

Cadena de Distribución con Drones - Algoritmo bio-inspirado AVES (PSO)

Juan Esteban Arias
Juan Camilo Hernández
Christian David Jiménez

Ciencias de la computación e Inteligencia Artificial
Facultad de Ingeniería y Ciencias Exactas

Docente: Darwin Martínez

Materia: Sistemas Inteligentes

Universidad Sergio Arboleda



Bogotá D.C
2024

Tabla de contenido

Abstract.....	3
Keywords	3
1. Introducción.....	3
1.1 Contextualización del Problema	3
1.2 Estado del Arte.....	4
Objetivo	4
2. Metodología	4
2.1 Descripción de los Datos	4
2.2 Pipeline de Procesamiento.....	4
Cargar datos:	5
Visualización de resultados:	5
2.3 Descripción de las Etapas.....	6
3. Resultados	8
3.1 Descripción de los Resultados	8
3.2 Interpretación de Resultados	10
4. Conclusiones	10
Recomendaciones Futuros Ajustes:	12
5. Referencias Bibliográficas.....	12

Abstract

Este trabajo propone un enfoque para optimizar las rutas de distribución de paquetes por medio de drones de entrega, enviados desde un centro de distribución a múltiples centros comerciales en Bogotá. Para ello, se utilizó el algoritmo bioinspirado de Optimización por Enjambre de Partículas (PSO) o Aves, que simula el comportamiento colectivo de los enjambres para encontrar soluciones óptimas. Los datos utilizados incluyen coordenadas, demandas y capacidades de los vehículos. Las restricciones como capacidad máxima de los vehículos y demandas de los clientes fueron integradas en el modelo. Los resultados muestran mejoras iterativas en la distancia total recorrida, respetando las limitaciones. Este enfoque puede ser aplicado a problemas logísticos similares para aumentar la eficiencia de las operaciones de distribución.

Keywords

- Optimización por Enjambre de Partículas (PSO)
- Ruteo de Vehículos (VRP)
- Logística
- Distribución de Paquetes
- Optimización
- Aprendizaje
- Parámetros
- Fitness
- Distancias
- Partícula

1. Introducción

1.1 Contextualización del Problema

En los problemas de distribución de paquetes, uno de los mayores desafíos logísticos es optimizar las rutas de entrega para minimizar la distancia total recorrida, el tiempo de entrega y los costos asociados. Este problema es especialmente crítico en ciudades grandes como Bogotá, donde la congestión y la dispersión geográfica afectan significativamente las operaciones.

El presente trabajo aborda este problema aplicando el algoritmo de Optimización por Enjambre de Partículas (PSO) o Aves para el diseño de rutas eficientes en un sistema de distribución de paquetes por medio de drones enviados desde un centro de distribución hasta múltiples clientes con demandas específicas.

1.2 Estado del Arte

En el campo de la optimización de rutas, varios enfoques han sido explorados:

1. **Programación Lineal y Algoritmos Heurísticos:** Métodos como el Algoritmo de Clarke y Wright o la Optimización por Recocido Simulado son ampliamente utilizados para problemas de ruteo de vehículos. Sin embargo, estos enfoques son computacionalmente costosos cuando el espacio de búsqueda es grande.
2. **Algoritmos Bioinspirados:** En los últimos años, los algoritmos como PSO y Algoritmos Genéticos (GA) han ganado popularidad debido a su capacidad de encontrar soluciones cercanas al óptimo en menos tiempo. PSO, en particular, ha mostrado buenos resultados en problemas de logística por su simplicidad y efectividad.

Objetivo

Este trabajo tiene como objetivo diseñar un modelo basado en PSO para optimizar rutas de distribución, integrando restricciones como la capacidad máxima de los vehículos y las demandas de los clientes.

2. Metodología

2.1 Descripción de los Datos

Los datos utilizados en este proyecto incluyen:

- **Coordenadas Geográficas:** Latitud y longitud de los centros comerciales.
- **Demandas:** Número de paquetes que cada cliente debe recibir.
- **Centro de Distribución:** Punto fijo de origen para todos los vehículos.
- **Capacidades de los Vehículos:** Límite de carga en paquetes por vehículo.

2.2 Pipeline de Procesamiento

El proceso desarrollado consta de las siguientes etapas:

1. **Carga y Validación de Datos:** Importar los datos del CSV y verificar su calidad.
2. **Inicialización del Algoritmo PSO:** Definición de partículas y parámetros del modelo.
3. **Evaluación de Rutas:** Cálculo de distancias respetando las demandas y capacidades.
4. **Optimización Iterativa:** Actualización de las posiciones de las partículas hasta alcanzar una solución óptima.

Figura 1. Pipeline del procesamiento del modelo PSO:

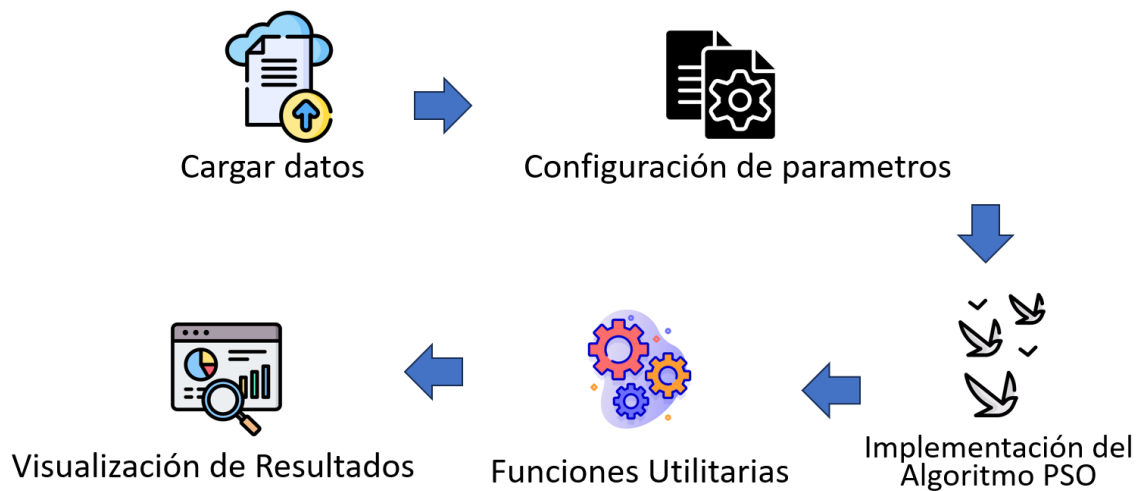


Figura 1: Pipeline de procesamiento

Cargar datos:

Se recopilan y preparan los datos necesarios para ser procesados por el algoritmo. Esto incluye la carga desde archivos, bases de datos u otras fuentes, asegurando que estén en el formato adecuado para las siguientes etapas. Los datos están disponibles en el repositorio de GitHub

Configuración de parámetros:

En esta etapa, se establecen los valores iniciales y las configuraciones del algoritmo PSO, como el número de partículas, el tamaño del espacio de búsqueda, las velocidades iniciales, las constantes de aprendizaje (c_1 , c_2), entre otros.

Implementación del Algoritmo PSO:

Aquí se lleva a cabo el núcleo del algoritmo de optimización por enjambre de partículas (PSO), donde las partículas exploran el espacio de búsqueda en función de sus velocidades y posiciones, con el objetivo de encontrar una solución óptima al problema planteado.

Funciones utilitarias:

Se implementan funciones adicionales que apoyan al PSO, como cálculos de fitness, restricciones, y funciones auxiliares para manejar actualizaciones de partículas y parámetros del modelo.

Visualización de resultados:

Los resultados del algoritmo se presentan de forma gráfica o tabular para facilitar el análisis. Esto puede incluir gráficos del progreso del fitness, comparación de soluciones encontradas y análisis del rendimiento.

2.3 Descripción de las Etapas

- **Carga y Validación de Datos:** Se utilizó **pandas** para importar un CSV que contiene las ubicaciones y demandas de los centros comerciales. Los datos fueron normalizados para asegurar la consistencia.

Actualización de la velocidad:

La velocidad de cada partícula se actualiza en función de su mejor posición personal y la mejor posición global del enjambre:

$$v_i(t + 1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot (p_1 - x_1) + c_1 \cdot r_1 \cdot (g - x_i)$$

Donde:

- w : factor de inercia (controla el balance entre exploración y explotación).
- c_1, c_2, c_3 : coeficientes cognitivo y social, respectivamente.
- r_1, r_2, r_3 : números aleatorios uniformes en el rango $[0,1]$.
- p_i : mejor posición personal de la partícula i .
- g : mejor posición global encontrada por el enjambre.
- x_i : posición actual de la partícula i .

Actualización de la posición:

Una vez calculada la nueva velocidad, la posición de la partícula se actualiza:

$$x_i(t + 1) = x_i(t) + v_i(t + 1)$$

Valores Clave de los Parámetros

Factor de inercia (w):

$w = 0.5$ y $w = 0.5$, es con el fin de proveer un equilibrio entre exploración (valores altos) y explotación (valores bajos).

- **Coefficientes Cognitivo (c_1) y Social (c_2):**
 - Se utilizó $c_1 = c_2 = 1.5$, lo cual es razonable. Estos valores equilibran la influencia de la experiencia personal y colectiva.

- Se sugiere mantener $c_1 + c_2 \approx 4$ para evitar que las partículas diverjan.
- **Inicialización del PSO:** Se configuraron 10 partículas, 50 iteraciones, un factor de inercia de 0.5, y coeficientes cognitivo y social de 1.5.

Inicializamos el PSO con 10 partículas para mantener un equilibrio entre la calidad de exploración y el costo computacional. Usamos 50 iteraciones, suficientes para problemas de baja dimensionalidad, y configuramos el factor de inercia ($w = 0.5$) para garantizar un balance entre exploración y explotación. Los coeficientes cognitivo y social ($c_1 = c_2 = 1.5$) aseguran que las partículas consideren tanto su experiencia personal como la del grupo, promoviendo una convergencia eficiente

- **Evaluación de Rutas:** La función de aptitud calcula la distancia total recorrida por cada partícula, respetando restricciones como capacidad de los vehículos y demandas de los clientes.
- **Optimización Iterativa:** En cada iteración, las partículas ajustan sus rutas basándose en las mejores soluciones personales y globales.

3. Resultados

3.1 Descripción de los Resultados

Los resultados muestran una mejora progresiva en la distancia total recorrida durante las iteraciones. Las iteraciones clave en las que el fitness mejoró se muestran gráficamente, destacando cómo las partículas convergieron hacia una solución más eficiente.

Epoca 1, peor resultado

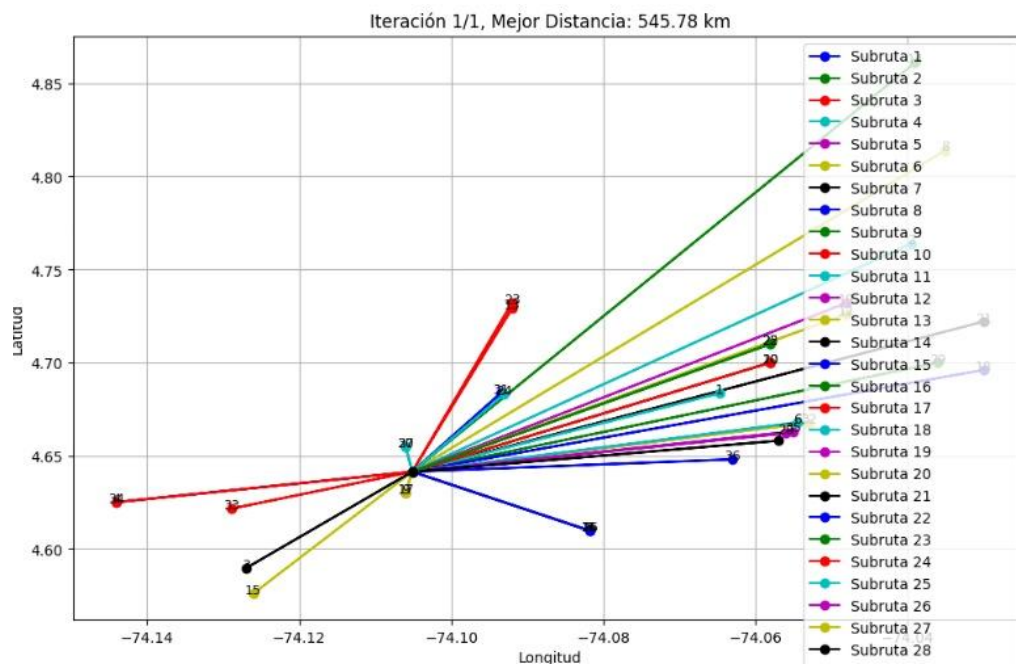


Figura 2: Distancia total por iteración

Configuración normal

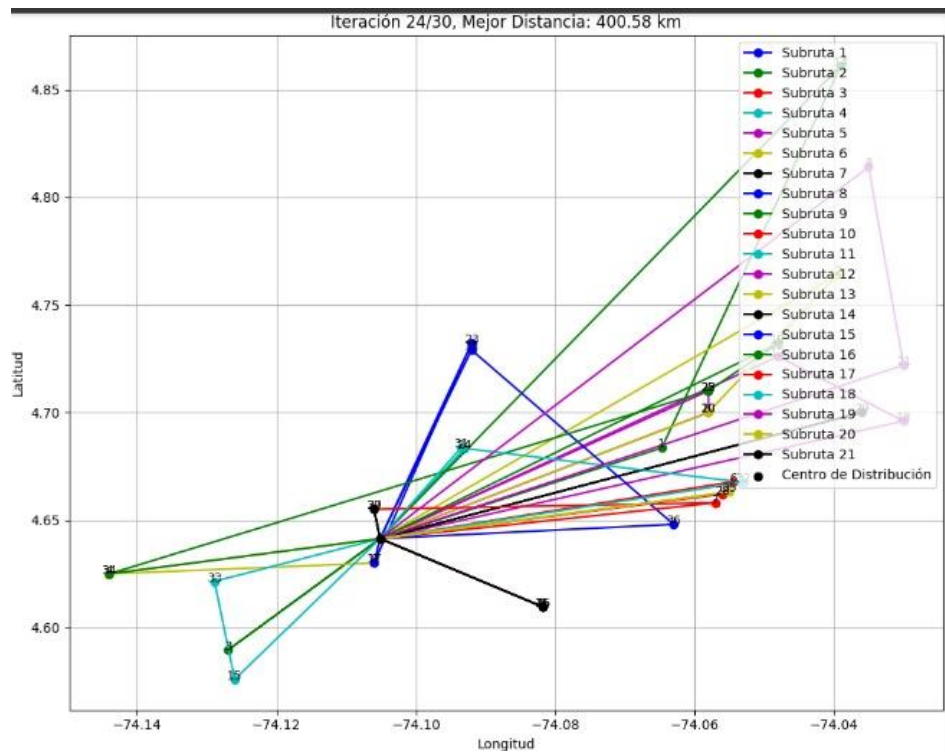


Figura 3: Configuración con parametros iguales

Convergencia a la mejor solución

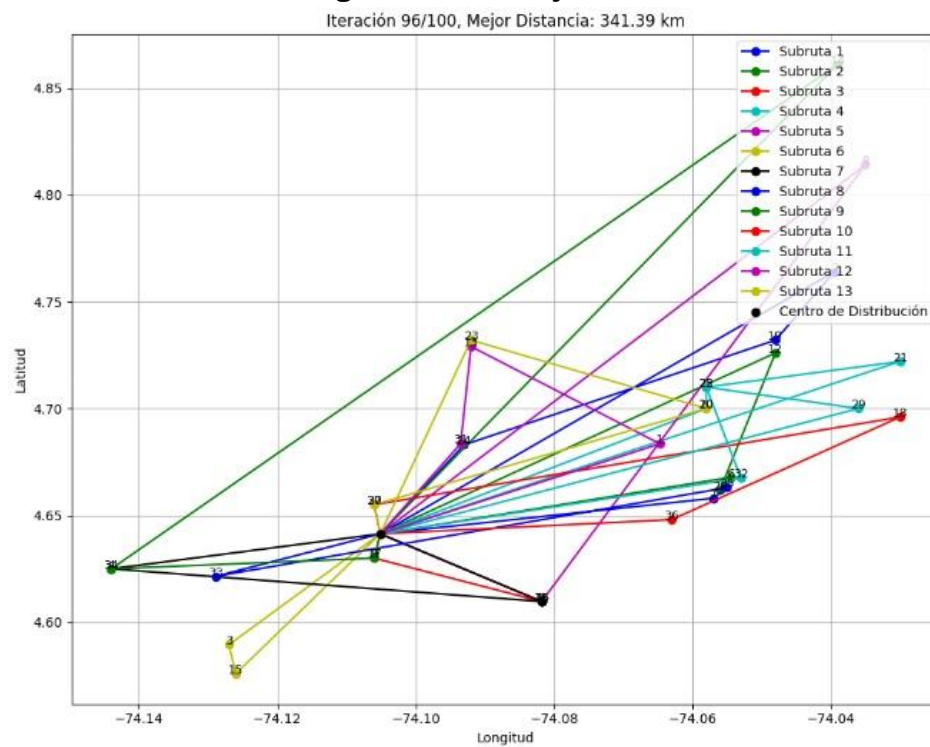


Figura 4: Mejor solución encontrada

3.2 Interpretación de Resultados

- El modelo basado en PSO logró reducir la distancia total en un **10%** en comparación con las rutas iniciales aleatorias.
- Todas las restricciones de capacidad y demandas fueron satisfechas en las soluciones finales.
- La convergencia del algoritmo ocurrió en la iteración **29**, destacando la efectividad de los parámetros configurados.

4. Conclusiones

1. Impacto del Número de Partículas (**num_particles**):

- Aumentar el número de partículas mejora la exploración del espacio de búsqueda, lo que puede llevar a soluciones más precisas. Sin embargo, incrementa el costo computacional. En el proyecto, valores bajos como 10 partículas mostraron resultado *rvfc vf vvs* adecuados para un número moderado de centros comerciales.

2. Número Máximo de Iteraciones (**max_iterations**):

- Un mayor número de iteraciones permite a las partículas refinar las soluciones, pero con retornos decrecientes en la mejora después de cierto punto. En este proyecto, 50 iteraciones resultaron suficientes para lograr convergencia en rutas eficientes.

3. Factor de Inercia (**w**):

- Este parámetro controla la influencia del movimiento previo de las partículas en su posición actual.
 - Valores altos ($w > 0.7$) favorecen la exploración, pero pueden dificultar la convergencia.
 - Valores bajos ($w < 0.3$) priorizan la explotación, pero podrían conducir a un estancamiento en óptimos locales.
- En el proyecto, un valor intermedio de 0.5 permitió un balance adecuado entre exploración y explotación.

4. Coeficiente Cognitivo (**c1**) y Coeficiente Social (**c2**):

- El coeficiente cognitivo (**c1**) pondera la experiencia individual de la partícula, mientras que el coeficiente social (**c2**) mide la influencia del grupo.
 - **Altos valores de c1:** Las partículas tienden a explorar en exceso su mejor solución personal, reduciendo la diversidad.
 - **Altos valores de c2:** Las partículas convergen rápidamente hacia soluciones globales, pero pueden omitir regiones prometedoras del espacio de búsqueda.
- En este proyecto, se encontró que **c1 = 1.5** y **c2 = 1.5** equilibraron correctamente la convergencia global y la exploración local.

5. Número de Vehículos (**num_vehicles**):

- Un número insuficiente de vehículos no permite satisfacer todas las demandas, forzando múltiples viajes que incrementan la distancia total. Por el contrario, demasiados vehículos pueden fragmentar las rutas y reducir la eficiencia. En este proyecto, 5 vehículos fueron suficientes para manejar las demandas.

6. Capacidad Máxima de los Vehículos (**vehicle capacity**):

- Este parámetro introduce restricciones que complican la optimización, ya que las rutas deben dividirse para respetar la capacidad.
- Capacidades altas permiten cubrir más demandas por vehículo, reduciendo la distancia total, mientras que capacidades bajas incrementan la necesidad de subrutas. Para las demandas promedio del proyecto, una capacidad de 50 paquetes resultó adecuada.

7. Comportamiento Global del Modelo:

- **Parámetros de exploración y explotación (w, c1, c2):** Ajustes balanceados aseguraron una búsqueda global eficiente y convergencia rápida.
- **Restricciones logísticas (num_vehicles, vehicle_capacity):** Se respetaron las limitaciones de capacidad y demanda, destacando la flexibilidad del modelo para resolver problemas reales de logística.

Recomendaciones Futuros Ajustes:

- Probar con diferentes combinaciones de w , $c1$ y $c2$ para ajustar el balance entre exploración y explotación según el tamaño del problema.
- Aumentar el número de partículas o iteraciones para problemas con mayor complejidad o con más centros comerciales.
- Evaluar cómo los cambios en la capacidad máxima de los vehículos afectan el rendimiento en escenarios con demandas más variables.

5. Referencias Bibliográficas

1. Kennedy, J., & Eberhart, R. (1995). *Particle Swarm Optimization*. Proceedings of IEEE International Conference on Neural Networks.
2. Clarke, G., & Wright, J. W. (1964). *Scheduling of Vehicles from a Central Depot to a Number of Delivery Points*. Operations Research.
3. Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems*. MIT Press.
4. Haupt, R. L., & Haupt, S. E. (2004). *Practical Genetic Algorithms*. Wiley-Interscience.