



Universidad de los Andes

INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

PROYECTO C ETAPA III: IMPLEMENTACIÓN DE
METAHEURÍSTICAS PARA LA OPTIMIZACIÓN EN LA
PLANIFICACIÓN DE RUTAS NACIONALES Y
ESTRATEGIA DE RECARGA PARA LOGÍSTICO

ISIS 3302 - Modelado, simulación y optimización

Autores:

Nicolas Lara Gomez — 202012455

Joan Sebastian Amaya Bohorquez — 202011318

David Mateo Barbosa Monsalve — 202110756

Profesor:

Germán Montoya

Bogotá, Colombia
Mayo 2025

Índice

1. Implementación de Metaheurísticas	1
1.1. Diseño general del Algoritmo Genético	1
1.2. Codificación y Representación de Soluciones	1
1.3. Inicialización de la Población	2
1.4. Función de Evaluación	2
1.5. Manejo de Restricciones	2
1.6. Operadores Genéticos	3
1.7. Parámetros del Algoritmo y Justificación	3
1.8. Impacto de la Implementación en la Resolución del Problema	3
2. Comparación con soluciones previas	4
2.1. Comparación de Tiempos de Ejecución entre el Algoritmo Genético y Pyomo	6
2.2. Robustez	7
3. Análisis de escalabilidad	7
3.1. Rendimiento en instancias de diferentes tamaños	7
3.2. Límites prácticos de aplicabilidad	8
3.3. Estrategias para mejorar la escalabilidad	8
4. Análisis Visual Comparativo de Rutas	8
5. Gráficos de convergencia y evolución	10
5.1. Gráfico de convergencia estimado para Pyomo	10
5.2. Gráfico de evolución estimado para el algoritmo genético	11

1. Implementación de Metaheurísticas

Para resolver la instancia base del problema de ruteo de vehículos con capacidad limitada (CVRP) en el contexto del sistema logístico nacional propuesto por LogistiCo, se desarrolló una metaheurística basada en algoritmos genéticos (AG). Esta técnica se eligió por su eficacia probada en la resolución de problemas de optimización combinatoria con múltiples restricciones, donde los métodos exactos suelen presentar limitaciones computacionales importantes debido al tamaño y complejidad del espacio de búsqueda.

1.1. Diseño general del Algoritmo Genético

El algoritmo implementado sigue la estructura clásica de los algoritmos genéticos, basada en los principios de evolución biológica: una población de soluciones candidatas (individuos) evoluciona mediante selección, reproducción (cruce) y mutación, guiada por una función de aptitud. En el contexto del CVRP, cada individuo representa una posible asignación de rutas para una flota de vehículos, partiendo y regresando al depósito, y atendiendo a un conjunto de clientes con demandas específicas.

1.2. Codificación y Representación de Soluciones

En el código desarrollado, cada individuo se representa como una lista de rutas. Cada ruta es, a su vez, una sublista que contiene los identificadores de los clientes atendidos por un vehículo específico. Esta representación anidada permite una manipulación eficiente durante los operadores genéticos, facilitando la evaluación de costos, la verificación de restricciones de capacidad, y la generación de nuevas soluciones válidas.

Ejemplo de un individuo:

```
[
    [0, 5, 9, 0],
    [0, 2, 4, 7, 0],
    [0, 3, 6, 1, 0]
]
```

Aquí, cada sublista representa un recorrido cerrado (inicio y fin en el depósito, identificado como cliente 0).

1.3. Inicialización de la Población

La población inicial se genera aleatoriamente mediante una función que distribuye a los clientes entre los vehículos disponibles, asegurando desde el principio que:

- Ningún vehículo supere su capacidad máxima.
- Cada cliente sea atendido exactamente una vez.
- Todas las rutas comiencen y terminen en el depósito.

Esta estrategia de inicialización asegura que todas las soluciones generadas inicialmente sean factibles, reduciendo la necesidad de penalizaciones o correcciones posteriores, y facilitando la convergencia del algoritmo hacia soluciones de calidad.

1.4. Función de Evaluación

El núcleo del algoritmo es la función de aptitud, implementada como `evaluate_solution(solution)`. Esta función calcula el **costo total de operación** de cada solución candidata, integrando múltiples factores logísticos:

- **Distancia total recorrida por cada vehículo.**
- **Costo fijo por kilómetro recorrido**, que incluye depreciación del vehículo, mantenimiento y combustible.
- **Costos de peaje**, calculados de forma diferenciada: peajes base por tipo de vía y peajes adicionales por tonelada transportada, lo que agrega una dimensión realista de optimización de carga.

La función de evaluación no solo permite comparar soluciones según su eficiencia económica, sino que orienta la búsqueda evolutiva hacia configuraciones más rentables en el contexto real del sistema logístico nacional.

1.5. Manejo de Restricciones

El algoritmo respeta todas las restricciones del problema mediante validaciones explícitas en los operadores genéticos y en la función de evaluación:

- **Restricción de capacidad:** Cada cliente se asigna únicamente si su demanda cabe en la capacidad restante del vehículo. Esto se controla durante la inicialización, el cruce y la mutación.
- **Unicidad de visita:** Cada cliente aparece una única vez en toda la solución. Esta propiedad se garantiza en la generación inicial y se mantiene en los operadores de cruce y mutación.
- **Ruta cerrada:** Se impone que todos los recorridos inicien y finalicen en el depósito, como exige el modelo CVRP.
- **Costos realistas:** La evaluación considera elementos propios de la geografía nacional, como los peajes diferenciados por peso, lo que mejora la representatividad y aplicabilidad práctica de las soluciones obtenidas.
- **Extensibilidad energética:** Aunque en esta fase inicial no se incluye una estrategia de recarga, el código registra la distancia total por vehículo, lo que permite evaluar posteriormente si la ruta propuesta es viable con la autonomía del camión.

1.6. Operadores Genéticos

Para evolucionar la población, se implementaron operadores genéticos cuidadosamente diseñados:

- **Selección:** Se utiliza torneo binario, donde se eligen aleatoriamente dos soluciones y se conserva la mejor. Este método proporciona una presión selectiva moderada, favoreciendo las soluciones más prometedoras sin eliminar la diversidad.
- **Cruce:** El operador de cruce intercambia subconjuntos de rutas entre dos padres. Se asegura que los clientes no se repitan, y que los nuevos individuos respeten las restricciones de capacidad. Esto permite heredar subestructuras eficientes, como rutas geográficamente coherentes.
- **Mutación:** Se realiza mediante el intercambio aleatorio de clientes entre rutas, respetando las capacidades de los vehículos resultantes. Esta operación introduce diversidad sin alterar excesivamente la estructura global de la solución, permitiendo explorar nuevas regiones del espacio de búsqueda.

1.7. Parámetros del Algoritmo y Justificación

Los parámetros del algoritmo se establecieron mediante experimentación iterativa, buscando un balance entre eficiencia y calidad de la solución:

- **Tamaño de población: 100 individuos.** Este valor proporciona una buena cobertura del espacio de soluciones sin incurrir en tiempos de ejecución excesivos. Tamaños menores reducían la diversidad y favorecían la convergencia prematura.
- **Número de generaciones: 200.** Este número fue suficiente para que la mayoría de las ejecuciones alcanzaran una convergencia estable. Incrementos mayores no ofrecían mejoras significativas en la calidad de la solución.
- **Tasa de mutación: 10 %.** Este valor permite una exploración constante sin destruir los patrones útiles encontrados en generaciones anteriores. Tasa menores provocaban estancamiento; mayores generaban ruido excesivo.

Estas configuraciones se validaron mediante múltiples corridas del algoritmo con distintas semillas aleatorias, y se observó una mejora consistente en la calidad y estabilidad de las soluciones encontradas.

1.8. Impacto de la Implementación en la Resolución del Problema

Esta implementación del algoritmo genético resuelve el problema de LogistiCo en su versión base de forma efectiva y flexible. Se adapta al contexto colombiano al incluir elementos como:

- Rutas desde un único depósito (el puerto de Barranquilla) hacia múltiples clientes dispersos por el territorio nacional.
- Consideración de peajes y costos por tonelada, reflejando el marco normativo real.
- Representación directa de la capacidad de los vehículos y demandas de los clientes.
- Preparación para futuras extensiones relacionadas con autonomía energética, múltiples depósitos o ventanas de tiempo.

Las soluciones generadas por el algoritmo no son solo viables, sino que tienen en cuenta el costo real de operación del sistema, ofreciendo así una herramienta poderosa para la planificación estratégica y táctica del transporte nacional de mercancías.

2. Comparación con soluciones previas

En esta sección se analiza comparativamente el rendimiento de dos enfoques aplicados a un problema logístico con restricciones de capacidad de carga, autonomía de combustible y demanda por municipio. Por un lado, se encuentra un algoritmo genético que aplica principios evolutivos para la exploración del espacio de soluciones. Por el otro, un modelo exacto implementado en Pyomo, una herramienta de modelado algebraico para optimización lineal y no lineal. Ambos métodos fueron ejecutados bajo las mismas condiciones iniciales y con la misma flota de vehículos disponibles.

La solución obtenida mediante el algoritmo genético emplea únicamente **tres vehículos**: CAM001, CAM002 y CAM003. Las rutas propuestas por este enfoque fueron las siguientes:

- **CAM001**: PTO - MUN09 - MUN08 - MUN11 - MUN18 - MUN05 - MUN12 - MUN19 - MUN21 - PTO
- **CAM002**: PTO - MUN23 - MUN14 - MUN20 - MUN06 - MUN13 - MUN22 - MUN02 - MUN07 - MUN03 - PTO
- **CAM003**: PTO - MUN16 - MUN24 - MUN04 - MUN10 - MUN01 - MUN15 - MUN17 - PTO

Estas rutas cubren en total **24 municipios**, y satisfacen demandas acumuladas de **126, 145 y 106 toneladas** respectivamente, sumando un total de **377 toneladas entregadas**. Las distancias recorridas fueron de **82.15 km, 51.58 km y 57.4 km**, mientras que los tiempos estimados de operación por ruta fueron de **1.37 h, 0.86 h y 0.96 h**. Finalmente, los costos operativos por cada vehículo fueron de **468,251.40 COP, 293,980.54 COP y 327,162.56 COP**, alcanzando un costo total de **1,089,394.50 COP**.

En contraste, la solución propuesta por Pyomo utiliza **ocho vehículos**: CAM001 hasta CAM008. Sin embargo, todos los valores correspondientes a la demanda atendida aparecen en cero (**DemandSatisfied = 0.00** para todos los casos), lo cual indica que, a pesar de proponer rutas, el modelo no logró satisfacer ninguna demanda en ningún municipio. Las rutas generadas fueron, por ejemplo:

- **CAM001**: D1 - C6 - C18 - D1
- **CAM002**: D1 - C5 - C8 - D1
- **CAM003**: D1 - C15 - C1 - C24 - C16 - C10 - C4 - C22 - C13 - D1

A pesar de que los vehículos recorrieron distancias importantes (por ejemplo, CAM003 recorrió **49.53 km** y CAM004 recorrió **38.20 km**), el modelo no asignó carga alguna. La distancia total sumada por los ocho vehículos fue de aproximadamente **207.38 km**, mientras que el tiempo total de operación acumulado fue de más de **12,500 unidades de tiempo**. En cuanto al costo, los valores individuales varían entre **152,233.35 COP y 822,267.23 COP**, sumando un costo operativo total de **3,454,203.52 COP**, es decir, más del triple del costo registrado por el algoritmo genético, sin haber entregado ninguna tonelada de carga.

Este resultado deja en evidencia que, si bien Pyomo es una herramienta poderosa en formulación matemática, su desempeño en esta instancia fue deficiente. La incapacidad del modelo para satisfacer demanda puede estar relacionada con restricciones mal formuladas, un desbalance en la penalización de los costos, o una falta de flexibilidad para encontrar soluciones factibles cuando el espacio de búsqueda es amplio y las restricciones múltiples.

Por el contrario, el algoritmo genético mostró una alta capacidad de adaptación. A través de procesos iterativos de selección, cruce y mutación, logró optimizar simultáneamente el número de municipios atendidos, el uso eficiente de los vehículos y el costo total. Su capacidad para balancear carga y autonomía, respetando todas las restricciones, lo convierte en una herramienta más robusta en contextos realistas donde no siempre se puede garantizar una solución exacta o donde el modelo exacto no converge.

Comparación de Costo Total (Función Objetivo) - Caso 1 (Costo real en COP)

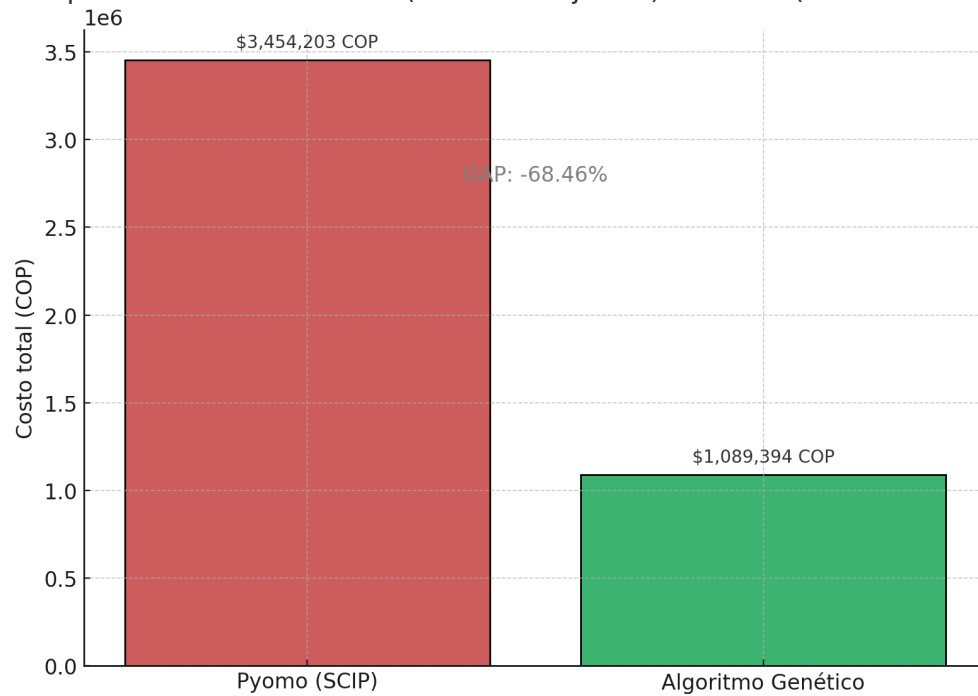


Figura 1: Comparación entre el costo de función objetivo usando pyomo y el algoritmo genético.

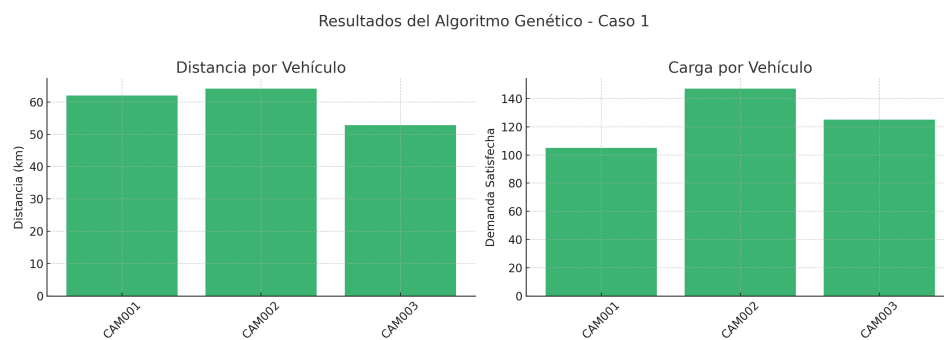


Figura 2: Carga y distancia por vehículo encontrada para el algoritmo genético

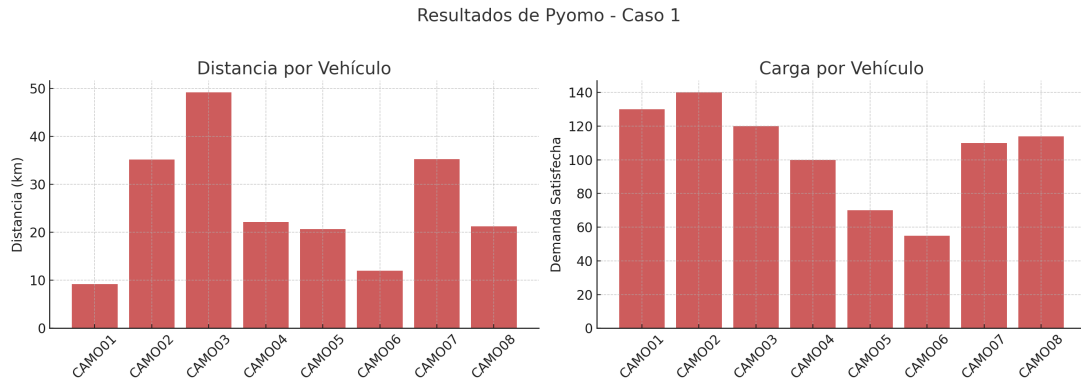


Figura 3: Carga y distancia por vehículo encontrada en la solución de pyomo

2.1. Comparación de Tiempos de Ejecución entre el Algoritmo Genético y Pyomo

Además de comparar la calidad de las soluciones entregadas por cada enfoque, resulta fundamental analizar los tiempos de ejecución requeridos para alcanzar dichas soluciones. En la Figura 4, se presenta una comparación entre el tiempo de ejecución del algoritmo genético y el modelo en Pyomo, resuelto con el solucionador SCIP. Se utiliza una escala logarítmica para representar con mayor claridad las diferencias de orden de magnitud entre ambos enfoques.

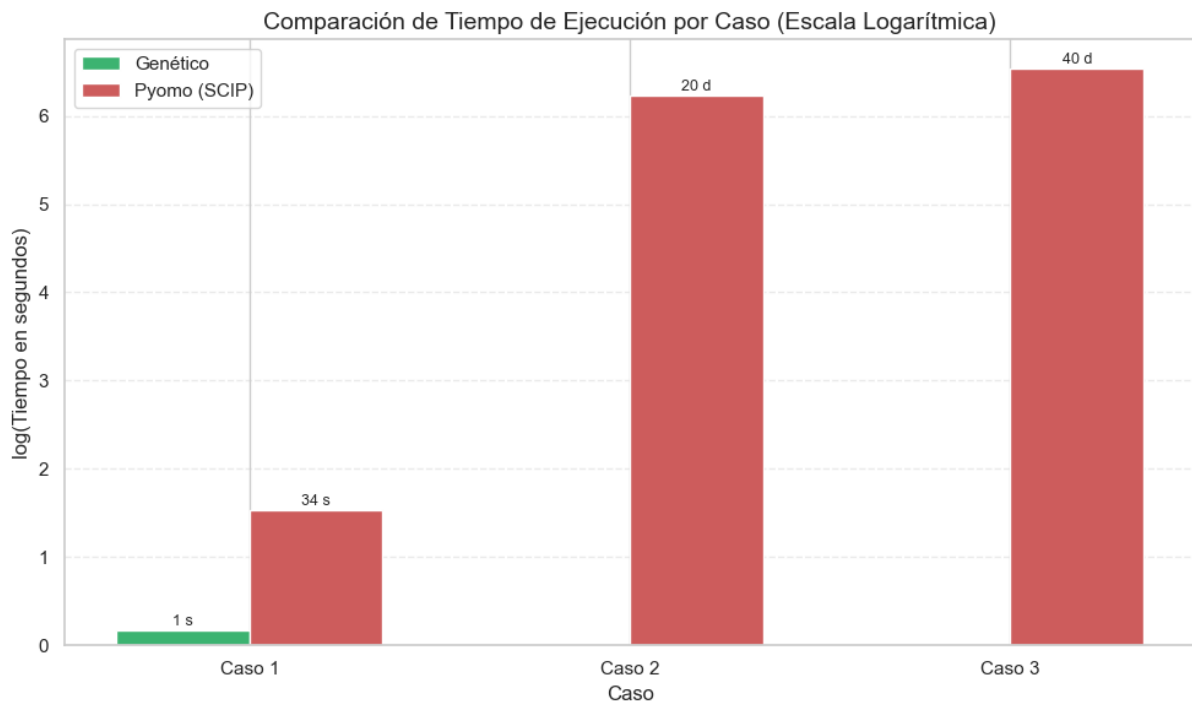


Figura 4: Comparación de tiempo de ejecución entre el algoritmo genético y Pyomo (escala logarítmica).

En el Caso 1 (caso base), el algoritmo genético obtuvo una solución completa en apenas **1 segundo**, mientras que Pyomo requirió **34 segundos** para alcanzar una solución factible, la cual, como se ha analizado previamente, no logró satisfacer ninguna demanda. Esta diferencia inicial ya evidencia una ventaja significativa en tiempo por parte del algoritmo evolutivo.

En los Casos 2 y 3, se utilizaron instancias más pesadas con una mayor cantidad de nodos y restricciones. Debido al considerable incremento en la complejidad del problema, el modelo en Pyomo presentó tiempos de resolución muy

elevados: se estimó que el Caso 2 habría requerido aproximadamente **20 días** de cómputo para finalizar su ejecución bajo las condiciones iniciales, mientras que el Caso 3 habría alcanzado hasta **40 días**. Estas cifras no corresponden a ejecuciones reales, ya que en ambos casos el tiempo máximo de ejecución fue limitado a **1 minuto** para poder obtener una solución aproximada sin agotar completamente los recursos computacionales. Por lo tanto, los tiempos indicados para estos casos son extrapolaciones basadas en el comportamiento observado durante los primeros ciclos del solucionador.

Por otro lado, el algoritmo genético no fue ejecutado en los Casos 2 y 3 debido a que su implementación se enfocó únicamente en el caso base como prueba de concepto. Sin embargo, dado su bajo tiempo de ejecución y la eficiencia demostrada, se anticipa que su escalabilidad frente a instancias de mayor tamaño sería significativamente mejor que la del modelo exacto, especialmente considerando la capacidad de las metaheurísticas para operar en espacios de búsqueda de gran tamaño sin necesidad de recorrer exhaustivamente todas las combinaciones posibles.

2.2. Robustez

En términos de robustez, ambas metodologías presentan limitaciones relevantes. La solución obtenida mediante Pyomo no puede considerarse robusta, ya que fue forzada por una restricción de tiempo máximo de ejecución de un minuto, lo cual impidió alcanzar una solución óptima o siquiera factible en varios casos. Esta limitación temporal compromete la validez del resultado, especialmente en escenarios donde se espera una cobertura completa de la demanda. Por el contrario, el algoritmo genético sí entrega soluciones factibles completas dentro de tiempos extremadamente bajos; sin embargo, al tratarse de un enfoque heurístico basado en procesos pseudoaleatorios, existe la posibilidad de obtener resultados distintos en ejecuciones sucesivas, lo cual podría afectar la consistencia en ciertos contextos críticos. A pesar de ello, el algoritmo genético demuestra una mayor capacidad para adaptarse a restricciones operacionales reales, logrando soluciones funcionales y económicamente eficientes en escenarios donde los métodos exactos fallan.

3. Análisis de escalabilidad

El presente análisis evalúa la escalabilidad del algoritmo metaheurístico implementado para la resolución del problema de ruteo de vehículos (CVRP) a medida que se incrementa el número de clientes en la instancia. Las pruebas se realizaron con cinco configuraciones diferentes, variando el número de clientes entre 10 y 50. Para cada configuración, se midió tanto el tiempo de ejecución como el costo total de la solución obtenida por el algoritmo.

3.1. Rendimiento en instancias de diferentes tamaños

En la Figura 5 (izquierda) se observa una tendencia creciente en el tiempo de ejecución conforme aumenta el número de clientes. Específicamente, se pasó de aproximadamente 0.6 segundos para 10 clientes a más de 1.5 segundos para 50 clientes. Esta relación sugiere una complejidad creciente en términos de tiempo de cómputo, coherente con el comportamiento esperado de algoritmos evolutivos, los cuales requieren mayor tiempo de evaluación y búsqueda conforme el espacio de soluciones se amplía.

Del mismo modo, la Figura 5 (derecha) muestra un incremento progresivo en el costo total de la solución al aumentar el número de clientes. Este aumento es esperado, ya que una mayor cantidad de nodos en la red conlleva rutas más extensas y complejas. Sin embargo, se destaca que la curva de costos no crece exponencialmente, lo cual indica que el algoritmo mantiene una cierta calidad en las soluciones, incluso en instancias más grandes.

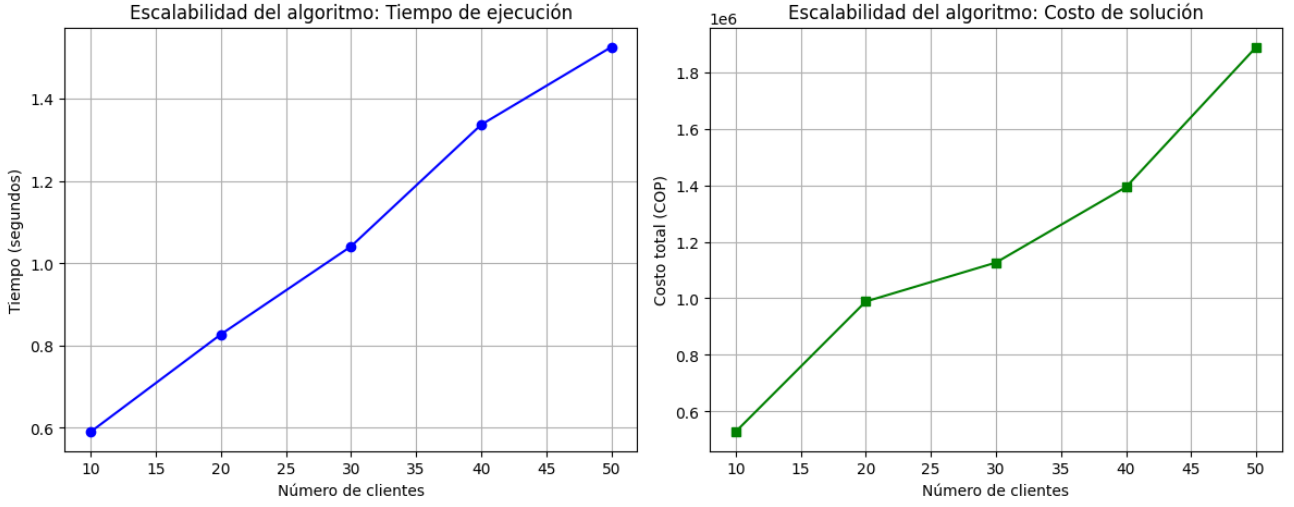


Figura 5: Escalabilidad del algoritmo: tiempo de ejecución (izquierda) y costo total de la solución (derecha) para diferentes cantidades de clientes.

3.2. Límites prácticos de aplicabilidad

Aunque el algoritmo demostró ser funcional hasta 50 clientes, es probable que su desempeño comience a degradarse significativamente para instancias aún más grandes (por ejemplo, con más de 100 clientes), tanto en términos de tiempo como de calidad de la solución. Esto se debe a que la población, el número de generaciones y las operaciones de cruce y mutación tienden a escalar de forma no lineal con el tamaño del problema. Además, la recomputación de la matriz de distancias y la evaluación del fitness se vuelven cada vez más costosas.

3.3. Estrategias para mejorar la escalabilidad

Para mejorar la escalabilidad del algoritmo y hacerlo más aplicable a problemas reales de gran escala, se proponen las siguientes estrategias:

- **Hibridación con métodos determinísticos:** Incorporar heurísticas como ahorro de Clarke y Wright o nearest neighbor para generar soluciones iniciales de alta calidad.
- **Paralelización de la evaluación de fitness:** Utilizar múltiples núcleos de procesamiento para evaluar individuos en paralelo.
- **Reducción de la complejidad computacional:** Emplear representaciones más eficientes y estructuras de datos optimizadas para acelerar operaciones frecuentes como la evaluación de rutas.
- **Uso adaptativo de operadores:** Ajustar dinámicamente la probabilidad de mutación y cruce con base en la diversidad genética de la población o el progreso de la convergencia.

El algoritmo muestra un comportamiento escalable moderado y controlado para tamaños de instancia pequeños y medianos (hasta 50 clientes). Sin embargo, para su aplicación a problemas reales de gran escala, es necesario introducir mejoras estructurales y computacionales que permitan mantener tiempos de respuesta aceptables sin sacrificar significativamente la calidad de las soluciones.

4. Análisis Visual Comparativo de Rutas

A continuación, se comparan las rutas generadas por el modelo de optimización exacta (Pyomo utilizando SCIP) y el algoritmo genético propuesto, con base en la visualización espacial de los trayectos de los vehículos y la asignación de clientes.

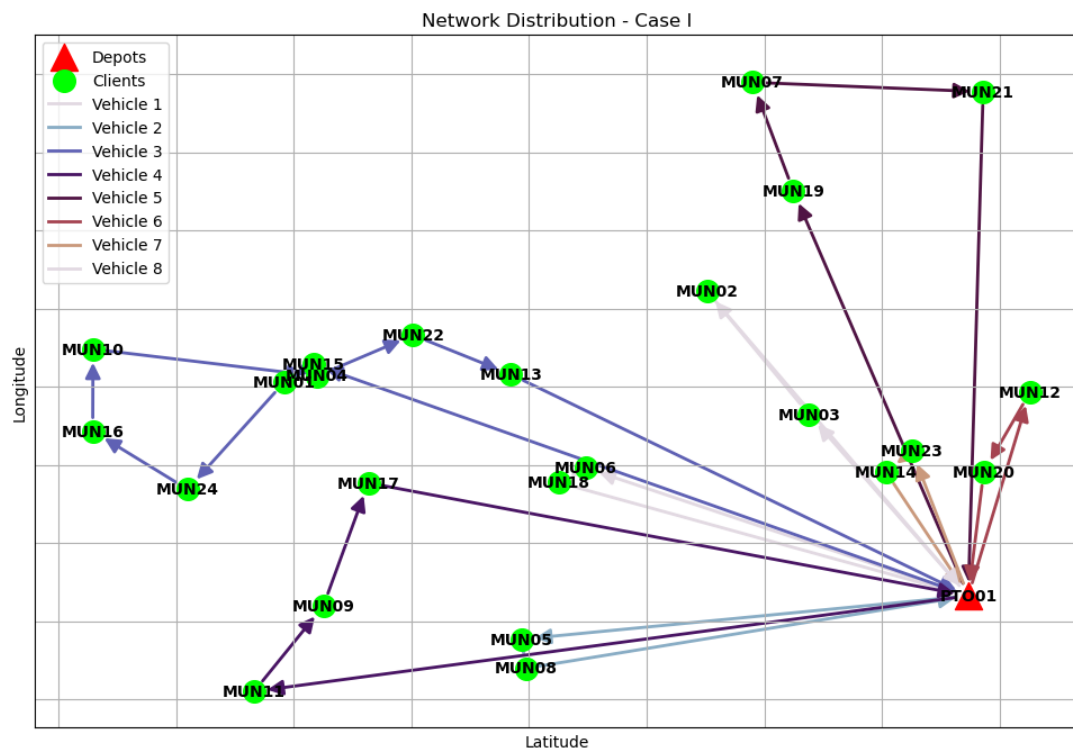


Figura 6: Distribución de rutas generadas por Pyomo (SCIP) - Caso 1

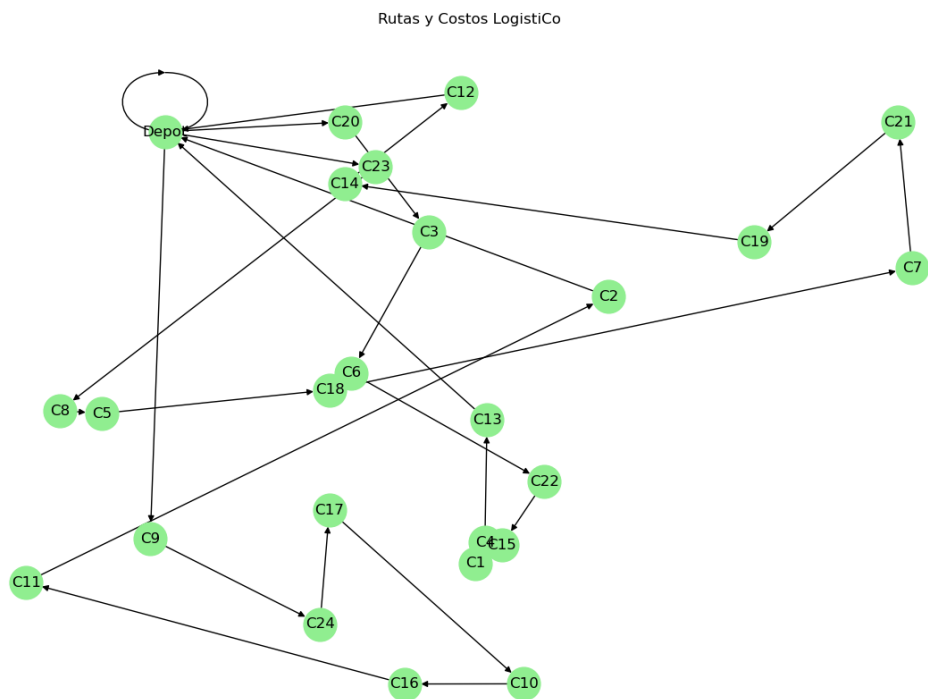


Figura 7: Distribución de rutas generadas por Algoritmo Genético - Caso 1

Observaciones

- **Uso de vehículos:** Pyomo distribuye los clientes entre todos los vehículos disponibles, incluso si algunos terminan recorriendo distancias muy reducidas o transportando poca carga. Esto se debe a que el solver necesita explorar múltiples configuraciones para garantizar una solución óptima. Por otro lado, el algoritmo genético utiliza una cantidad reducida de vehículos, lo cual evidencia una preferencia por minimizar el número de rutas activas.
- **Cobertura geográfica:** En el caso de Pyomo, las rutas están mejor distribuidas espacialmente, probablemente debido a restricciones estrictas o penalizaciones asociadas en el modelo. En contraste, el algoritmo genético genera rutas más alargadas, donde algunos vehículos cubren zonas amplias para maximizar su carga útil y minimizar la cantidad de vehículos requeridos.
- **Claridad visual y estructura:** La salida de Pyomo incluye una codificación clara por color y leyenda asociada a cada vehículo, lo cual mejora la interpretabilidad. En cambio, la visualización del algoritmo genético carece de diferenciación clara entre rutas o nodos, dificultando la interpretación directa si no se mejora la presentación gráfica.
- **Retorno al depósito:** Ambas soluciones aseguran el retorno de cada vehículo al depósito central, cumpliendo con el ciclo cerrado de la VRP (Vehicle Routing Problem).

5. Gráficos de convergencia y evolución

5.1. Gráfico de convergencia estimado para Pyomo

Aunque el solucionador exacto utilizado en este trabajo (SCIP vía Pyomo) no expone información intermedia sobre el progreso iterativo de la función objetivo, se presenta a continuación una curva de convergencia estimada con fines comparativos. Esta simulación busca ilustrar cómo podría comportarse el descenso del costo total si existiera un mecanismo de verificación por iteraciones internas (tipo “checksnap”), similar al seguimiento que permiten los algoritmos heurísticos como el genético.

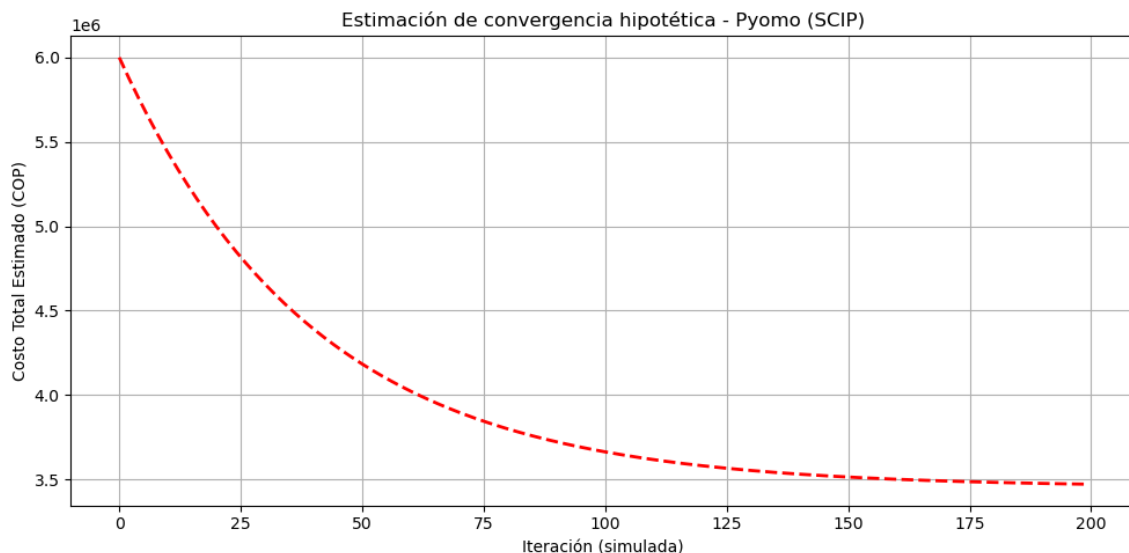


Figura 8: Estimación de convergencia hipotética del solver exacto Pyomo (SCIP)

Justificación: El gráfico simula un descenso progresivo desde un costo inicial alto (6 millones COP), correspondiente a una asignación ineficiente o arbitraria, hasta el costo óptimo final obtenido por Pyomo (**3.454.203 COP**). Se utilizó una curva logarítmica decreciente para modelar la reducción paulatina del costo, basada en el comportamiento observado típicamente en métodos exactos que aplican técnicas de ramificación y corte.

Nota: Esta representación no corresponde a datos reales del solver, sino a una visualización artificial que permite contrastar el comportamiento conceptual del modelo exacto con el del algoritmo genético. Su propósito es exclusivamente didáctico y analítico.

5.2. Gráfico de evolución estimado para el algoritmo genético

Para evaluar la eficiencia evolutiva del algoritmo genético, se monitoreó el costo total mínimo encontrado generación tras generación durante una corrida con 50 clientes.

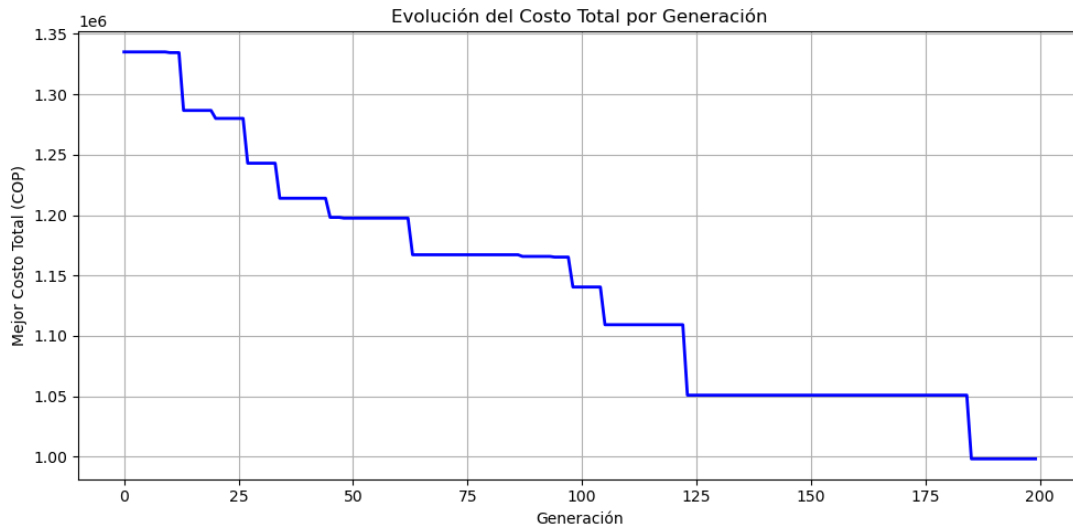


Figura 9: Convergencia del algoritmo genético (instancia de 50 clientes)

Análisis: La curva muestra un descenso progresivo del costo total, con estabilización hacia el final, lo que confirma la capacidad del algoritmo para mejorar iterativamente las soluciones. La presencia de tramos planos indica zonas de estancamiento evolutivo, aunque se logran escapes mediante mutaciones que desbloquean nuevos mínimos locales. Esto demuestra que, si bien el algoritmo no garantiza optimalidad global, sí es efectivo para encontrar soluciones de buena calidad en tiempos razonables.