



INSTITUTO TECNOLÓGICO DE CULIACÁN

Ingeniería en Sistemas Computacionales

Tópicos de Inteligencia Artificial

Hora: 12:00 – 01:00 PM

Optimización de riego con Enjambre de Partículas

Equipo:

Peña López Miguel Ángel

Robles Rios Jacquelin

Docente:

Mora Félix Zuriel Dathan

Culiacán, Sinaloa

28/10/2025

1	índice
2	<i>Descripción del problema</i> 3
3	<i>Justificación de uso de algoritmo</i> 4
4	<i>Diseño de algoritmo</i> 5
5	<i>Resultados</i> 7
6	<i>Análisis de eficiencia</i> 15
7	<i>Conclusión</i> 16
8	<i>Referencias</i> 17

2 Descripción del problema

El desafío que abordar es optimizar la ubicación de sensores de humedad en campos agrícolas de Guasave, Sinaloa, para maximizar la eficiencia del riego y el aprovechamiento de los recursos hídricos. La optimización busca determinar la mejor distribución de sensores, considerando variables críticas que afectan la retención y disponibilidad de agua en el suelo.

La topografía del terreno de la región de Guasave es predominantemente plana, pero existen micro variaciones en elevación (entre 10 y 50 metros sobre el nivel del mar) que impactan la retención de humedad. Y mayormente su distribución de cultivos se considera de las diferentes necesidades hídricas de los principales cultivos de la zona: Maíz grano (*21.7% de la producción estatal*), Tomate rojo (9.9%), y Chile verde (*producción significativa en invierno*).

En esta zona geográfica se debe considerar además la variabilidad del suelo, donde se deben integrar factores de degradación como la disminución de materia orgánica (de 1.5% a menos de 1% en 20 años) y el aumento de salinidad en zonas de mal drenaje, condiciones que afectan directamente la retención de agua y la eficiencia del riego.

El problema se modelará como un espacio de búsqueda multidimensional donde cada "solución" es un conjunto de coordenadas (Latitud, Longitud) que representa la ubicación de un sensor. El objetivo es encontrar la configuración de sensores que ofrezca la máxima cobertura o representatividad de la variabilidad hídrica del campo.

3 Justificación de uso de algoritmo

El algoritmo de optimización por enjambre de partículas (PSO) es una metaheurística de búsqueda poblacional inspirada en el comportamiento social de bandadas de aves o bancos de peces. Este fue elegido por sus ventajas que se alinean con los requisitos y necesidades del proyecto.

El problema de ubicación de sensores en un campo agrícola se define en un espacio de búsqueda continuo de coordenadas geoespaciales (latitud, longitud), y según el artículo Hydro Easy este algoritmo es: *"Muy utilizado en ciencia e ingeniería para encontrar los mínimos y máximos globales de un sistema, y su funcionamiento se basa en el comportamiento colectivo de los animales en los que el movimiento de cada individuo es el resultado de combinar las decisiones individuales de cada uno con el comportamiento del resto"* (2023). *Easy Hidro*

Por ello se hace pensar en que se desea encontrar la máxima optimización de la configuración de un enjambre o conjunto de sensores de humedad dentro de la zona planteada de Guasave, Sinaloa con los límites ya otorgados además de agregar un equilibrio entre exploración y explotación: el algoritmo utiliza factores cognitivos (memoria de la mejor posición individual, P_best y sociales, atracción hacia la mejor posición global o local, Gbest o L_best) para balancear la exploración de nuevas regiones con la explotación de soluciones prometedoras, reduciendo la probabilidad de quedar atrapado en óptimos locales.

4 Diseño de algoritmo

El problema de optimización de la ubicación de 100 sensores en el campo agrícola de Guasave se modeló como un problema de optimización continuo y multidimensional de 200 variables (100 pares de latitud y longitud). Cada partícula en el enjambre de partículas (PSO) representa una solución completa, es decir, la configuración de la red de los 100 sensores.

Parámetros, modelado matemático y ecuaciones de partículas:

- ❖ **Parámetros PSO:** Los coeficientes utilizados para el balance de exploración/explotación fueron:

Inercia ($\omega = 0.7$)

Componente cognitiva ($c_1 = 1.5$)

Componente social ($c_2 = 1.5$)

Esto debido a que según la revista Iberoamericana de Inteligencia Artificial: “*Típicamente a ω se asigna un valor fijo de 0.8 y en otros casos se le asigna un valor entre 1 y 1.5, A los pesos c_1 y c_2 se asigna el valor de 2.*” (2006). *Inteligencia Artificial. Revista Iberoamericana*, 10(32).

- ❖ **Partícula (X_i):** Vector de $2N$ dimensiones donde se ven 100 sensores cada uno con Latitud y Longitud se representa de la siguiente manera:

$$X_i = (Lat_1, Lon_1, Lat_2, Lon_2, \dots, Lat_n, Lon_n)$$

- ❖ **Espacio de búsqueda:** Definido por los límites geográficos obtenidos de los datos de Guasave otorgados por el documento:

$$\begin{aligned} & (\text{Lat.: } 25.52 \text{ a } 25.62, \text{ Lon: } -108.52 \text{ a } -108.42) \\ & (\text{Norte a Sur: } 11.13\text{km}, \text{ Este a Oeste: } 10.04\text{km}) \end{aligned}$$

- ❖ **Ecuaciones de movimiento de partícula:**

Actualizar velocidad:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1 (P_{best_i}(t) - P_i(t)) + c_2 r_2 (P_{global}(t) - P_i(t))$$

Actualizar posición:

$$X_i(t+1) = X_i(t) + V_i(t+1)$$

Donde:

$V_i(t+1)$ = Nueva velocidad

$V_i(t)$ = Velocidad actual

ω = (Inercia) Mantiene la dirección de la partícula

c_1 = Componente cognitivo, regula el comp. exploratorio del algoritmo
 c_2 = Componente social, movimiento hacia la mejor pos. encontrada
 $r_1 r_2$ = Factores de aleatoriedad (num alea entre 0 y 1)
 $X_I(t)$ = Posicion actual de la Particula

- ❖ **Función de aptitud:** Se busca maximizar, está diseñada para recompensar las ubicaciones de los sensores que logran la máxima cobertura de la variabilidad de las condiciones del campo. La función integra las tres variables criticas (Humedad, Salinidad, Elevación):

$$F(X) = \sum_{k=1}^N \text{Puntaje}(\text{Sensor}_k) \times \text{Penalización}_{\text{Distancia}}$$

Puntaje x Sensor: Se calcula en dos etapas:

1. Cálculo de importancia del punto (peso): Prioriza la cobertura de áreas que se desvían de las condiciones óptimas o promedio, y se otorga un puntaje alto si el sensor está dentro de la distancia máxima de obertura de un punto importante alto (peso)

$$\text{Peso} = (\text{Peso}_{cultivo} \times 0.4) + (\Delta \text{Humedad} \times 0.3) + (\text{Salinidad} \times 0.2) + (\Delta \text{Elevación} \times 0.1)$$

$\Delta \text{Humedad}$: Es la diferencia absoluta entre la humedad actual y la óptima del cultivo

Optimas de cultivo: (Maiz: 30%), (Tomate: 40%), (Chile: 25%)

$\Delta \text{Elevación}$ = Es la diferencia absoluta respecto a la elevación actual y el promedio 30m

2. Cálculo de distancia entre 2 puntos para ello se seguirá lo siguiente planteado:

“Para calcular la distancia entre dos puntos en Python a partir de su latitud y longitud, se pueden usar diversos métodos como la fórmula de Haversine, el módulo math, la distancia geodésica o la distancia ortodrómica. Estos cálculos son útiles en aplicaciones que involucran datos de ubicación, como navegación, cartografía, geolocalización, seguimiento, logística, entrega, actividades al aire libre e investigación.” Yadav, A. (2023, March 30) AskPython.

Considerando esto se tomó la fórmula de Haversine para determinar la distancia entre 2 puntos (sensores), esto ya que es útil para distancias cortas como lo es la zona de estudio.

$$d = 2r \cdot \arcsin \left(\sqrt{\sin^2 \left(\frac{\Delta\ell}{2} \right) + \cos(\ell_1) \cdot \cos(\ell_2) \cdot \sin^2 \left(\frac{\Delta\lambda}{2} \right)} \right)$$

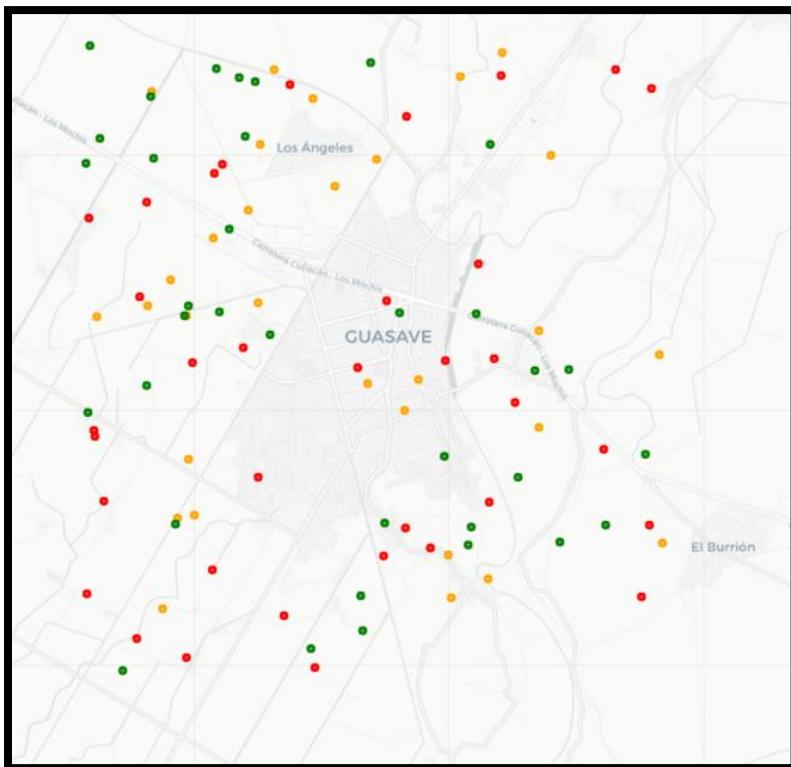
Donde:

- d = distancia entre los dos puntos
 - r = radio de la tierra (aproximadamente 6371 km)
 - φ_1, φ_2 = latitudes de los puntos (en radianes)
 - λ_1, λ_2 = longitudes de los puntos (en radianes)
 - $\Delta\varphi = \varphi_2 - \varphi_1$
 - $\Delta\lambda = \lambda_2 - \lambda_1$
3. Calidad de cobertura: Se premia la proximidad (distancia ≤ 0.5 km) del sensor a los puntos de datos de alto peso, e incluye un bonus por el número total de puntos cubiertos.
 4. Restricción de aglomeración: Se aplica una penalización fuerte ($\times 0.3$) si la distancia entre cualquier par de sensores cae por debajo de 0.05km o 50m, asegurando que la red sea distribuida y no redundante.

5 Resultados

Para probar la eficiencia del algoritmo se mostrarán diferentes resultados de las siguientes compilaciones del algoritmo utilizando cambios de parámetros importantes, algunos reduciendo tiempo de computación, pero logrando optimizaciones clave como objetivo de la resolución del problema. d

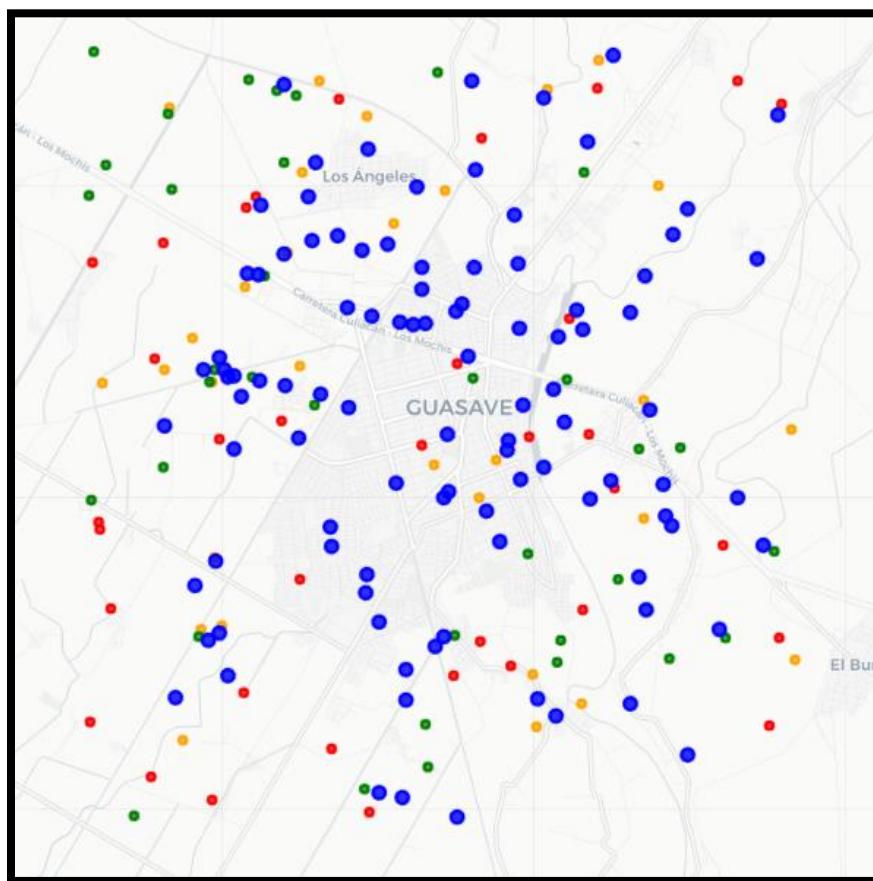
Mapa de la posición de los sensores originalmente:

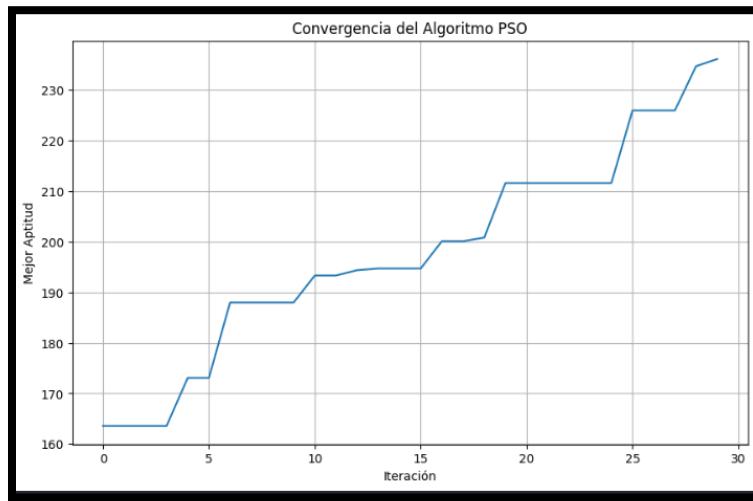


- Primera compilación con los siguientes parámetros:

Parámetro	Valor
Num_sensores	100
Num_particulas	15
Iter_max	30
Inercia	0.7
C_1	1.5
C_2	1.5

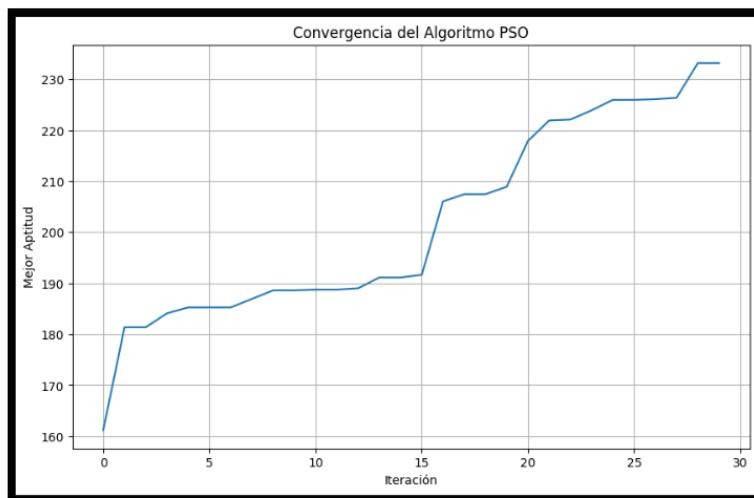
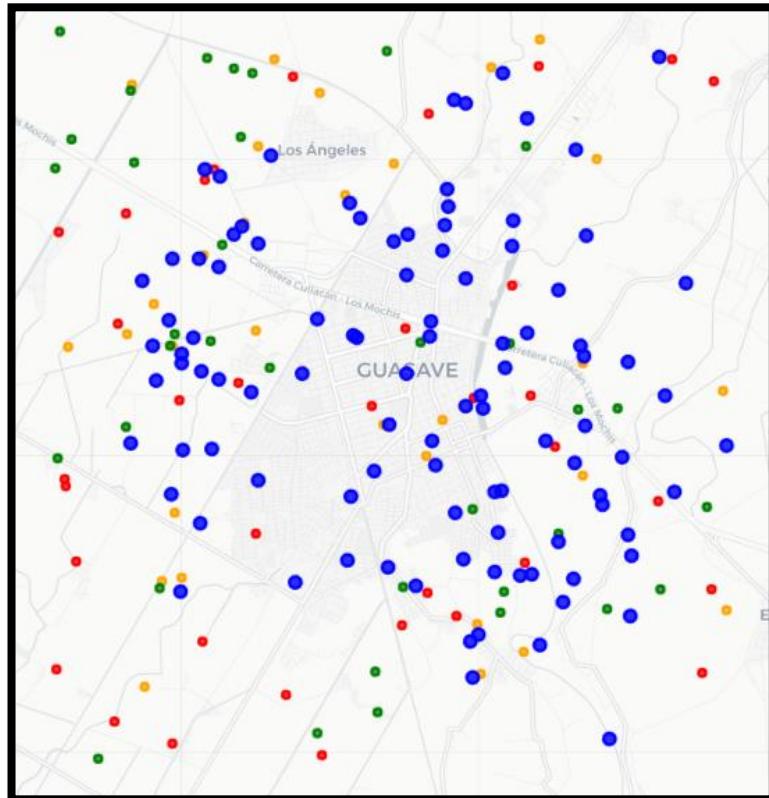
Mapa con la mejor configuración después de haber aplicado el algoritmo PSO y valores de los componentes





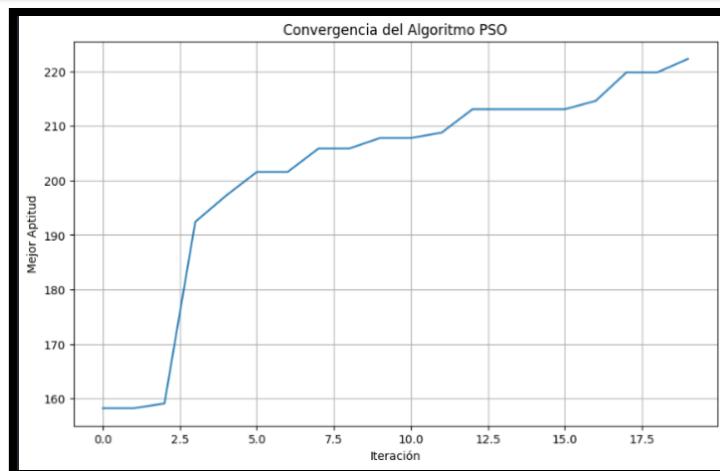
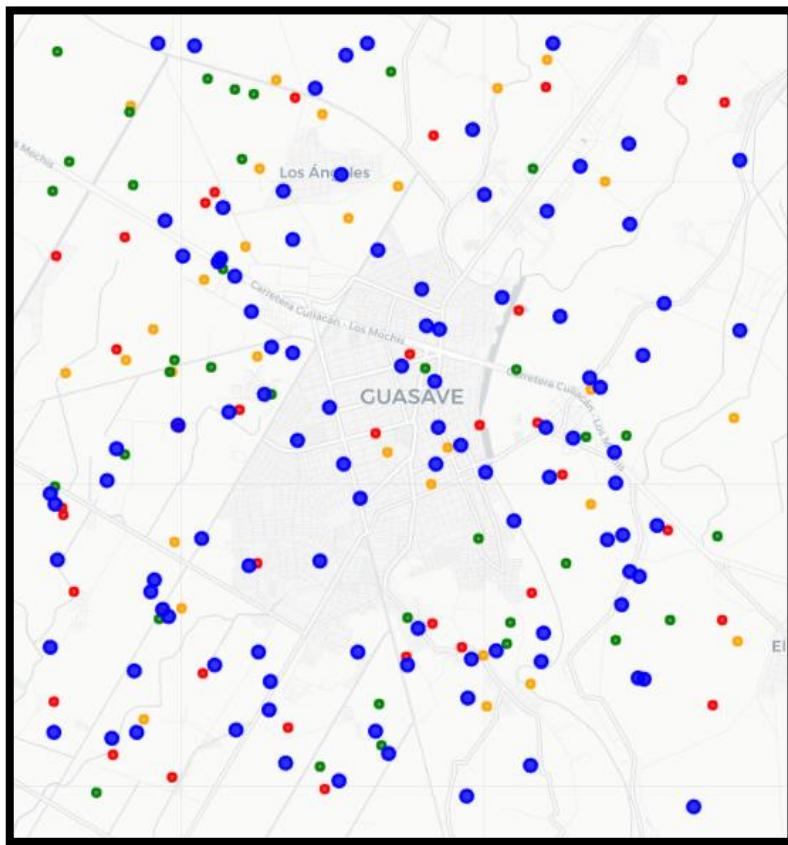
- Segunda compilación con los siguientes parámetros:

Parámetro	Valor
Num_sensores	100
Num_partículas	30
Iter_max	30
Inercia	0.7
C_1	1.5
C_2	1.5



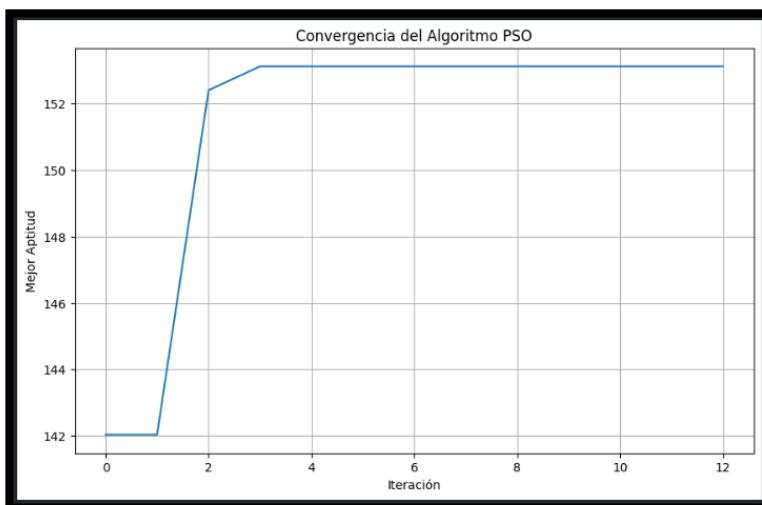
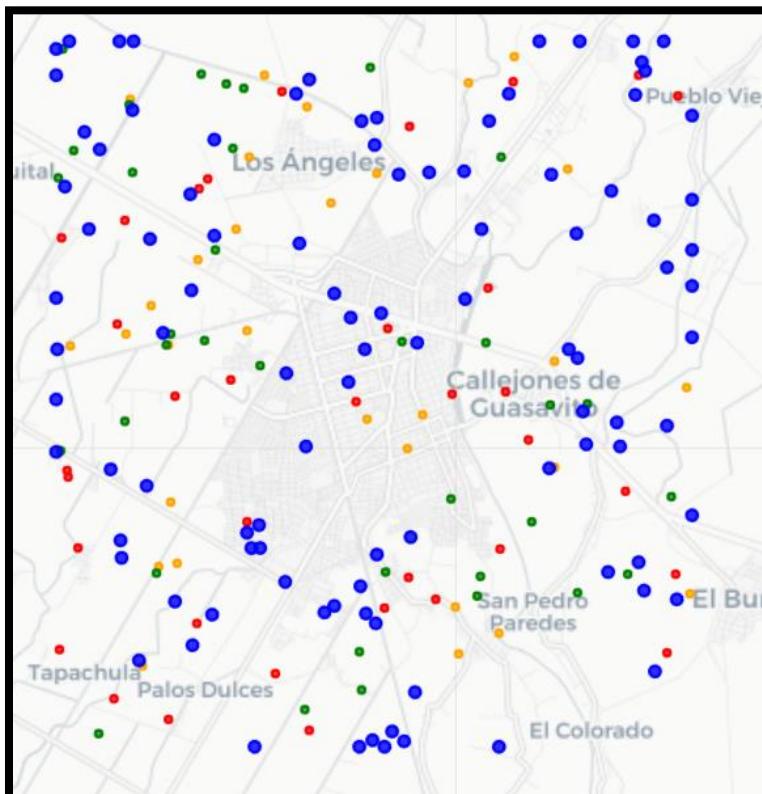
- Tercera compilación con los siguientes campos:

Parámetro	Valor
Num_sensores	100
Num_partículas	35
Iter_max	20
Inercia	0.7
C_1	1.5
C_2	1.5



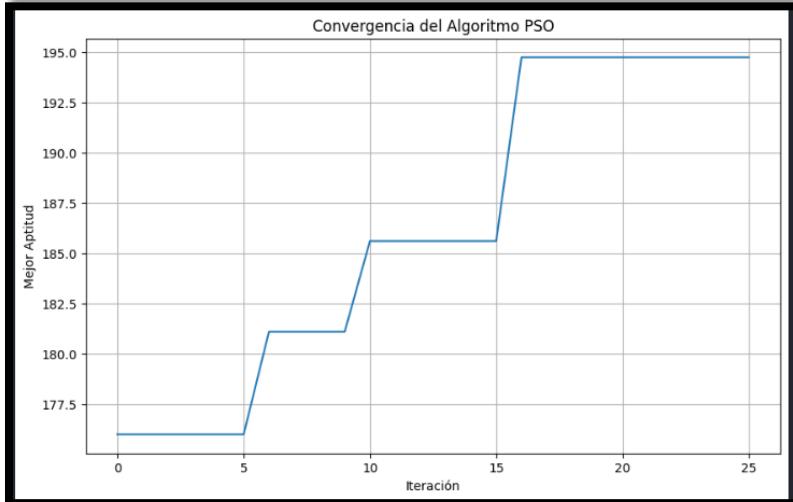
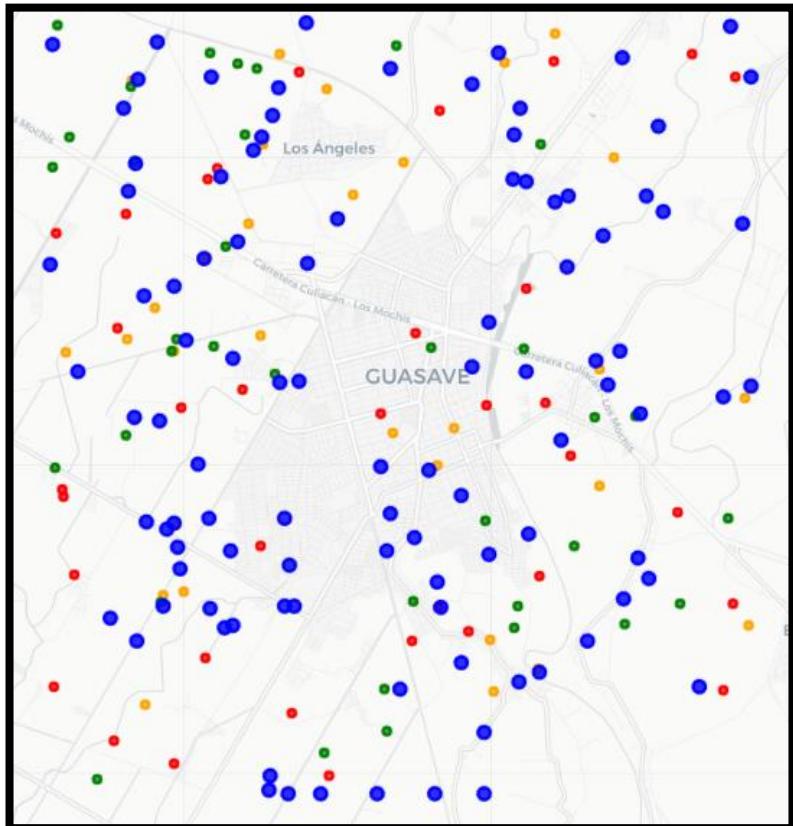
- Cuarta compilación con los siguientes campos:

Parámetro	Valor
Num_sensores	100
Num_partículas	15
Iter_max	30
Inercia	1
C_1	2
C_2	2



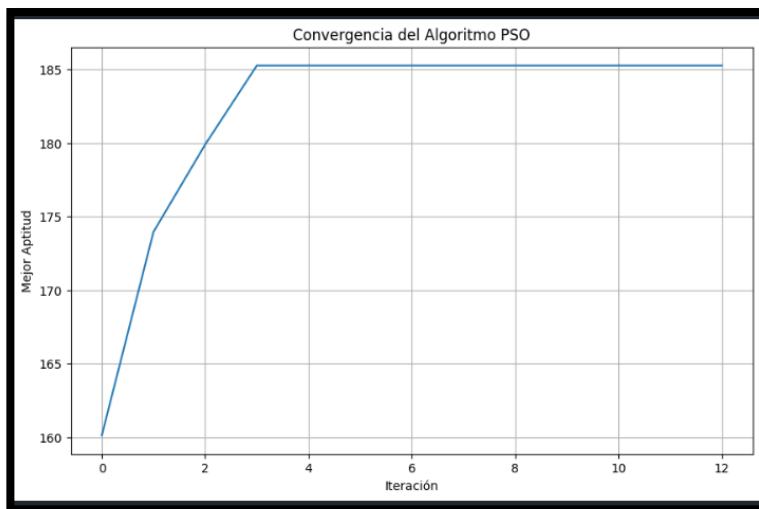
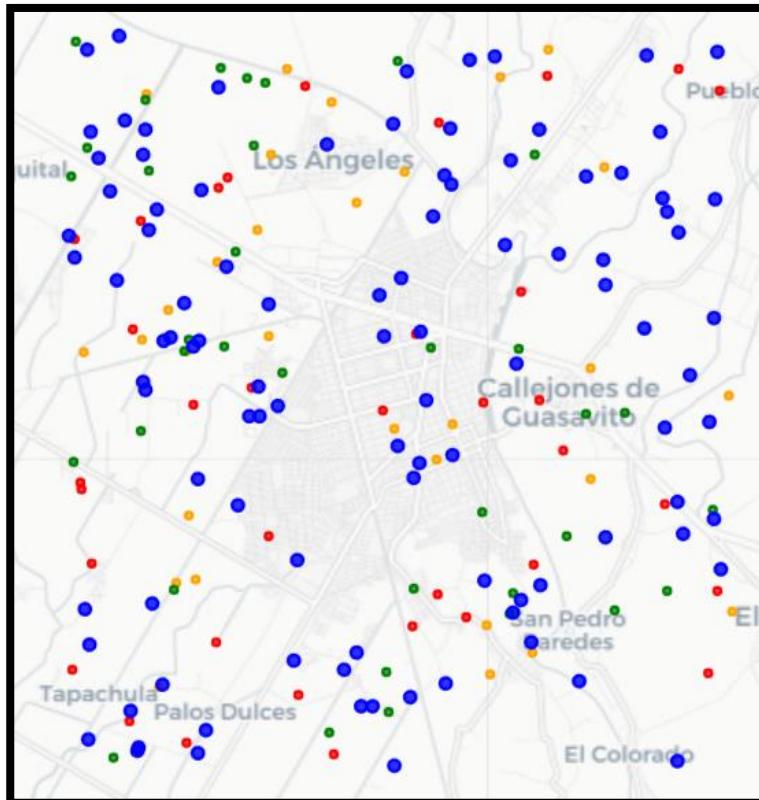
- Quinta compilación con los siguientes campos:

Parámetro	Valor
Num_sensores	100
Num_particulas	25
Iter_max	100
Inercia	1
C_1	2
C_2	2



- Sexta compilación con los siguientes campos:

Parámetro	Valor
Num_sensores	100
Num_partículas	25
Iter_max	1000
Inercia	1
C_1	2
C_2	2



Resumen de eficiencia del PSO

N. Prueba	Partículas	Iteraciones máx.	Inercia (ω)	C1	C2	Mejor aptitud	It. de convergencia
1	15	30	0.7	1.5	1.5	232	28
2	30	30	0.7	1.5	1.5	232	27
3	35	20	0.7	1.5	1.5	220	18
4	15	30	1.0	2.0	2.0	152	4
5	25	100	1.0	2.0	2.0	195	16
6	25	1000	1.0	2.0	2.0	185	4

6 Análisis de eficiencia

El objetivo principal de la optimización fue encontrar la distribución de 100 sensores que maximizara la cobertura ponderada, priorizando las áreas con mayor potencial de riesgo hídrico, para ello la fase de experimentación del algoritmo PSO se centró en la calibración de parámetros para encontrar la combinación que permitiera alcanzar la aptitud máxima de manera eficiente.

La mejor solución, con una aptitud final de 232, fue lograda por las pruebas 1 y 2. Este éxito dependió de un factor de inercia $\omega = 0.7$ y coeficientes cognitivos/sociales (c_1, c_2) de 1.5. Esta calibración proporcionó el balance ideal entre la tendencia a mantener la dirección (inercia) y la atracción a las mejores posiciones encontradas.

La importancia de esta calibración se hizo evidente al observar la cuarta compilación, que utilizó los valores más comunes de la literatura ($\omega = 0.8$, $c_1 = c_2 = 2$). Esta configuración mostró una convergencia rápida y prematura en la Iteración 4 con una aptitud subóptima de 152, lo que indica que el enjambre se estancó en un mínimo local. El uso de $\omega = 0.7$ fue, por lo tanto, la clave para un proceso de búsqueda lo suficientemente lento y exhaustivo, permitiendo al algoritmo escapar de los óptimos locales y encontrar la configuración de sensores de mayor calidad.

Además, el análisis de las compilaciones exitosas revela que la primera compilación fue más eficiente, logrando el mismo puntaje máximo con un número menor de partículas (15 vs. 30), optimizando así los recursos computacionales.

7 Conclusión

El proyecto ha resuelto exitosamente el desafío de optimizar la ubicación de 100 sensores de humedad en los campos agrícolas de Guasave, Sinaloa, al formularlo como un problema de optimización continuo de 200 variables (Latitud y Longitud por sensor).

La solución final, obtenida con los parámetros optimizados del PSO, cumple rigurosamente con el objetivo de máxima cobertura y representatividad de la variabilidad hídrica del campo. La función de aptitud priorizó de manera efectiva las áreas con factores de degradación como la salinidad y la disminución de materia orgánica, así como las zonas alejadas de la humedad óptima del cultivo. Crucialmente, la distribución resultante de los sensores evita la redundancia y aglomeración al aplicar una penalización fuerte a cualquier sensor ubicado a menos de 50 metros, garantizando que cada punto de monitoreo proporcione información única y valiosa para la toma de decisiones de riego.

En resumen, la red de sensores resultante está ubicada estratégicamente para maximizar la eficiencia del riego y el aprovechamiento de los recursos hídricos en la región de Guasave.

8 Referencias

- APLICACIÓN ALGORITMO DE ENJAMBRE DE PARTICULAS PARA OPTIMIZACIÓN PATZ EN REDES DE RIEGO. (2023). *Easy Hidro.* <https://www.serea23.com/wp-content/uploads/2023/06/6257.pdf>
- Optimización de Enjambre de Partículas aplicada al problema del Viajero Bi-objetivo. (2006). *Inteligencia Artificial. Revista Iberoamericana,* 10(32). <https://www.redalyc.org/pdf/925/92503209.pdf>
- Yadav, A. (2023, March 30). Distance between two Geo-Locations in Python - AskPython. AskPython. <https://www.askpython.com/python/examples/find-distance-between-two-geo-locations>