



**TECNOLÓGICO
NACIONAL DE MÉXICO**

Instituto tecnológico de Culiacán

Sistema de enrutamiento para tiendas de autoservicios

Materia:

Temas de Inteligencia Artificial

Integrantes:

Ramirez Medina Cristian Andrea

Castro Figueroa Daniel Sebastian

Maestro:

Dr. Mora Felix Zuriel Dathan

Repositorio: <https://github.com/DanielSdc/TopicosIA>

1. Introducción

El desarrollo de una aplicación para la optimización de rutas de distribución surge de la necesidad en el sector logístico de minimizar los costos operativos y mejorar la eficiencia en la cadena de suministro [1]. Este proyecto se centra en encontrar la forma más eficiente de surtir a todas las tiendas desde los centros de distribución, reduciendo significativamente los gastos asociados a la distancia recorrida.

El problema de encontrar la ruta óptima para visitar un conjunto de nodos es un caso del Problema del Viajante, una clase de problemas de optimización combinatoria de alta complejidad[2]. Para abordarlo, se emplearán algoritmos de optimización heurística, específicamente el Recocido Simulado, para determinar la secuencia de visitas que minimice los costos de la ruta. La aplicación se fundamentará en el cálculo de la distancia euclidiana para establecer la matriz de distancias, sirviendo como base para el diseño y la selección de las rutas más viables.

2. Objetivo General

Desarrollar una aplicación funcional para la optimización de rutas de distribución que determine la secuencia de visitas más eficiente entre centros de distribución y tiendas, con el fin de reducir los costos de distancia recorrida.

3. Objetivos Específicos

1. Diseñar y crear la matriz de distancias basándose en la ubicación de los centros de distribución y las tiendas utilizando la distancia euclidiana como métrica de separación.
2. Implementar el algoritmo de recocido simulado para encontrar la ruta óptima que conecte todos los puntos de tiendas desde los centros de distribución, buscando la minimización de costos.
3. Desarrollar un sistema que permita la visualización de las rutas propuestas sobre un mapa y generar la documentación y el código necesario para la operación del sistema.
4. Realizar pruebas de la aplicación con un conjunto limitado de nodos para demostrar la funcionalidad y la reducción efectiva de los costos de distancia.

4. Justificación

La realización de este proyecto se justifica por la necesidad de optimizar la eficiencia logística en la distribución de productos. El modelo se enfoca directamente en la reducción de costos operativos, particularmente en la distancia recorrida. El costo logístico en México alcanza el 9.7% en algunas grandes empresas[3], y el transporte al ser uno de los mayores gastos variables en la cadena de suministro, la aplicación de una herramienta de optimización tiene un impacto directo en la rentabilidad y la sostenibilidad del negocio.

Para problemas tan complejos de optimización combinatoria como el MDVRP, en los que no se conoce una solución exacta, las metaheurísticas surgen como una solución que se basa en la exploración de distintas soluciones sin la necesidad de comprender el problema a fondo.

5. Alcance

- Creación del código y la estructura modularizada para la optimización y visualización de rutas.
- Generación de matrices de distancias. Los centros de distribución son considerados los nodos iniciales de las rutas.
- Implementación del método de Recocido Simulado para la optimización de rutas.
- El proyecto entregará la ruta óptima en términos de mínima distancia.

6. Desarrollo

6.1. Descripción del Problema

En los puntos siguientes se abordará un desafío logístico conocido como el Problema de Enrutamiento de Vehículos con Múltiples Depósitos (MDVRP) [4]. El objetivo será determinar un conjunto de rutas óptimas para una serie de vehículos con el fin de minimizar los costos operativos, garantizando la cobertura completa del servicio.

El escenario se compone de las coordenadas de longitud y latitud de los siguientes elementos:

- 10 Centros de Distribución (CDs): Son el punto de origen y destino de cada ruta.
- 90 Tiendas: Son los nodos que deben de ser visitados por los vehículos de cada CD.

Puntos a considerar para cada solución:

- Todas las tiendas deben ser visitadas exactamente una vez (no se permiten múltiples visitas).
- Cada ruta debe iniciar en un CD y acabar en el mismo CD de origen.
- No es necesario que todos los CD sean utilizados; puede haber CDs que sólo se visiten a sí mismos (costo operativo cero).
- Se asume que cada CD cuenta con un único vehículo encargado de realizar la ruta.

6.2. Generación matriz de distancias

Como se mencionó en la descripción del problema, se parte de un documento que cuenta unicamente con las coordenadas de latitud y longitud, por lo que será necesario crear una matriz de distancia para medir el costo de una forma mas sencilla, para ello se utilizará la formula de Haversine, que nos permite obtener las distancias euclidianas de una esfera [5], es decir, la distancia de una linea recta de un punto A a un punto B. La fórmula de Haversine para calcular la distancia d entre dos puntos con coordenadas (ϕ_1, λ_1) y (ϕ_2, λ_2) en una esfera de radio R es:

$$d = 2R \cdot \arcsin \left(\sqrt{\sin^2 \left(\frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

6.3. Selección algoritmo heurístico

Para resolver este problema de optimización se ha optado por el recocido simulado dado que en escenarios como el Problema de Ruteo de Vehículos (VRP) este ha demostrado ser más eficiente que otros algoritmos de búsqueda metaheurísticos como la Búsqueda Tabú o el Vecino Más Próximo [6].

El recocido simulado es un algoritmo de optimización que imita el proceso físico de enfriamiento lento de materiales, como metales o vidrios, para alcanzar una configuración de baja energía (valor óptimo de una función en un espacio de búsqueda grande), o mejor dicho, el “óptimo global”. El algoritmo sigue la secuencia iterativa partiendo ya de una solución inicial, y mientras no se cumpla algún criterio de parada:

1. Generación vecino: En cada iteración dentro una una temperatura es creado un nuevo vecino partiendo de la solución actual, esto mediante swaps.
2. Criterio aceptación: El criterio de aceptación del recocido simulado, es cumpliendo alguna de las siguientes condiciones:
 - Si el cambio mejora la solución ($\Delta < 0$), se acepta.
 - Si el cambio empeora la solución, puede aceptarse con una probabilidad $e^{-\Delta/T}$, siendo Δ la diferencia de costos entre la solución actual y el vecino, y T la temperatura actual.
3. Secuencia de enfriamiento: Se parte de una temperatura (T) la cual es la base para el criterio de aceptación de cualquier solución, ya que a temperaturas altas la probabilidad de aceptación es mayor, mientras que a temperaturas bajas es menor [7].

6.4. Implementación computacional

6.4.1. Representación

Cada solución se representará mediante un arreglo que contenga los identificadores de las 90 tiendas una sola vez, y estos se encontrarán en medio de al menos el identificador de su CD asignado.

[1, 7, 9, 22, 1, 2, \dots , 10, 45, 67, 10]

La solución inicial se creará a partir de los siguientes dos procesos:

1. Cada tienda será asignada a su CD más cercano en distancia euclidiana.
2. De forma aleatoria se creará una ruta en cada uno de los 10 CDs con sus tiendas asignadas.

6.4.2. Función Objetivo

Para medir el costo de cada solución se utilizará la matriz de distancias en kilómetros creada a partir de las coordenadas de los 100 puntos y sus respectivas distancias con los demás puntos (matriz de 100x100). Primeramente, se separará el arreglo de la solución actual (vecino) en las 10 rutas (una por cada CD), y se calculará la distancia acumulada accediendo a la matriz a partir del identificador del punto actual y el siguiente.

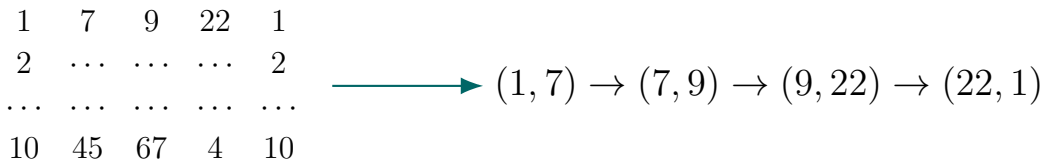


Figura 1: Representación de rutas y sus pares de adyacencia.

6.4.3. Parámetros

Para la ejecución del algoritmo se seleccionaron los siguientes parametros y sus respectivos valores:

1. **Temperatura inicial (10.0):** Se optó por una temperatura inicial relativamente baja, ya que no partimos de una solución aleatoria, lo que nos permite ser más estrictos para la selección de soluciones.
2. **Temperatura final (0.001):** Al tener una temperatura inicial "baja" es necesario tener una temperatura final muy baja para poder darle tiempo al algoritmo de encontrar mejores soluciones.
3. **Enfriamiento (0.99):** Se optó por enfriamiento poco brusco para de igual manera que con la temperatura, poder darle tiempo suficiente de explorar varias soluciones.
4. **Iteraciones por temperatura (1000):** Con esta cantidad de iteraciones le damos la posibilidad al algoritmo de hacer una búsqueda de soluciones a fondo en cada temperatura.

6.4.4. Generación de Vecinos

La generación de vecinos es crucial, ya que determina la calidad de las nuevas soluciones. Se utilizarán dos métodos con igual probabilidad (50 % cada uno):

1. *Swap básico*: se seleccionan dos índices de la solución actual (ambos tiendas) y se intercambian.



Figura 2: Representación de un swap en una lista.

2. *Mover tienda*: se selecciona una tienda, se remueve y se inserta en alguna de todas las posiciones que cumplan con alguna de las siguientes condiciones:

- En medio de dos tiendas.
- En medio de una tienda y un CD.
- En medio del mismo CD.



Figura 3: Representación movimiento tienda en una lista.

6.4.5. Criterio de Aceptación

El criterio de aceptación será el estándar del recocido simulado, cumpliendo alguna de las siguientes condiciones:

- Si el cambio mejora la solución ($\Delta < 0$), se acepta.
- Si el cambio empeora la solución, puede aceptarse con una probabilidad $e^{-\Delta/T}$, siendo Δ la diferencia de costos entre la solución actual y el vecino, y T la temperatura actual.

6.5. Resultados obtenidos

El algoritmo fue capaz de obtener una solución de 153.22 Km, con los siguientes paramatros:

Cuadro 1: Tabla Rutas de la solución de menor costo.

Centro de Distribución	Ruta	Costo
CD 01	1, 70, 1	0.98 km
CD 02	2, 47, 48, 94, 45, 60, 41, 97, 100, 57, 26, 18, 27, 92, 64, 61, 99, 28, 58, 79, 38, 44, 2	33.72 km
CD 03	3, 3	0.00 km
CD 04	4, 4	0.00 km
CD 05	5, 5	0.00 km
CD 06	6, 6	0.00 km
CD 07	7, 59, 36, 88, 16, 11, 15, 86, 39, 90, 78, 71, 76, 53, 68, 32, 7	27.68 km
CD 08	8, 56, 84, 40, 91, 62, 19, 42, 73, 55, 50, 29, 75, 30, 67, 8	24.53 km
CD 09	9, 33, 14, 77, 21, 65, 85, 51, 37, 22, 17, 35, 96, 83, 34, 46, 80, 23, 93, 81, 54, 13, 69, 82, 74, 20, 89, 49, 72, 95, 25, 87, 66, 31, 98, 43, 12, 63, 52, 24, 9	66.31 km
CD 10	10, 10	0.00 km
Costo total		153.22 km

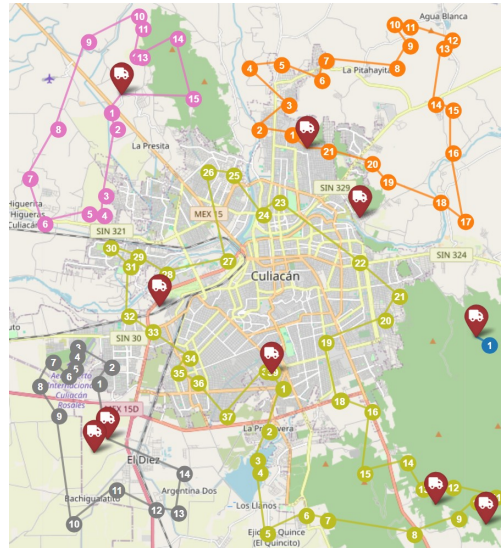


Figura 4: Mapa mejor solución (153.22 Km)

7. Análisis de efectividad

Para poder analizar la efectividad del algoritmo implementado, se llevaron a cabo 50 ejecuciones, de las cuales se llegaron a las siguientes conclusiones:

1. **Mejora solución inicial:** El algoritmo logró conseguir un porcentaje de mejora media relativa de 57.24 % y mejora media absoluta de 219.76 Km.

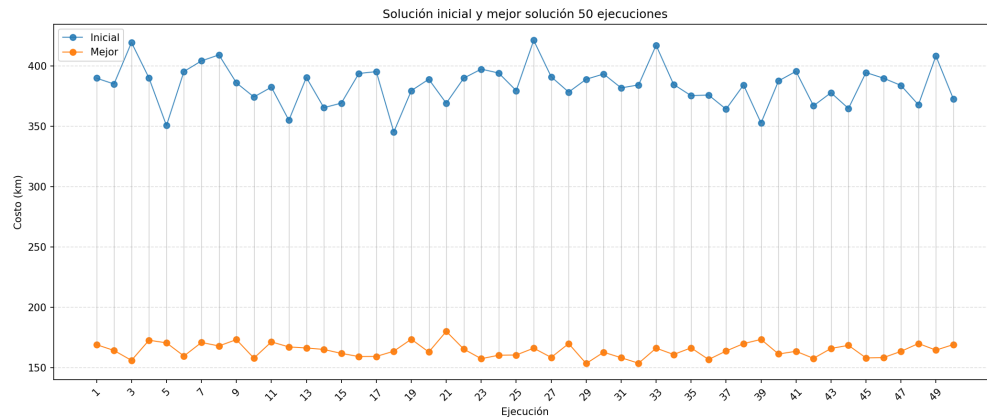


Figura 5: Grafica costo solución inicial y costo mejor solución de 50 ejecuciones

2. **Iteraciones:** Al partir de una solución inicial no completamente aleatoria, y que se puede considerar como buena, no nos es necesario empezar con una temperatura alta, y poder ser mas selectivos en las soluciones, y dedicar esos recursos computacionales en las iteraciones finales para obtener los mejores resultados posibles.

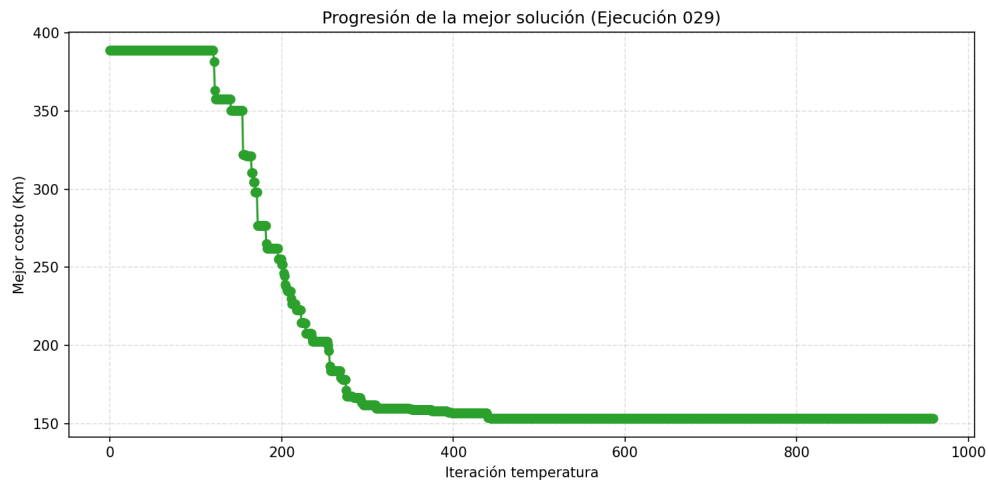


Figura 6: Progresión costo por Iteraciones de temperatura en la mejor solución

8. Agenda de trabajo

Cuadro 2: Cronograma de Actividades del Proyecto.

Fecha	Actividad
2 de Octubre	Análisis del problema: Identificación del tipo de problema de enrutamiento de vehículos, definición de alcance y objetivos del proyecto.
3 de Octubre	Investigación sobre las posibles heurísticas útiles para la resolución del problema, y selección de una de ellas.
4 de Octubre	Análisis sobre entornos para el desarrollo del proyecto y sus respectivas utilidades.
5 de Octubre	Desarrollo de script para obtener una matriz de distancias a partir de las coordenadas de latitud y longitud utilizando la fórmula de Haversine.
6 de Octubre	Desarrollo de código para el mapeo de los puntos en un mapa interactivo y agrupamiento de tiendas en base a distancia.
7 de Octubre	Desarrollo de la solución inicial semi-aleatoria y mapeo de esta misma.
8 de Octubre	Implementación del recocido simulado, así como sus parámetros y criterios de aceptación y parada.
9 de Octubre	Desarrollo de gráficos visuales para las soluciones finales.
10-12 de Octubre	Redacción del documento presente.

9. Conclusiones

El desarrollo de este proyecto muestra una implementación exitosa de un sistema de optimización de rutas para el Problema de Enrutamiento de Vehículos con Múltiples Depósitos (MDVRP), utilizando la metaheurística del Recocido Simulado, donde se demostró la efectividad del algoritmo logrando una mejora media del 57.24% en el costo de las rutas. El modelo aquí propuesto no solo cumplió con el objetivo de encontrar soluciones de bajo costo, si no que también, aplicó estrategias en la solución inicial para de manera eficaz acelerar el proceso hacia soluciones de bajo costo.

Finalmente, este proyecto brinda una base sólida para futuras mejoras, principalmente en la matriz de distancia, en la que el siguiente paso sería utilizar una API para obtener valores del mundo real, así logrando aumentar la precisión y aplicabilidad al mundo real.

Referencias

- [1] Elver A. Bautista and Jaime Hernán Cárdenas. Diseño de un modelo de optimización de rutas de transporte. *El Hombre y la Máquina*, (32):52–67, 2009. URL: <https://www.redalyc.org/articulo.oa?id=47811604005>.
- [2] Colaboradores de Wikipedia. Problema del viajante — Wikipedia, la enciclopedia libre, 2025. URL: https://es.wikipedia.org/wiki/Problema_del_viajante.
- [3] GS1 México. 5 formas de reducir el costo logístico en las empresas. Blog de GS1 México, 1 2021. URL: <https://blog.gs1mexico.org/reducir-costo-logistico-en-las-empresas>.
- [4] Heechul Bae and Ilkyeong Moon. Multi-depot vehicle routing problem with time windows considering delivery and installation vehicles. *Applied Mathematical Modelling*, 40(13-14):6536–6549, 2016. doi:10.1016/j.apm.2016.01.059.
- [5] Wikipedia contributors. Haversine formula — Wikipedia, the free encyclopedia, 2025. URL: https://en.wikipedia.org/wiki/Haversine_formula.
- [6] Purnawan Adi Wicaksono, Addinul Rachman, Sulistyo Wibowo, and Ratna Sari. Comparison of simulated annealing, nearest neighbour, and tabu search methods to solve vehicle routing problems. In *IOP Conference Series: Earth and Environmental Science*, volume 426, page 012138, 2020. doi:10.1088/1755-1315/426/1/012138.
- [7] Lukas Palatinus and Riccardo Spagna. Crystal structure determination. In Tapash Chakraborty, editor, *Encyclopedia of Condensed Matter Physics (Second Edition)*, pages 29–40. Academic Press, 2024. doi:10.1016/B978-0-323-90800-9.00211-0.