

Tecnológico Nacional de México

Optimización de rutas de distribución para tiendas utilizando Recocido Simulado.

Materia:

Tópicos de Inteligencia Artificial

Integrantes:

Dávila Bejarano Víctor Jesús
Flores Medina Martin

Profesor:

Mora Félix Zuriel Dathan

12 de Octubre del 2025

Repositorio de GitHub

<https://github.com/MartinMFM/topicsIA>

Contents

1	Introduction	3
2	Objetivo General	3
3	Objetivos Específicos	3
4	Justificación	4
5	Alcance	4
6	Desarrollo	5
6.1	Descripción del Problema	5
6.2	Selección y Explicación del Algoritmo Heurístico	5
6.3	Implementación Computacional	6
6.3.1	Función Objetivo	7
6.4	Resultados Obtenidos	7
6.5	Análisis de efectividad	9
7	Agenda de trabajo	10
8	Conclusiones	11
	Referencias	11

1 Introduction

En el competitivo entorno empresarial actual, la optimización del transporte se ha consolidado como la columna vertebral de cualquier cadena de suministro eficiente [1]. Las actividades logísticas, y en particular el transporte, juegan un papel vital para satisfacer las necesidades del cliente, ayudando a las empresas a ganar mercados al asegurar que los productos lleguen al lugar correcto, en el momento adecuado y con la calidad requerida [2]. La optimización de estos procesos no solo busca una mayor eficiencia, sino que impacta directamente en la reducción de costos operativos y en el aumento de la satisfacción del cliente, convirtiéndose en una necesidad para responder a las demandas del mercado [3].

El problema organizacional de diseñar un conjunto óptimo de rutas para una flota de vehículos que debe atender a un grupo de clientes geográficamente dispersos se conoce formalmente como el Problema de Enrutamiento de Vehículos (VRP, por sus siglas en inglés) [4]. Este es un problema de optimización combinatoria esencial y de gran importancia práctica en los campos de la logística, el transporte y la distribución [5]. El objetivo es determinar las rutas de menor costo que permitan servir a todos los clientes bajo un conjunto de restricciones específicas.

La complejidad inherente al VRP radica en que está clasificado como un problema NP-difícil (NP-hard) [6]. Esto implica que encontrar la solución óptima global mediante métodos exactos es computacionalmente inviable para la mayoría de los escenarios del mundo real, que suelen ser de gran escala y complejidad. En este contexto, los métodos heurísticos y metaheurísticos adquieren una relevancia fundamental. A diferencia de los métodos exactos, las heurísticas están diseñadas para encontrar soluciones de alta calidad en un tiempo de cómputo razonable, lo que las hace más adecuadas para aplicaciones prácticas [7]. Por lo tanto, para obtener soluciones realistas y muy cercanas al óptimo, el uso de heurísticas es la estrategia más extendida y efectiva tanto en la investigación académica como en la industria [5].

2 Objetivo General

Desarrollar e implementar una metodología computacional, fundamentada en un algoritmo metaheurístico, para la optimización de las rutas de distribución de una cadena de restaurantes, con el fin de minimizar la función de costo total asociada a la operación logística y maximizar su eficiencia.

3 Objetivos Específicos

- Analizar y caracterizar el problema de enrutamiento, identificando sus restricciones operativas.
- Formular un modelo matemático que represente el problema como una variante del VRP.

- Seleccionar y justificar la metaheurística de solución más adecuada para el problema.
- Desarrollar la implementación computacional del algoritmo seleccionado.
- Validar el modelo mediante la simulación de escenarios de distribución realistas.
- Evaluar el desempeño de la solución utilizando Indicadores Clave de Rendimiento (KPIs).
- Documentar la metodología, desarrollo y resultados del proyecto.

4 Justificación

La optimización de rutas de distribución es una necesidad estratégica, ya que el transporte representa uno de los costos logísticos más elevados y es un factor clave en la competitividad y satisfacción del cliente [8, 1]. Resolver el Problema de Enrutamiento de Vehículos (VRP) permite a las empresas minimizar costos operativos y responder de manera más ágil a las demandas del mercado [3]. Para problemas de escala real, que involucran múltiples clientes y vehículos, encontrar la solución óptima global mediante métodos exactos es computacionalmente inviable, ya que el tiempo requerido para ello crece de manera exponencial. En este contexto, las metaheurísticas ofrecen un enfoque pragmático y eficiente, pues están diseñadas para encontrar soluciones de muy alta calidad en un tiempo de cómputo razonable, lo que las hace más adecuadas para aplicaciones prácticas [7]. Estas técnicas son cruciales para evitar quedar atrapado en óptimos locales, un riesgo común en problemas de optimización complejos [9]. La implementación de una solución heurística transforma la toma de decisiones, pasando de un modelo reactivo a un enfoque proactivo y basado en datos. Estas herramientas otorgan visibilidad y control sobre la cadena de suministro, permitiendo a los planificadores simular escenarios y tomar decisiones informadas para la mejora continua de la eficiencia logística [10, 2].

5 Alcance

El presente proyecto se centra en el desarrollo de una solución computacional que optimiza las rutas de distribución mediante el algoritmo de recocido simulado. Para delimitar el alcance, el modelo se enfocará en la ruta de un solo vehículo por centro de distribución, partiendo del supuesto de que este cuenta con la capacidad suficiente para abastecer a todas las sucursales. Asimismo, los cálculos se basarán en distancias geográficas estáticas, sin considerar factores dinámicos como el tráfico o cierres viales. Como resultado final, el sistema mostrará las iteraciones del algoritmo en su proceso hacia la solución óptima y generará una visualización de la ruta en un mapa.

6 Desarrollo

6.1 Descripción del Problema

El problema central consiste en optimizar las rutas de distribución desde múltiples centros de distribución hacia las diversas sucursales de una cadena de restaurantes, con el objetivo principal de minimizar el costo total de la operación. Este escenario se enmarca dentro de la categoría de Problemas de Enrutamiento de Vehículos (VRP), y más específicamente, como una variante del Problema de Enrutamiento de Vehículos Multidepósito (MDVRP), ya que las rutas se originan desde más de un punto de partida [7]. El objetivo es diseñar un conjunto de rutas de costo mínimo, donde cada ruta parte de uno de los centros de distribución, atiende a un subconjunto de sucursales dentro de su zona, garantizando que cada sucursal sea visitada exactamente una vez por un único vehículo [8].

6.2 Selección y Explicación del Algoritmo Heurístico

Para este proyecto, se ha seleccionado el algoritmo de Recocido Simulado, una técnica de búsqueda local que ha sido probada con éxito en numerosos problemas de optimización combinatoria, incluyendo el VRP y sus variantes [11, 12]. La principal justificación para la elección del Recocido Simulado es su **probada habilidad para escapar de óptimos locales** [13] y su facilidad de implementación, flexibilidad y robustez lo han consolidado como una técnica muy popular en la investigación de operaciones [14]. La justificación de esta elección se basa en el excelente equilibrio que ofrece entre **eficacia, robustez y simplicidad de implementación** [14]. Aunque tanto el Recocido Simulado como la Búsqueda Tabú poseen mecanismos para escapar de óptimos locales, lo hacen a través de filosofías distintas. La Búsqueda Tabú emplea una memoria adaptativa (la lista tabú) para prohibir movimientos recientes, lo que requiere una gestión más compleja de sus estructuras y parámetros [9]. Por el contrario, el Recocido Simulado logra una exploración robusta del espacio de soluciones mediante un mecanismo probabilístico más sencillo y elegante, lo que lo convierte en una opción ideal para este proyecto, ya que permite implementar una solución potente y flexible sin la sobrecarga que implicaría el manejo de las estructuras de memoria de corto y largo plazo de la Búsqueda Tabú.

El Recocido Simulado se inspira en el proceso metalúrgico de recocido, donde un material se calienta a alta temperatura y luego se enfria lentamente para que sus átomos se reorganicen en una estructura de mínima energía y máxima resistencia [13]. En la optimización, el "estado" del sistema es una solución candidata (una ruta), y la "energía" es el valor de la función objetivo (el costo total de la ruta). El algoritmo sigue una estructura iterativa:

1. **Generación de Solución Inicial:** Se parte de una solución inicial S , que puede ser generada de forma aleatoria o mediante una heurística simple.

2. **Definición de Vecindad:** En cada iteración, se genera una solución "vecina" S' aplicando una pequeña modificación a S , como intercambiar la posición de dos clientes en la ruta (movimiento de tipo *swap*).
3. **Criterio de Aceptación de Boltzmann:** Se calcula la diferencia de costo $\Delta E = \text{costo}(S') - \text{costo}(S)$.
 - Si $\Delta E \leq 0$, la nueva solución es mejor, por lo que se acepta siempre ($S \leftarrow S'$).
 - Si $\Delta E > 0$, la nueva solución es peor. Sin embargo, puede ser aceptada con una probabilidad dada por la **función de probabilidad de Boltzmann**: $P(\text{aceptar}) = e^{-\Delta E/T}$, donde T es la "temperatura" actual [14].
4. **Esquema de Enfriamiento:** La temperatura T se reduce gradualmente en cada iteración siguiendo una **secuencia de enfriamiento geométrica o exponencial**, donde la temperatura se actualiza mediante la fórmula $T_{k+1} = \alpha \cdot T_k$, con un factor $\alpha < 1$. A temperaturas altas, el algoritmo explora ampliamente el espacio de soluciones aceptando casi cualquier movimiento, mientras que a temperaturas bajas, se vuelve más selectivo y converge hacia una solución de alta calidad.

6.3 Implementación Computacional

La solución se desarrolló en el lenguaje de programación **Python**, seleccionado por su versatilidad y su robusto ecosistema de librerías para el cómputo científico y la visualización de datos. La implementación se estructuró de manera modular para separar la configuración, la carga de datos y la lógica de optimización.

La configuración del sistema se centralizó en un archivo `config.py`, que permite ajustar fácilmente los parámetros clave del algoritmo de recocido simulado, como la temperatura inicial (`TEMPERATURA_INICIAL = 5000`), la tasa de enfriamiento (`TASA_ENFRIAMIENTO = 0.995`) y las condiciones de término. Este enfoque facilita la experimentación y el ajuste fino del comportamiento del algoritmo.

El núcleo de la lógica se encuentra en la clase `RouteOptimizer`, la cual gestiona el proceso de optimización completo. Esta clase:

1. Carga los datos de los centros de distribución y las tiendas desde archivos externos.
2. Invoca al algoritmo de Recocido Simulado para cada zona de distribución de forma independiente.
3. Almacena y consolida los resultados, calculando el costo total optimizado como la suma de los costos de cada ruta individual.

El proceso de optimización se ejecuta por zonas, tratando cada centro de distribución y sus tiendas asignadas como un subproblema independiente. Para

cada zona, se invoca a la clase `SimulatedAnnealing`, que es la responsable de ejecutar el algoritmo para encontrar la ruta de menor costo.

6.3.1 Función Objetivo

La función objetivo es la métrica que el algoritmo busca minimizar. Para este problema, se define como el **costo total de la ruta**. Es importante destacar que este costo no se basa únicamente en la distancia, sino que es una función compuesta. A partir de los archivos de configuración del sistema, se especifica el uso de dos matrices de entrada: una de distancias (`matriz_distancias.xlsx`) y otra de costos de combustible (`matriz_costos_combustible.xlsx`).

Por lo tanto, la función de costo para cualquier traslado entre dos puntos en una ruta es una combinación ponderada de la distancia geográfica y el gasto de combustible asociado. El costo total del sistema es la suma de los costos optimizados de todas las rutas generadas para cada zona.

6.4 Resultados Obtenidos

La ejecución del algoritmo de optimización generó un conjunto de rutas eficientes para cada uno de los 10 centros de distribución. El costo total combinado de la operación logística, tras aplicar el proceso de recocido simulado, se redujo a un valor final de **335.95**.

Es relevante señalar que, dado que el costo es una función combinada de distancia y combustible, el valor "335.95" representa **unidades de costo monetario**. Si, por ejemplo, los datos de la matriz de combustible estuvieran expresados en Pesos Mexicanos (MXN), el resultado final estaría en esa misma moneda. Para efectos de este informe, se interpreta como una unidad monetaria abstracta.

Una representación visual de estas rutas optimizadas se puede observar en la **Figura 1**. Los resultados detallados para cada zona se presentan en la **Tabla 1**, donde se resume el número de tiendas atendidas, la capacidad total de venta gestionada y el costo optimizado correspondiente.

Zona	N.º de Tiendas	Capacidad Total	Costo Optimizado
Centro de Distribución 1	1	17,757	2.16
Centro de Distribución 2	18	232,230	62.11
Centro de Distribución 3	5	62,614	25.98
Centro de Distribución 4	7	83,196	27.34
Centro de Distribución 5	10	138,627	41.12
Centro de Distribución 6	2	31,260	7.87
Centro de Distribución 7	14	169,321	52.16
Centro de Distribución 8	9	102,795	34.22
Centro de Distribución 9	13	176,976	43.99
Centro de Distribución 10	11	133,181	39.00
TOTAL	90	1,147,957	335.95

Table 1: Resultados de la optimización por zona de distribución.

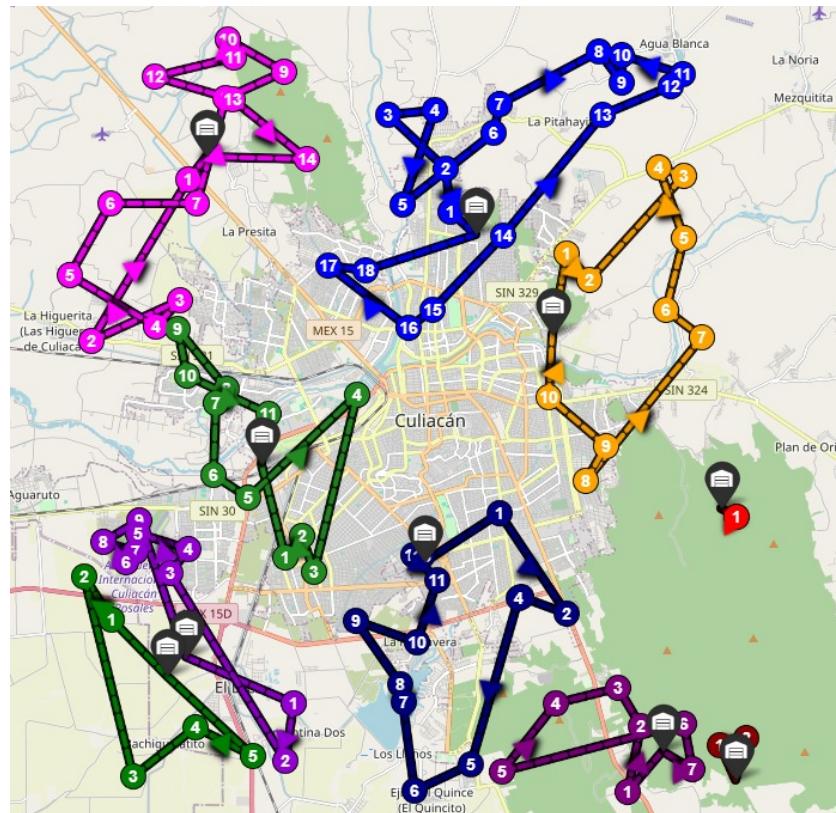


Figure 1: Visualización de las rutas optimizadas para los 10 centros de distribución, con un costo total de 335.95 unidades monetarias.

A modo de ejemplo, la ruta optimizada para la zona más compleja, el "Centro de Distribución 2", que atiende a 18 tiendas, fue la siguiente:

- **Ruta Zona 2:** Centro de Distribución 2 → Tienda 37 → Tienda 84 → Tienda 35 → Tienda 50 → Tienda 38 → Tienda 31 → Tienda 87 → Tienda 16 → Tienda 47 → Tienda 8 → Tienda 17 → Tienda 82 → Tienda 90 → Tienda 34 → Tienda 72 → Tienda 64 → Tienda 79 → Tienda 10 → Centro de Distribución 2.

6.5 Análisis de efectividad

El modelo implementado demostró ser altamente efectivo para abordar el problema de enrutamiento planteado. La efectividad se puede analizar desde tres perspectivas clave:

1. **Calidad de la Solución:** El algoritmo convergió exitosamente hacia una solución de bajo costo para el sistema completo, alcanzando un costo total de 335.95. La capacidad del Recocido Simulado para evitar óptimos locales fue fundamental para explorar el vasto espacio de soluciones y encontrar rutas eficientes, incluso en zonas con un número elevado de tiendas como la Zona 2 (18 tiendas) y la Zona 7 (14 tiendas).
2. **Escalabilidad y Eficiencia Computacional:** La estrategia de dividir el problema general en subproblemas por zona fue crucial para la efectividad del modelo. Este enfoque permitió resolver cada caso de manera independiente, haciendo que el problema fuera computacionalmente manejable. El sistema optimizó rutas para un total de 90 tiendas distribuidas en 10 zonas distintas, demostrando que la metodología es escalable y aplicable a operaciones logísticas de tamaño considerable.
3. **Robustez y Flexibilidad:** Gracias a la parametrización en el archivo `config.py`, el modelo es robusto y adaptable. Los parámetros seleccionados, como una temperatura inicial de 5000 y una tasa de enfriamiento de 0.995, permitieron un equilibrio adecuado entre la exploración (aceptando soluciones peores al inicio) y la explotación (convergiendo hacia una solución de alta calidad al final). Esto asegura que el algoritmo no se estanke prematuramente en una solución subóptima y pueda ser ajustado para resolver problemas con diferentes características.

7 Agenda de trabajo

Fecha	Actividad Realizada
Viernes, 3 de Octubre	Investigación y Definición del Problema: Análisis inicial del Problema de Enrutamiento de Vehículos y recopilación de bibliografía. Definición de los objetivos generales y específicos.
Sábado, 4 de Octubre	Estructuración del Proyecto: Redacción de la introducción, justificación y alcance del proyecto. Estructuración inicial del documento.
Domingo, 5 de Octubre	Análisis y Selección del Algoritmo: Investigación comparativa de metaheurísticas (Recocido Simulado vs. Búsqueda Tabú). Selección y justificación detallada del Recocido Simulado.
Lunes, 6 de Octubre	Desarrollo - Fase 1: Implementación de las funciones base en Python: cálculo de distancias (Haversine), función de costo y generación de soluciones iniciales aleatorias.
Martes 7 y Miércoles 8 de Octubre	Desarrollo - Fase 2: Implementación del núcleo del algoritmo de Recocido Simulado, incluyendo el criterio de aceptación de Boltzmann y el esquema de enfriamiento.
Jueves, 9 de Octubre	Ejecución de Experimentos: Ejecución del algoritmo con los parámetros finales definidos (T_0, α, L, T_f). Generación y almacenamiento de los resultados (rutas optimizadas y costos).
Viernes, 10 de Octubre	Ánalysis y Visualización de Resultados: Análisis de la efectividad y la mejora obtenida. Desarrollo de los scripts para generar los mapas de las rutas optimizadas.
Sábado, 11 de Octubre	Consolidación y Documentación Final: Redacción de las conclusiones, revisión final del documento, formato de referencias y compilación de la versión final del proyecto.

Table 2: Cronograma de actividades del proyecto.

8 Conclusiones

Este proyecto representó una inmersión práctica en el complejo mundo de la optimización logística, culminando con una herramienta funcional que demuestra la viabilidad del Recocido Simulado para resolver problemas de enrutamiento de rutas a una escala considerable. El resultado más tangible fue ver cómo el algoritmo transformaba sistemáticamente rutas costosas en trayectorias optimizadas, validando el impacto directo que estas técnicas pueden tener en la eficiencia operativa. Aprendimos a utilizar el algoritmo de recocido simulado en un ejercicio más realista y comprendimos la verdadera magnitud de un problema de optimización combinatoria. Además, a separar el problema en partes más pequeñas por ejemplo el de separar por zonas. Este proyecto trabajo sienta las bases para una herramienta mucho más robusta. El siguiente paso lógico sería enriquecer el modelo con las restricciones del mundo real que deliberadamente dejamos fuera del alcance: la capacidad de los vehículos, las ventanas horarias de entrega y la gestión de una flota completa. La integración con una API de mapas para usar datos de tráfico en tiempo real transformaría el sistema de un modelo académico a una solución logística dinámica y potente.

References

- [1] Sophus AI. *The Benefits of Transportation Optimization in Supply Chain Design*. <https://sophus.ai/benefits-of-transportation-optimization-in-supply-chain-design/>. Consultado: 2024-10-04.
- [2] Leonora Bottani and Riccardo Accorsi. “Editorial for Special Issue “Logistics and Supply Chain Process Optimization””. In: *Processes* 12.6 (2024), p. 1134. DOI: 10.3390/pr12061134.
- [3] Matus Vojtek et al. “Optimisation in Transport and Logistic Processes”. In: *10th International Conference on Transport Systems Telematics*. Vol. 44. Transportation Research Procedia. 2020, pp. 251–258. DOI: 10.1016/j.trpro.2020.02.033.
- [4] Ilhamdaniah Ilhamdaniah, Ilyas Masudin, and Dian Palupi Restuputri. “A review on vehicle routing problem for perishable goods”. In: *Cogent Engineering* 7 (1 2020), p. 1816148. DOI: 10.1080/23311916.2020.1816148.
- [5] Wen-Jing Li, Jie-Sheng Liu, and Wen-Long Zhang. “Heuristics for Vehicle Routing Problem: A Survey and Recent Advances”. In: *arXiv preprint arXiv:2303.04147* (2023).
- [6] S. Priyadarshini and T. T. Mirnalinee. “A Review on Vehicle Routing Problem and its Variants”. In: *Journal of Emerging Technologies and Innovative Research (JETIR)* 6 (2 2019), pp. 706–711.

- [7] Kris Braekers, Katrien Ramaekers, and An Caris. “The vehicle routing problem: State of the art classification and review”. In: *Computers & Industrial Engineering* 99 (2016), pp. 300–313. doi: 10.1016/j.cie.2015.12.007.
- [8] Elver A. Bermeo Muñoz et al. “Diseño de un modelo de optimización de rutas de transporte”. In: *Estudios Gerenciales* 25.111 (2009), pp. 111–136.
- [9] Fred Glover. “Future Paths for Integer Programming and Links to Artificial Intelligence”. In: *Computers & Operations Research* 13.5 (1986), pp. 533–549. doi: 10.1016/0305-0548(86)90048-1.
- [10] Inbound Logistics. *Logistics Optimization: Importance, Process, and Optimization*. <https://www.inboundlogistics.com/articles/logistics-optimization/>. Consultado: 2024-10-04.
- [11] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. “Optimization by Simulated Annealing”. In: *Science* 220.4598 (1983), pp. 671–680.
- [12] Ibrahim H. Osman. “Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem”. In: *Annals of Operations Research* 41 (1993), pp. 421–451.
- [13] Jair J. de la Cruz H. et al. “Análisis comparativo de las aproximaciones heurísticas Ant-Q, recocido simulado y búsqueda tabú en la solución del problema del agente viajero”. In: *Ingeniería y Desarrollo* 14 (2003), pp. 141–157.
- [14] C. García-Martínez et al. “Búsquedas locales y metaheurísticas”. In: *La Inteligencia Artificial y sus Aplicaciones*. Ed. by J. M. Corchado, E. Corchado, and S. Omatu. Ediciones Universidad de Salamanca, 2008, pp. 111–140.