

## 7. Conclusión

El desarrollo e implementación del sistema utilizando algoritmo de optimización por enjambre de partículas para la optimización de ubicación de sensores de humedad en los campos de Guasave, Sinaloa, demostró ser altamente efectivo, cubriendo un 97 % total de los cultivos con una configuración de cuatro sensores estratégicamente distribuidos por el campo y dándoles un radio favorable de 0.03 para cubrir de manera amplia. La implementación exitosa valida la eficacia del enfoque de minimización de costos para resolver problemas en contextos agrícolas. El sistema desarrollado sienta las bases para incorporar técnicas más adelante de inteligencia artificial enfocadas en la agricultura o incluso en inteligencia de recursos hídricos en regiones agrícolas críticas para la seguridad alimentaria.

## Referencias

- [1] Marco Díaz. «Guasave late por una AgriculturaConCiencia». En: *Divulgación-CIMMYT*. 2019, págs. -. URL: <https://goo.su/Q9eDKv>.
- [2] Diogo Freitas, Luiz Guerreiro Lopes y Fernando Morgado-Dias. «Particle Swarm Optimisation: A Historical Review Up to the Current Developments». En: *Entropy* (2020), pág. 362. DOI: 10.3390/e22030362.
- [3] Ahmed G. Gad. «Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review». En: *Archives of Computational Methods in Engineering* (2022), págs. 2531-2561. DOI: 10.1007/s11831-021-09694-4.
- [4] Omar Mancera González. «Las codependencias de la agroindustria en Sinaloa, México». En: *Perfiles Latinoamericanos* (2023). DOI: 10.18504/p13161-006-2023.
- [5] J. Kennedy y R. Eberhart. «Particle Swarm Optimization». En: *Proceedings of IEEE International Conference on Neural Networks* (1995), págs. 1942-1948.
- [6] Zijie Xu y Yuhui Sun. «Particle Swarm Optimization for Agricultural Problems». En: *Highlights in Business, Economics and Management* (2025).