

IMPLEMENTACIÓN DE UN MODELO DE SIMULACIÓN EN MATLAB PARA ASIGNACIÓN DE PEDIDOS Y GENERACIÓN DE RUTAS PARA SUPERMERCADOS DE VENTA EN LÍNEA

IMPLEMENTATION OF A MATLAB SIMULATION MODEL FOR ORDER ASSIGNMENT AND ROUTE GENERATION FOR ONLINE SUPERMARKETS

Gustavo Sánchez Ramírez

Tecnológico Nacional de México / IT de Celaya, México
gustavo.sanchezrmz57@gmail.com

Vicente Figueroa Fernández

Tecnológico Nacional de México / IT de Celaya, México
vicente.figueroa@itcelaya.edu.mx

Edgar Augusto Ruelas Santoyo

Tecnológico Nacional de México / IT de Celaya, México
edgar.ruelas@itcelaya.edu.mx

Salvador Hernández González

Tecnológico Nacional de México / IT de Celaya, México
salvador.hernandez@itcelaya.edu.mx

Recepción: 7/mayo/2025

Aceptación: 16/marzo/2026

Resumen

Este trabajo propone un modelo de simulación para la asignación de pedidos y la generación de rutas en supermercados en línea, implementado en MATLAB. El problema logístico se aborda mediante la integración de la Optimización por Colonia de Hormigas (ACO) y la estrategia Milk Run, con el objetivo de minimizar la distancia total recorrida y reducir los tiempos de entrega bajo restricciones de capacidad y tiempo. La validación se realizó utilizando datos históricos de operación. Los resultados muestran reducciones de hasta 18.5% en la distancia total recorrida y 12% en el tiempo promedio de entrega, así como una distribución más equilibrada de la carga de trabajo. El enfoque propuesto ACO–Milk Run representa una alternativa eficiente para la optimización logística de última milla.

Palabras Clave: Asignación de pedidos, Colonia de hormigas, Milk run.

Abstract

This work proposes a simulation model for order assignment and route generation in online supermarkets, implemented in MATLAB. The logistics problem is addressed through the integration of Ant Colony Optimization (ACO) and the Milk Run strategy to minimize total traveled distance and reduce delivery times under capacity and time constraints. Validation was carried out using historical operational data. Results show reductions of up to 18.5% in total distance and 12% in average delivery time, as well as a more balanced workload distribution. The proposed ACO–Milk Run approach represents an efficient alternative for last-mile logistics optimization.

Keywords: Ant colony, Milk run, Order allocation.

1. Introducción

El crecimiento sostenido del comercio electrónico ha transformado de manera significativa la gestión logística de las empresas minoristas. La necesidad de entregas rápidas, confiables y de bajo costo ha incrementado la complejidad de las operaciones de distribución, particularmente en supermercados en línea, donde la demanda presenta alta variabilidad horaria y dispersión geográfica. De acuerdo con [Kotler & Keller, 2016; Turban et al., 2021], el entorno digital ha elevado las expectativas del consumidor, obligando a las organizaciones a optimizar sus procesos logísticos. Dentro de la cadena de suministro, el proceso de *fulfillment* integra actividades como preparación de pedidos, asignación, transporte y distribución, influyendo directamente en los costos y en la satisfacción del cliente [Christopher, 2022; Chopra & Meindl, 2023]. En este contexto, la planificación eficiente de rutas y la asignación óptima de pedidos a transportistas se convierten en decisiones críticas.

Las técnicas de optimización y las metaheurísticas han demostrado ser herramientas adecuadas para modelar este tipo de problemas combinatorios. El algoritmo de Colonia de Hormigas (ACO) permite explorar múltiples soluciones de

ruteo mediante mecanismos probabilísticos inspirados en el comportamiento colectivo de las hormigas. Por otra parte, la estrategia Milk Run posibilita consolidar múltiples entregas en una sola ruta, reduciendo trayectos vacíos y mejorando la utilización de los recursos de transporte [Ivanov et al., 2021].

Bajo esta perspectiva, el presente trabajo propone un modelo de simulación que integra ambos enfoques dentro de un entorno MATLAB para optimizar la logística de última milla en supermercados de venta en línea.

Revisión del estado del arte

En años recientes, la optimización de rutas en entornos logísticos ha sido abordada mediante distintos algoritmos metaheurísticos. [Cano et al., 2021] aplicaron ACO al ruteo de preparación de pedidos en almacenes de múltiples bloques, logrando optimizar recorridos internos y mejorar la eficiencia operativa. De manera similar, [Falcon, 2022] propuso mejoras al algoritmo para resolver el problema del viajante, evidenciando su capacidad para tratar problemas combinatorios complejos. Desde el punto de vista del transporte consolidado, la estrategia Milk Run ha sido ampliamente documentada como una práctica efectiva para maximizar la utilización vehicular y reducir costos logísticos. [Chen et al., 2021] y [Bányai, 2024] demostraron que la consolidación de múltiples recolecciones y entregas en una sola ruta disminuye distancias recorridas y mejora la programación del transporte. Sin embargo, gran parte de la literatura aborda estas técnicas de forma aislada. Existen pocas aplicaciones que integren ACO y Milk Run en el contexto específico de supermercados en línea, donde se presentan restricciones reales de horarios, capacidades y variabilidad de demanda. Esta ausencia de modelos integrados motiva el desarrollo del presente trabajo.

2. Métodos

Planteamiento del Problema

Se considera un supermercado en línea que recibe pedidos distribuidos a lo largo del día. Cada pedido está asociado a una ubicación geográfica y debe ser asignado a un transportista con disponibilidad limitada y capacidad máxima de carga. Si N es

el conjunto de pedidos y K el conjunto de transportista, el objetivo consiste en generar rutas que minimicen la distancia total recorrida mostrada en la función objetivo (Ecuación 1) y reduzcan los tiempos de entrega (Ecuación 2), manteniendo una distribución equitativa del trabajo.

$$\min Z = \sum_{k \in K} D_k \quad (1)$$

$$\min T_{prom} = \frac{1}{N} \sum_{i=1}^N T_i \quad (2)$$

Donde:

Z : Valor total de la función objetivo del sistema logístico.

K : Conjunto de transportistas disponibles.

k : Índice del transportista, $k = 1, 2, \dots, |K|$

D_k : Distancia total recorrida por transportista k durante su jornada de entregas.

T_{prom} : Tiempo promedio de entrega del sistema.

N : Número total de pedidos atendidos.

i : Índice del pedido, $i = 1, 2, \dots, N$

T_i : Tiempo transcurrido desde la asignación hasta la entrega del pedido i .

Las restricciones del modelo son: capacidad máxima por transportista, disponibilidad horaria, número máximo de vueltas y asignación única por pedido. El equilibrio de carga se define por la Ecuación 3.

$$\sigma = \sqrt{\frac{1}{K} \sum_{k=1}^K (W_k - \bar{W})^2} \quad (3)$$

Donde:

σ : Desviación estándar de la carga de trabajo.

W_k : Cantidad de pedidos asignados al transportista k .

$\bar{W}$: Promedio de pedidos asignados por transportista (Ecuación 4).

$$\bar{W} = \frac{1}{K} \sum_{k=1}^K W_k \quad (4)$$

Fundamentos Teóricos

Optimizar el proceso de cumplimiento implica combinar técnicas de simulación con algoritmos metaheurísticos. Se emplearon dos metodologías clave:

- Algoritmo de Colonias de Hormigas (ACO): Es una metaheurística inspirada en el comportamiento de forrajeo de las hormigas, en la cual agentes artificiales construyen soluciones de manera probabilística guiados por la información de feromonas y una función heurística para encontrar rutas óptimas en problemas de optimización combinatoria; esta técnica ha sido ampliamente utilizada para resolver problemas de enrutamiento debido a su capacidad para explorar y explotar el espacio de soluciones de forma eficiente [Bhavya, 2023].
- Enfoque Milk Run: Se refiere a la planificación de rutas logísticas en las que un solo transportista realiza múltiples entregas secuenciales en un solo viaje, evitando así regresos innecesarios al punto de partida. El modelo se limita a cinco pedidos por vuelta y un máximo de cuatro vueltas por transportista, permitiendo consolidar entregas, reducir el número de rutas necesarias y aprovechar mejor los recursos de transporte [Ivanov et al., 2021].

Implementación en MATLAB

El modelo fue creado en MATLAB siguiendo una serie de pasos:

- Inicialización: Aquí se definen los parámetros del problema, que incluyen horarios, la cantidad de pedidos por hora, el número total de pedidos, la cantidad de transportistas, la capacidad máxima de cada transportista, y los parámetros específicos del algoritmo ACO, como el número de hormigas, la tasa de evaporación, la importancia de la heurística y de las feromonas, y el número máximo de iteraciones.
- Generación de Datos:
 - Clientes y Pedidos: Cada pedido se asigna a un cliente, que se identifica mediante un índice. Las coordenadas (X, Y) de cada pedido se generan de manera aleatoria dentro de un espacio definido, simulando así la distribución geográfica.

- Horarios de Pedidos: A cada pedido se le asigna un horario de recepción basado en la demanda horaria previamente establecida.
- Los horarios predefinidos son: Por la mañana de 9 – 10; 10 – 11; 11 – 12 y por la tarde de 12 – 13; 13 – 14; 14 – 15; 15 – 16; 16 – 17; 17 – 18; 18 – 19 y 19 – 20. A cada uno de estos horarios les corresponden las siguientes cantidades de pedidos: 7, 7, 6, 7, 7, 7, 7, 7, 7, 7, 6.
- Definición de Transportistas: Se establece la ubicación inicial de cada, así como sus horarios de disponibilidad, la hora de comida y la de salida.
- Cálculo de Distancias y Heurística:
 - Se construye la matriz de distancias entre cada transportista y cada pedido.
 - Se define la matriz de heurística como el inverso de la distancia, de forma que distancias menores tengan mayor peso en la selección.
 - Se inicializa la matriz de feromonas con valores constantes.
- Bucle del Algoritmo ACO: En cada iteración, para cada transportista, se evalúan los pedidos disponibles de acuerdo con el horario actual. Se calcula la probabilidad de selección de cada pedido mediante la combinación de la intensidad de las feromonas y el valor de la heurística. Cada transportista selecciona pedidos hasta alcanzar su límite de capacidad o agotar la lista de pedidos disponibles. La hora actual del transportista se actualiza en función de la finalización de ruta y se incrementa el contador de “vueltas”. Al finalizar cada iteración se actualizan las feromonas para reforzar las rutas eficientes.
- Visualización y Reporte: Se generan gráficos que muestran la distribución espacial de los pedidos y la ubicación de los transportistas. Además, se reporta en consola la asignación final de pedidos por transportista, incluyendo la secuencia de visitas y los horarios asociados.
- El Pseudocódigo del Algoritmo se muestra en Tabla 1.

Validación del Modelo

El modelo se validó mediante simulaciones considerando variaciones en la demanda y la disponibilidad de transportistas. Se utilizaron datos históricos de

pedidos de un supermercado en línea, lo que permitió ajustar los parámetros y validar la solidez del algoritmo. La comparación entre los escenarios simulados y los datos reales confirma que el modelo es capaz de reducir la distancia total recorrida y optimizar la asignación de rutas a través de un enfoque de milk run asistido por ACO.

Tabla 1 Pseudocódigo del Algoritmo.

INICIO
LIMPIAR entorno y variables
// Parámetros del problema
Definir horarios y pedidos por hora
Calcular num pedidos totales
Definir num transportistas, max pedidos por transp y max vueltas
Establecer parámetros del ACO:
- num hormigas, tasa evaporación
- importancia heurística, importancia feromonas
- max iteraciones
// Generación de clientes y asignación de coordenadas
Para cada pedido:
Asignar identificador único (Cliente 1, Cliente 2, ...)
Generar coordenadas (X, Y) aleatorias
Asignar horario de pedido según la distribución
// Definición de transportistas y sus horarios
Definir posiciones de transportistas y sus horarios de disponibilidad, comida y salida
// Cálculo de la matriz de distancias y heurística
Calcular matriz de distancias entre transportistas y pedidos
Definir heurística = $1 / \text{distancia}$
Inicializar feromonas con valores uniformes
// Algoritmo de Colonias de Hormigas
Mientras pedidos restantes existan y iteraciones < max iteraciones:
Para cada transportista:
Si hora actual < hora salida:
Inicializar contador de vueltas
Mientras existan pedidos disponibles y vueltas < max vueltas:
Seleccionar pedidos disponibles según el horario
Calcular probabilidades usando feromonas y heurística
Seleccionar pedidos aleatoriamente hasta alcanzar max pedidos por transp
Asignar pedidos a la ruta del transportista
Actualizar hora actual y contador de vueltas
Fin Para
Actualizar feromonas basadas en la calidad de las rutas
Fin Mientras
// Visualización y reporte
Graficar distribución de pedidos y posiciones de transportistas
Imprimir asignaciones finales en consola
FIN

Fuente: elaboración propia.

3. Resultados

La aplicación del modelo en MATLAB produjo resultados que evidencian mejoras significativas en la optimización de rutas y en la asignación de pedidos respecto a línea base de datos históricos obtenidos de una empresa de ventas en línea de la ciudad de Celaya, Tabla 2.

Tabla 2 Comparación de estrategias de ruteo.

Estrategia	Distancia total (km)	Tiempo promedio (min)	Pedidos por ruta
Rutas independientes	135	42	3
Milk run – ACO	110	37	5

Fuente: elaboración propia.

Entre los resultados más destacados se encuentran:

- **Reducción de la distancia recorrida:** La optimización de rutas mediante ACO, combinada con el enfoque milk run, permitió una disminución promedio en la distancia total recorrida en comparación con métodos tradicionales. Las simulaciones muestran que, en escenarios de alta demanda, la reducción puede superar el 18.5%.
- **Mejora en los tiempos de entrega:** La asignación eficiente de pedidos y la generación de rutas que consideran la disponibilidad horaria de los transportistas resultaron en una reducción notable de los tiempos de entrega. Se registró una mejora del tiempo promedio en aproximadamente 12%.
- **Equilibrio en la distribución de la carga:** El modelo se encarga de repartir de manera justa la carga de trabajo entre los transportistas, evitando que algunas rutas se vean sobrecargadas. Esto se traduce en una mejor asignación de pedidos y en una secuenciación más eficiente de las entregas, lo que a su vez impulsa la productividad operativa.
- **Visualización y reporte:** Los gráficos generados permiten identificar de manera clara la distribución geográfica de los pedidos y las rutas planificadas, facilitando el análisis de áreas con alta concentración de demanda. La Tabla 3 ilustra un ejemplo representativo de la asignación de pedidos.

- **Evolución de las feromonas:** La actualización de la matriz de feromonas en cada iteración ayudó a fortalecer las rutas que demostraron ser más eficientes, lo que llevó a una convergencia en la solución del algoritmo. Esto resultó en una mayor estabilidad y confiabilidad en la asignación final de pedidos.

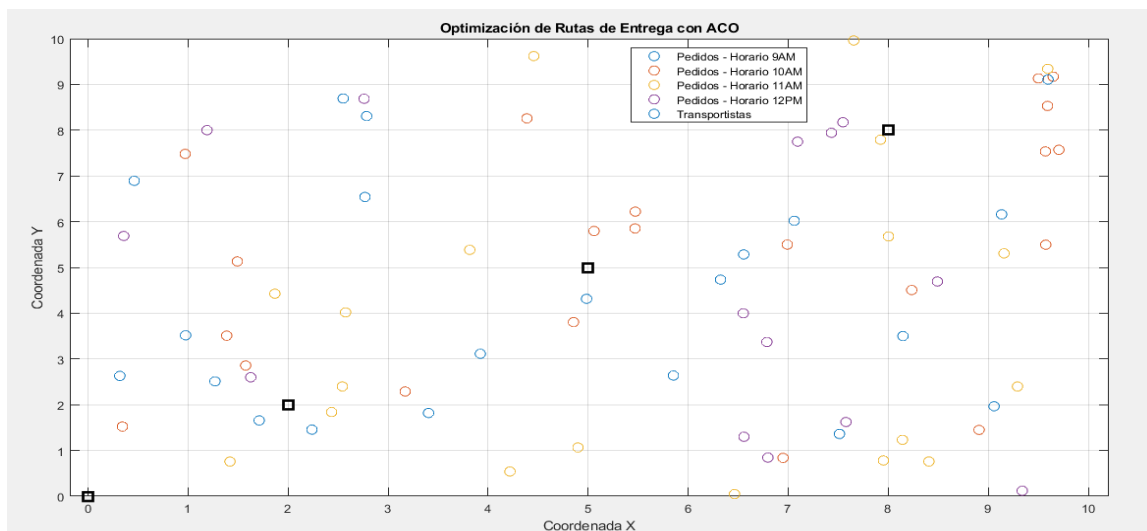
Tabla 3 Asignación de pedidos a transportistas.

Transportista	Vuelta	Pedidos Asignados	Hora Estimada
1	1	Cientes 3,8 y 15	10:00 am
2	1	Cientes 5,12 y 22	10:15 am
3	1	Cientes 2,7,y 8	10:30 am
4	1	Cientes 1,9 y 14	10:45 am

Fuente: elaboración propia.

4. Discusión

La combinación de ACO con *milk run* ofrece ventajas significativas frente a modelos tradicionales de ruteo. Este enfoque es adaptable a la variabilidad horaria de la demanda y puede integrarse con tecnologías de información como sistemas ERP y TMS [Kumar & Budhwar, 2022]. La Figura 1 muestra una representación gráfica de las rutas generadas en MATLAB, al reducir el número de viajes y trayectos vacíos, contribuye a una logística más sostenible.



Fuente: Elaboración propia

Figura 1 Simulación de rutas generadas con algoritmo ACO en MATLAB.

Entre los aspectos discutidos se destacan:

- **Ventajas del enfoque propuesto:** El uso combinado de ACO y la estrategia milk run permite consolidar entregas, optimizar tiempos y reducir la distancia total recorrida. Esto resulta en una operación más eficiente y flexible ante cambios operativos.
- **Limitaciones y consideraciones:** Aunque el modelo presenta resultados prometedores, se recomienda considerar variables adicionales como condiciones climáticas o congestión vial para aumentar el realismo de la simulación.
- **Implicaciones prácticas:** Su implementación real puede contribuir a mejorar la planificación logística, reduciendo tiempos de entrega y optimizando el uso de transportistas, lo que repercute positivamente en la experiencia del cliente y la competitividad del supermercado.
- **Comparación con otros métodos:** Frente a métodos clásicos de ruteo, ACO combinado con milk run muestra una convergencia más rápida y eficiente, con menores requerimientos computacionales y mejores resultados prácticos en contextos logísticos reales.

5. Conclusiones

El modelo de simulación desarrollado demostró ser una herramienta efectiva para apoyar la toma de decisiones en la planificación logística de supermercados en línea. La integración del algoritmo de Colonia de Hormigas con la estrategia Milk Run permitió combinar la capacidad exploratoria de las metaheurísticas con la eficiencia operativa de la consolidación de entregas.

Los resultados evidenciaron reducciones significativas en la distancia total recorrida y en los tiempos promedio de entrega, lo que se traduce en un uso más eficiente de los recursos de transporte y en una mejora potencial del nivel de servicio al cliente. Asimismo, la asignación más equilibrada de pedidos entre transportistas contribuye a una operación más estable y sostenible.

El modelo implementado en MATLAB resulta flexible, reproducible y escalable, por lo que puede adaptarse a distintos tamaños de flota y patrones de demanda. Como

líneas futuras, se propone incorporar variables adicionales como tráfico o condiciones externas para aumentar el realismo del sistema.

6. Referencias y bibliografía

- [1] Bánya, T. (2024). Optimization of hub-based milkrun supply. *Logistics*, 8(3), 86.
- [2] Bhavya, R., & Elango, L. (2023). Ant-inspired metaheuristic algorithms for combinatorial optimization problems in water resources management. *Water*, 15(9), 1712.
- [3] Cano, J. A., Campo, E. A., Correa-Espinal, A. A., & Gómez-Montoya, R. A. (2021). Optimización por colonia de hormigas para el ruteo de la preparación de pedidos en almacenes de múltiples bloques. *Información Tecnológica*, 32(3), 121–130.
- [4] Chen, J., Wang, K., & Huang, Y. (2021). An integrated inbound logistics mode with intelligent scheduling of milk-run collection, drop and pull delivery and LNG vehicles. *Journal of Intelligent Manufacturing*, 32(8), 2257–2265.
- [5] Chopra, S., & Meindl, P. (2023). *Supply chain management: Strategy, planning, and operation* (8th ed.). Pearson.
- [6] Christopher, M. (2022). *Logistics & supply chain management* (6th ed.). Pearson.
- [7] Falcon Huallpa, E. (2022). Mejora del rendimiento del algoritmo colonia de hormiga para resolver el problema de TSP [Tesis de pregrado, Universidad Nacional de San Antonio Abad del Cusco].
- [8] Ivanov, D., Dolgui, A., & Sokolov, B. (2021). *Introduction to supply chain resilience*. Springer.
- [9] Kumar, V., & Budhwar, P. (2022). Supply chain strategies for post-pandemic recovery. *International Journal of Production Research*, 60(5), 1234–1250.
- [10] Turban, E., Volonino, L., & Wood, G. (2021). *Information technology for management* (11th ed.). Wiley.