

Logistic Planning for Used Vehicle Oil Collection Using a Genetic Algorithm

Elio Beltran¹, Pablo Jeger¹, Teresa Unzaga¹, Anibal M. Blanco² y Humberto Heluane¹

¹ Departamento de Ingeniería de Procesos y Gestión Industrial-FACET-UNT, Av. Independencia 1800, San Miguel de Tucumán, 4000, Argentina. Email addresses: {pjeger, tunzaga;hheluane@herrera.unt.edu.ar}

² Planta Piloto de Ingeniería Química (CONICET), Camino La Carrindanga Km. 7, Bahía Blanca, 8000, Argentina. Email address: ablanco@plapiqui.edu.ar

Abstract. Reverse logistics is responsible for managing products that, once consumed, generate waste that could be harmful to the environment and could be recycled for reuse. Used vehicle oil is a an abundant liquid waste, hazardous due to its high content of toxic elements and heavy metals. When these compounds come into contact with water, soil, or the atmosphere, whether by spills or incineration, they cause significant environmental pollution, posing a serious risk to the health of living organisms. To assist in the management of this oil recovery, a genetic algorithm was developed to plan the logistics of collection from the generating points in order to store it in a facility for safe disposal or further use. The aim of this algorithm is to determine the appropriate routes to reduce the variable and environmental operational costs of this process.

Keywords: Genetic Algorithm, Reverse Logistics, Used Oil.

Planificación logística de recolección de aceite vehicular usado mediante un Algoritmo Genético

Resumen. La logística inversa se encarga de la gestión de productos que, una vez consumidos, generan residuos que podrían ser contaminantes del medio ambiente y tienen el potencial de ser reciclados para volver a usarse. El aceite vehicular usado es un residuo líquido muy abundante y peligroso debido a su alto contenido de elementos tóxicos y metales pesados. Al entrar en contacto con el agua, el suelo o la atmósfera, ya sea por derrames o por incineración, estos componentes producen una contaminación ambiental relevante, representando un grave riesgo para la salud de los seres vivos. Con el propósito de asistir en la gestión de recuperación de este aceite, se desarrolló un algoritmo genético para programar la logística de recolección desde los puntos generadores con el fin de acopiarlo en un depósito para su posterior utilización o disposición de manera segura. El objetivo de este algoritmo es determinar las rutas adecuadas de manera de disminuir los costos operativos variables y ambientales de dicha operación.

Palabras claves: Algoritmo genético, Logística inversa, Aceite usado.

1 Introducción

Las actividades productivas requieren de una planificación logística que permita distribuir sus productos de manera eficiente y lograr satisfacer las necesidades de los clientes. Por otro lado, el consumo o uso de dichos productos genera residuos que podrían ser peligrosos para el medio ambiente y que podrían ser reciclados para volver a usarse. Debido a requerimientos legales y eventuales beneficios económicos, las empresas y estados se interesan en la recuperación de los residuos, lo que se conoce como logística inversa. Varios autores (Hammes G. *et al.* 2020; Kabir; M.I. 2013; Chan, F.T.S. *et al.* 2011; Agrawal S. *et al.* 2014) se han dedicado recientemente al estudio de aplicaciones de logística inversa en diferentes campos como, por ejemplo, industrias de la construcción, farmacéutica, automotriz y electrónica, mientras que otros (Alumur S.A. *et al.* 2012; Alshamsi A. *et al.* 2015; Rogers D.S. *et al.* 2012) analizaron el modelado de redes de trabajo generadas por la logística inversa.

Las empresas interesadas en sistemas productivos sustentables han encontrado en la logística inversa una herramienta idónea para lograr coordinar estrategias operativas de recolección, almacenamiento, transporte y distribución para recuperar los productos (Clottey T. *et al.* 2012; Giannetti B.F. *et al.* 2013; Park J.Y. 2014). Esta práctica fomenta la reutilización y reciclado contribuyendo a la conservación de los recursos naturales.

Por otro lado, la logística inversa interesa por el aporte que puede ofrecer como herramienta para la gestión de residuos, especialmente aquellos que son potencialmente peligrosos en aquellos países donde existen regulaciones o leyes específicas sobre el manejo de residuos (Ardjmand E. *et al.* 2015; Gomes C.S. *et al.* 2008; Matar N. *et al.* 2014).

Los aceites usados son clasificados por la ONU como residuos peligrosos debido a su alto contenido de elementos tóxicos y metales pesados, derivados de la degradación de aditivos y componentes durante su uso, y que, al entrar en contacto con el agua, el suelo o la atmósfera, ya sea por derrames o por incineración, producen una contaminación ambiental significativa, representando un grave riesgo para la salud de los seres vivos. Como consecuencia, el mal uso o descarte inadecuado del aceite vehicular usado, puede a largo plazo afectar la biodiversidad y generar riesgos para la salud humana.

Para diseñar sistemas de gestión eficientes, la logística inversa puede beneficiarse de las metodologías provenientes de la investigación operativa para desarrollar herramientas de asistencia a la toma de decisiones basados en modelos matemáticos de optimización. En particular, el Algoritmo Genético (AG) es un método de búsqueda que imita el proceso de selección natural en el cual dos nuevos resultados se obtienen de la combinación de dos resultados obtenidos precedentemente (Russell S.J. *et al.* 2003). Las soluciones obtenidas por este método metaheurístico son una alternativa aceptable frente a métodos de optimización que requieren una cantidad considerable de recursos computacionales para ser resueltos, especialmente en problemas combinatorios de gran porte como usualmente lo son la planificación de transporte y logística (Vidal T. *et al.* 2012; Allahviranloo M. *et al.* 2014). En este trabajo se estudió la planificación de la recolección y el almacenamiento de aceite usado mediante AG y se logró determinar

las rutas adecuadas con el objetivo de disminuir los costos operativos y ambientales de dicha operación.

2 Planteamiento del Problema

El problema que se quiere resolver se puede resumir de la siguiente manera.

A partir de los siguientes datos:

- Los puntos generadores de aceite usado
- La cantidad de aceite generado por cada uno de los puntos
- La ubicación del centro de acopio del residuo
- Las distancias entre todos los nodos (puntos de generación y de acopio)
- La velocidad promedio de los camiones
- La velocidad de carga y descarga de aceite según la bomba del camión
- El tiempo muerto en cada punto de recolección
- La flota de camiones disponible que realizará la recolección, capacidad máxima y características de los vehículos.
- El costo unitario de transporte
- El costo de C en CO₂ equivalente

Se quiere determinar:

- La cantidad de viajes de recolección que deben realizarse
- El recorrido que debe realizarse en cada viaje
- La distancia recorrida en cada viaje
- El tiempo de duración de cada viaje
- La carga recolectada en cada viaje
- Los costos de operación y ambientales

La selección de los camiones a ser utilizados en la recolección del residuo se realiza teniendo en cuenta la oferta de dichos vehículos en el mercado y la normativa vigente con respecto a la circulación de vehículos pesados en la ciudad. Adicionalmente, en un distrito dado, es posible determinar la ubicación geográfica de todos los puntos generadores y la cantidad de aceite usado colectado por mes en cada uno de ellos. Para ubicar el centro de acopio se adopta un punto equidistante a todos los nodos de generación. Para este fin se pueden emplear herramientas como Google Maps, así como para calcular las distancias entre los puntos entre sí y de cada uno de ellos con el depósito. Las distancias calculadas deben, idealmente, tener en cuenta el sentido de circulación por la ciudad.

3 Algoritmo Genético

Un Algoritmo Genético (AG) es una metaheurística inspirada por el proceso de selección natural para buscar soluciones de buena calidad a problemas de optimización

no lineales y mixto enteros a partir de operadores de selección, cruzamiento y mutación. En un AG, una población se compone de soluciones al problema de optimización (individuos) que evolucionan hacia mejores valores del indicador de desempeño adoptado.

La evolución comienza con una población de individuos inicial generada aleatoriamente y es sometida a un proceso iterativo, donde cada iteración es llamada “generación” en el cual se aplican los operadores antes mencionados. En cada generación se evalúa la función objetivo (fitness) de cada individuo. Cuanto mejor sea el fitness de un individuo, mayores son sus probabilidades de ser seleccionado para participar en la creación de la siguiente generación.

3.1 Representación de las soluciones

Para este problema, cada individuo de la población debe representar una solución del problema, es decir, una posible manera de recolectar todo el aceite de los puntos generadores, partiendo del depósito y volviendo nuevamente a él. Debido a que se debe respetar tanto la limitación de capacidad de los camiones como el tiempo de operación máximo establecido, esta recolección puede demandar más de un viaje.

En este trabajo, cada individuo es representado como una permutación de los primeros n números naturales, donde n es el número de nodos o puntos de recolección (sin contar el depósito). Por ejemplo, la Fig. 1 representa un individuo para una red de 10 nodos.

5	3	8	1	2	6	7	9	10	4
---	---	---	---	---	---	---	---	----	---

Fig. 1: Representación de un individuo como una permutación de los nodos.

Para cada individuo, los recorridos de los viajes deben determinarse a partir del siguiente algoritmo:

1. El camión del primer viaje sale del depósito y se dirige al primer nodo del individuo (nodo 5 en la Fig. 1).
2. Como el camión del primer viaje es cargado con el aceite disponible en este nodo, se recalcula la capacidad disponible del camión para el resto del viaje.
3. Como el camión del primer viaje debe trasladarse hacia el primer nodo, cargar el aceite y eventualmente volver a descargarlo en el depósito, se recalcula el tiempo disponible del primer viaje considerando la velocidad promedio del camión, la velocidad de carga y descarga de la bomba y el tiempo muerto en el primer nodo.
4. Para evaluar si el segundo nodo del individuo (nodo 3 en la Fig. 1) puede ser incluido en el recorrido del primer viaje, se chequean dos condiciones:
 - a. Que la cantidad de aceite a recolectar en este nodo no supere la capacidad restante del camión recalculada en el punto 2.

- b. Que el tiempo que implica incluir a este nodo en el recorrido (o sea, la suma del tiempo de viaje del nodo 5 al 3, el tiempo de viaje del nodo 3 al depósito, el tiempo de carga de aceite en el nodo 3, el tiempo muerto en el nodo 3 y el tiempo de descarga del nuevo aceite recolectado en el depósito) no supere el tiempo disponible recalculado en el punto 3.
- 5. Si no se cumple alguna de las condiciones 4a y 4b, el primer viaje vuelve desde el primer nodo hacia el depósito, y un segundo viaje comienza desde el depósito hacia el segundo nodo del individuo.
- 6. Si en cambio se cumplen las condiciones 4a y 4b, el segundo nodo es incluido en el recorrido del primer viaje.
- 7. Se repiten los puntos 2 y 3 para recalcular las nuevas disponibilidades de volumen y tiempo y se repite el punto 4 para evaluar si el tercer nodo del individuo (nodo 8 en la Fig. 1) se incluye o no en el recorrido del primer viaje.
- 8. Estos pasos se repiten hasta que, eventualmente, un último viaje incluye el último nodo del individuo (nodo 4 en la Fig. 1). Nótese que, a lo sumo, se necesitan n viajes para recorrer todos los nodos.

3.2 Inicio de la población

Inicialmente, se genera aleatoriamente una cantidad Pob de permutaciones de los n nodos, donde Pob es un parámetro conocido como “tamaño de la población”.

3.3 Evaluación de la función objetivo

Cada individuo de la población es evaluado según la Ec.(1).

$$FO = c_u \sum_{k=1}^M D_{tot}(k) + (c_d + w c_c) \sum_{i=1}^{T(k)} C_{mo}(i, k) \quad (1)$$

Donde:

- M es el número total de viajes necesarios.
- $D_{tot}(k)$ es la distancia total recorrida en el k -ésimo viaje.
- c_u es el costo variable unitario sin tener en cuenta el combustible (lubricantes, filtros, neumáticos y mantenimiento del vehículo)
- $T(k)$ es el total de trayectos que se realizan en el k -ésimo viaje.
- $C_{mo}(i, k)$ es el consumo de combustible en el i -ésimo trayecto del k -ésimo viaje.
- c_d es el costo unitario del combustible.
- w factor de emisión de CO_2 .
- c_c es el costo de C en emisiones de CO_2 .

A su vez, el consumo de combustible en cada trayecto se calcula según las Ecs. (2) y (3).

$$C_{mo}(i, k) = \rho(i, k) \cdot D(i, k) \quad (2)$$

$$\rho(i, k) = \rho_o + \frac{\rho_f - \rho_o}{Q_T} \cdot G(i, k) \quad (3)$$

Donde:

- $\rho(i, k)$ es el consumo específico en el i-ésimo trayecto del k-ésimo viaje.
- $D(i, k)$ es la distancia recorrida en el i-ésimo trayecto del k-ésimo viaje.
- ρ_o es el consumo específico cuando el camión está vacío.
- ρ_f es el consumo específico cuando el camión transporta la carga máxima.
- Q_T es la carga máxima del camión.
- $G(i, k)$ es la carga del camión en el i-ésimo trayecto del k-ésimo viaje.

3.4 Elite

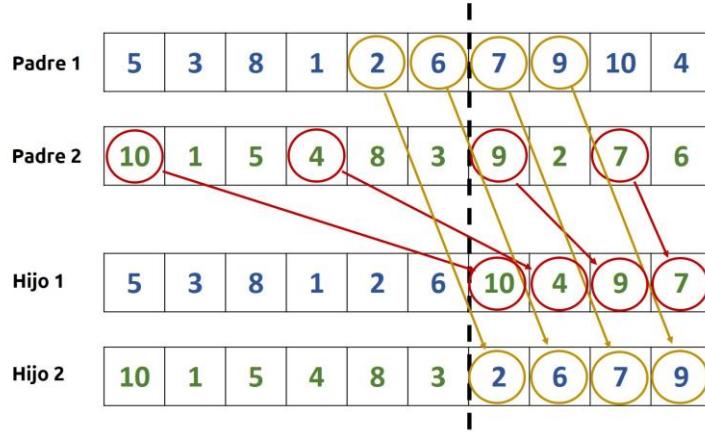
Una vez evaluado el total de la población, el mejor individuo (elite) es guardado. Este proceso se repite al principio de cada generación. Cada vez que aparece un mejor individuo, la elite es modificada.

3.5 Selección

Después de que todos los individuos de una generación son evaluados, el AG debe elegir los mejores individuos según su fitness. En este trabajo se adoptó el método conocido como SUS (“*Stochastic Universal Sampling*”) (Baker, J.E. 1987)

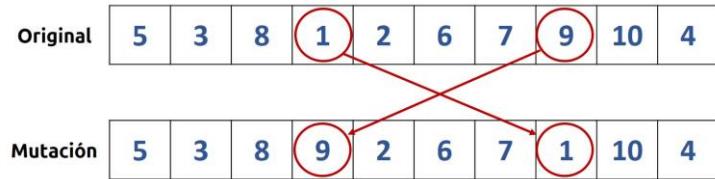
3.6 Cruzamiento

A continuación, la población seleccionada es agrupada de a pares (llamados “padres”) que son combinados para generar dos nuevos individuos (llamados “hijos”). Esta operación se denomina cruzamiento. Se dice que el cruzamiento se encarga de la explotación del espacio de búsqueda. Para este trabajo se adopta el método de cruzamiento denominado *partially mapped crossover* (Goldberg, D.E. *et al.* 1985) y se cruza al 100% de la población (es decir, no hay clonación más allá de la elite). La Fig. 2 muestra un ejemplo del operador utilizado de cruzamiento entre dos individuos.

**Fig. 2:** Operador de cruceamiento utilizado en el AG

3.7 Mutación

El operador de mutación se utiliza para mantener una diversidad genética en la población. A diferencia del operador de cruceamiento, la mutación se aplica a los elementos de la población individualmente. En general no se muta el total de la población en cada generación, sino sólo algunos elegidos al azar. Se dice que el operador de mutación se encarga de la exploración del espacio de búsqueda. La cantidad de individuos mutados en la población es $p_m \cdot Pob$ donde p_m es la fracción de población mutada y también es un parámetro del AG. La Fig. 3 muestra un ejemplo del operador de mutación utilizado en este AG, denominado *swap mutation* (Chen, S.H. *et al.* 2011).

**Fig. 3:** Operador de mutación utilizado en el AG

3.8 Terminación

El AG repite los operadores de evaluación, selección, cruceamiento y mutación en cada iteración o generación hasta que se cumpla alguna condición de terminación. En general esta terminación consiste en un tiempo de ejecución específico, en un número total de generaciones o bien en un número total de generaciones sin encontrar un nuevo mejor punto. A fines de propiciar una comparación equitativa entre las distintas estrategias de resolución, se ejecutará el algoritmo en un tiempo fijo de ejecución. La Fig. 4 representa en un diagrama de flujo el funcionamiento del AG utilizado en este trabajo.

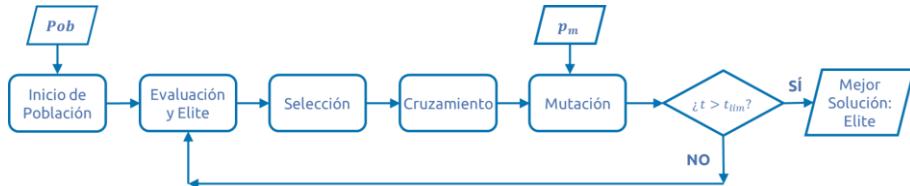


Fig. 4: Diagrama de flujo del AG considerado.

4 Caso de estudio inicial

Inicialmente, se planteó un caso de estudio sencillo con 10 centros de generación de aceite, para testear las prestaciones del algoritmo propuesto. Las distancias de todos los pares de nodos posibles, considerando incluso el depósito o centro de acopio (nodo 0), y la cantidad de aceite a recolectar Q de cada nodo se presenta en el apéndice

La Tabla 1 presenta los valores del resto de parámetros necesarios para el planteo del problema.

Tabla 1: Valores de los parámetros del modelo

c_u : costo variable unitario	0,173 [U\$D/km]
c_d : costo del combustible	0,915 [U\$S/l]
w : factor de emisión de CO ₂	$2,61 \times 10^{-3}$ [ton CO ₂ /l]
C_c : costo de emisión de CO ₂	2 [U\$D/ton CO ₂] *
Capacidad de los camiones	5.000 [L]
Velocidad de succión de la bomba	50 [L/min]
Tiempo máximo de la jornada laboral	8 [h]
Tiempo muerto por nodo (ingreso, egreso del vehículo y tareas administrativas)	20 [min]
Velocidad promedio del camión	30 [km/h]

* (<http://www.carbonneutralplus.com>)

4.1 Solución

La solución de este problema posee un valor de la función objetivo de U\$D 16,38, distribuidos de la siguiente manera: U\$D 8,30 en combustible, U\$D 8,03 en otros costos variables y U\$D 0,047 en costo de C en emisiones de CO₂ equivalente. Esta solución fue obtenida previamente utilizando Programación Lineal Mixta Entera (MILP) con un modelo implementado en GAMS usando CPLEX como Solver en un procesador con 8 Gb de memoria RAM. Se obtuvo y demostró la optimalidad de esta solución en 3.219 segundos de ejecución.

Tabla 2: Volumen de aceite recolectado, tiempo y distancia recorrida en cada ruta.

Ruta	Nodos visitados	Vol. de aceite recolectado [L]	Tiempo usado [h]	Distancia [km]
1	0-8-5-0	3500	3,61	8,4
2	0-2-9-1-0	3300	3,99	13,9
3	0-3-4-7-10-0	4600	5,35	18,6
4	0-6-0	2500	2,52	5,5

En la Tabla 2 se detallan los nodos visitados, el volumen de aceite recolectado, la distancia recorrida y el tiempo utilizado en cada viaje. Todos los nodos son visitados sólo una vez y cada ruta comienza y termina en el centro de acopio (nodo 0). Además, en cada viaje el camión carga un volumen inferior a la capacidad permitida y cada viaje se realiza en un tiempo menor a la jornada laboral establecida.

4.2 Calibración del Algoritmo Genético

Con el objetivo de encontrar valores apropiados para los parámetros del AG (Pob y p_m), se estudió la respuesta del AG variando los valores de estos parámetros. Para esto, se tomaron valores del parámetro Pob entre 100 y 1.500 individuos con un paso de 100, y se tomaron valores del parámetro p_m entre 10% y 30% con un paso del 5%. Para cada combinación de valores de Pob y p_m , se realizaron 100 ejecuciones del AG con tiempo de ejecución de 1 minuto para el caso de estudio inicial de 10 nodos.

Si bien para todos los valores de los parámetros Pob y p_m el AG logró encontrar la solución óptima en más del 75% de sus ejecuciones, el mejor rendimiento se logró para una población de 500 individuos y una probabilidad de mutación del 20%. En este caso, el AG encontró la solución óptima en 88 de 100 ejecuciones de 1 minuto.

Se tomaron estos valores de los parámetros Pob y p_m para las ejecuciones del AG en la sección siguiente.

5 Respuesta del AG ante redes de mayor tamaño

5.1 Porcentaje de aciertos para redes de 20 nodos o menos

El algoritmo genético fue ejecutado para seis casos de estudio, considerando redes de 10, 12, 14, 16, 18 y 20 nodos. Para cada caso de estudio, se realizaron en total 400 ejecuciones correspondiendo 100 ejecuciones a cada tiempo de ejecución máximo de 1, 2, 3 y 4 minutos.

Para comparar los distintos experimentos se define el porcentaje de aciertos como: el número de veces que el algoritmo converge a la menor de todas las soluciones encontradas en las distintas corridas sobre el total de ejecuciones realizadas.

La variación de este porcentaje de aciertos con el número de nodos en la red y el tiempo de ejecución del AG se presenta en la Fig. 5.

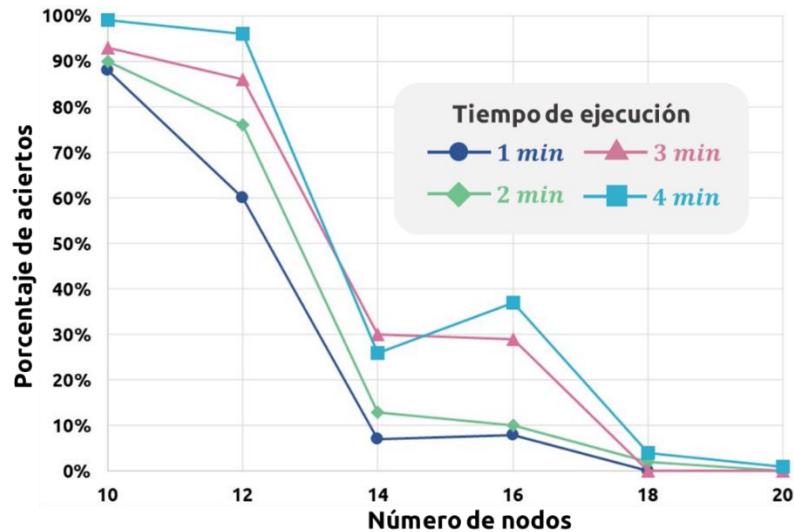


Fig. 5: Porcentaje de aciertos del AG para redes de hasta 20 nodos.

Por un lado, puede observarse que, en general y como es esperable, un mayor tiempo de ejecución implica un mejor rendimiento del AG. Por ejemplo, en 4 minutos de tiempo máximo de ejecución, el AG encontró la solución óptima del caso de estudio inicial 99 de 100 veces, y repitió 90 veces la solución mínima para una red de 12 nodos. Sin embargo, para redes de 14 nodos en adelante, el porcentaje de aciertos del algoritmo genético decrece notablemente, y para redes de 20 nodos el AG no puede repetir dos veces el mismo valor mínimo de su función objetivo. Se concluye entonces que, en tiempos de ejecución cortos, el AG propuesto carece de reproducibilidad significativa en redes mayores de 14 nodos.

5.2 Dispersión en redes de hasta 70 nodos

Si bien los resultados del AG pueden no ser reproducibles para redes de mayor tamaño, se consideró estudiar la dispersión de los resultados obtenidos por el AG. Para esto, se tomaron casos de estudio de 20, 30, 40, 50, 60 y 70 nodos. Para cada caso de estudio, se ejecutó 200 veces el AG (100 veces en 1 minuto y 100 veces en 10 minutos). Para cada caso de estudio y cada tiempo de ejecución, se calculó el valor medio de la función objetivo obtenida. Luego, se calculó la diferencia porcentual entre el valor medio obtenido y el mínimo correspondiente a cada caso de estudio en particular. Si los resultados obtenidos por el AG en 100 ejecuciones no se alejan demasiado del valor mínimo obtenido, esta diferencia porcentual debería ser baja.

Los resultados de las diferencias porcentuales obtenidas se presentan en la Fig. 6.

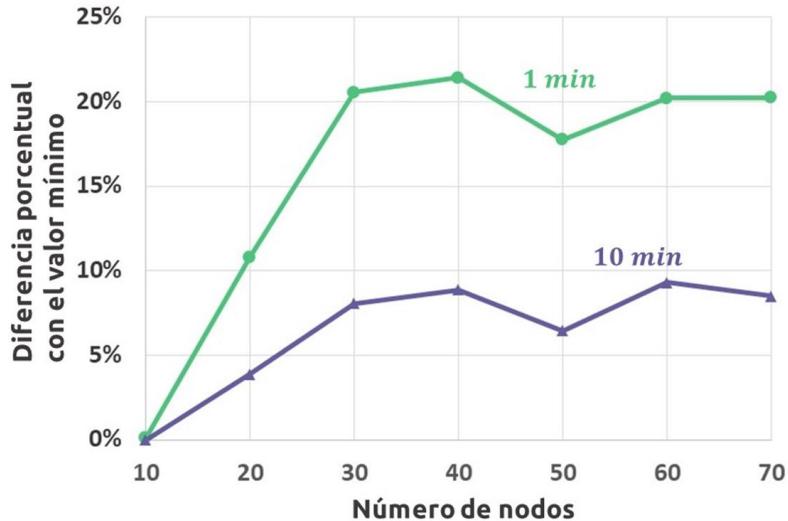


Fig. 6: Diferencia porcentual con el mínimo para redes de hasta 70 nodos.

Por un lado, se observa que, de 20 a 30 nodos, la diferencia porcentual entre el valor medio y el valor mínimo de función objetivo obtenidos crece notablemente (de 10,8% a 20,6% en 1 minuto de ejecución y de 3,9% a 8,1% en 10 minutos de ejecución). Sin embargo, a partir de 30 nodos, esta diferencia porcentual se mantiene acotada (menos del 22% en 1 minuto y menos del 10% en 10 minutos). Además, el aumento en el tiempo de ejecución de 1 a 10 minutos produce una mejora notable y significativa en el rendimiento medio del AG para todos los casos de estudio analizados.

6 Conclusiones

En este trabajo se desarrolló un Algoritmo Genético (AG) para la planificación logística de recolección de aceite vehicular usado en ciudades, con el fin de almacenarse y eventualmente reutilizarse.

La tarea del AG es determinar el recorrido de los camiones encargados de la recolección de este aceite. El AG codifica las soluciones como individuos, las modifica y las evalúa según una función objetivo. Esta función objetivo contempla no solo los costos del combustible consumido por los camiones de recolección sino también los costos adicionales del vehículo (lubricantes, filtros, neumáticos y mantenimiento) e incluso el costo de carbono emitido como CO₂ equivalente, considerando que estos camiones utilizan combustibles fósiles. De esta manera, se pretende contemplar tanto el costo económico de la planificación logística como su impacto ambiental.

Inicialmente, se tomó un caso de estudio de 10 nodos, para el cual se contaba con una solución óptima encontrada con Programación Lineal Mixta Entera. El Algoritmo Genético logró encontrar esta misma solución óptima en 88 de 100 ejecuciones de 1

minuto, y en 99 de 100 ejecuciones de 4 minutos. Sobre la base de este resultado se concluye que el AG se presenta como una alternativa capaz de encontrar soluciones óptimas en tiempos de ejecución cortos, si bien se sabe no es capaz de garantizar la optimalidad de las soluciones encontradas.

Posteriormente, se estudió la respuesta del AG en redes de hasta 20 nodos. En estos casos, se observó que la reproducibilidad del AG disminuye notoriamente a partir de los 14 nodos. Para tiempos de ejecución de 4 minutos y redes de 20 nodos, el AG ya no es capaz de encontrar dos veces la misma solución.

Finalmente, se estudió la respuesta del AG en redes de hasta 70 nodos para estudiar la dispersión de los resultados obtenidos. Se encontró entonces que la diferencia porcentual entre el valor medio y el valor mínimo de función objetivo encontrados es menor al 22% en tiempos de ejecución de 1 minuto, y menor al 10% en tiempos de ejecución de 10 minutos. Además, esta diferencia porcentual se mantiene estable con el crecimiento del tamaño de la red a partir de 30 nodos.

Los resultados del estudio en redes de hasta 70 nodos indican que, si bien el AG no garantiza soluciones óptimas, sí puede encontrar soluciones aceptables para redes de gran tamaño en tiempos de ejecución de 10 min. Por otro lado, es esperable que un aumento adicional en el tiempo de ejecución pueda reducir incluso más la dispersión de los resultados encontrados.

Durante todo el trabajo, se tomaron tiempos de ejecución relativamente cortos (entre 1 y 10 minutos). Esto se debe a que, para cada caso de estudio, se realizaron 100 ejecuciones para obtener valores medios representativos del funcionamiento del AG propuesto, y por lo tanto estudiar el resultado del AG para una ejecución de una hora requiere 100 horas (más de cuatro días) por caso de estudio.

Como trabajos futuros, se estudiará la respuesta del AG para mayores tiempos de ejecución. Además, se pretende incluir una variación en los parámetros del AG con el tiempo de ejecución (un aumento progresivo en el tamaño de la población o en la probabilidad de mutación, para que la búsqueda de mejores soluciones se vuelva más agresiva con el correr del algoritmo). Como alternativa, se considerará la posibilidad de que los camiones puedan recoger una parte del aceite disponible en un punto de recolección si no cuentan con la capacidad disponible para recoger el total. De esta manera, los vehículos regresarán al centro de acopio cargados completamente, y la carga restante sería recogida en un nuevo viaje.

Agradecimientos. Este trabajo fue parcialmente financiado por la U. Nacional de Tucumán a través del Proyecto PIUNT E723 (2023-2026) y Conicet PIP 0308.

Bibliografía

1. Hammes G., De Souza E.D., Taboada Rodriguez C. M., Rojas Millan R.H., Mojica Herazo J.C.. (2020). Evaluation of the reverse logistics performance in civil construction. *Journal of Cleaner Production*, Vol. 248, 119212.
2. Kabir; M.I. (2013). Reverse Logistics in Pharmaceutical Industry, *International Journal of Information Systems and Supply Chain Management*, 2 (1):96-100.

3. Chan, F. T. S., Chan H.K., Jain, V. (2011). A framework of reverse logistics for the automobile industry. *International Journal of Production Research*, 50(5): 1318–1331.
4. Agrawal S., Singh R.K., Murtaza, Q. (2014). Forecasting product returns for recycling in Indian electronics industry. *J Adv Manag Res* 11(1):102–14.
5. Alumur S.A., Nickel S., Saldanha-da-Gama F., Verter V. (2012). Multi-period reverse logistics network design. *Eur J Oper Res* 220 (1):67–78
6. Alshamsi A., Diabat A. (2015). A reverse logistics network design, *Journal of Manufacturing Systems*, 37(3): 589-598.
7. Rogers D.S., Melamed B., Lembke R.S. (2012). Modeling and Analysis of Reverse Logistics, *Journal of Business Logistics*, 3(2):107-117
8. Clottee T., Benton W.C. Jr, Srivastava R. (2012). Forecasting product returns for remanufacturing operations. *Decis Sci* 43(4):589–614.
9. Giannetti B.F., Bonilla S.H., Almeida C.M. (2013). An energy-based evaluation of a reverse logistics network for steel recycling. *J Clean Prod* 46:48–57.
10. Park J.Y. (2014). Assessing determinants of industrial waste reuse: the case of coal ash in the United States. *Resour Conserv Recycl* 92:116–27.
11. Ardjmand E., Weckman Z., Park N., Taherkhani P., Singh M. (2015). Applying genetic algorithm to a new location and routing model of hazardous materials. *Int J Prod Res* 53(3):916–28.
12. Gomes C.S., Nunes K.R., Xavier L.H., Cardoso R., Valle R. (2008). Multi criteria decision making applied to waste recycling in Brazil. *Omega* 36(3):395–404.
13. Matar N., Jaber M.Y., Searcy C. (2014). A reverse logistics inventory model for plastic bottles. *Int J Logist Manag* 25(2):315–33
14. Russell S.J., Norvig P. (2003) *Artificial Intelligence: A Modern Approach* Upper Saddle River, Prentice-Hall, New Jersey, EEUU.
15. Vidal T., Crainic T. G., Gendreau M., Lahrichi N., Rei W. (2012). A hybrid genetic algorithm for multidepot and periodic vehicle routing problems, *Oper. Res.*, 60(3), 611624.
16. Allahviranloo M., Chow J.Y.J., Recker W.W. (2014). Selective vehicle routing problems under uncertainty without recourse. *Transp. Res. E. Logistics Transp. Rev.*, Vol. 62, 6888.
17. Baker, J. E. (1987). Reducing Bias and Inefficiency in the Selection Algorithm. *Proceedings of the Second International Conference on Genetic Algorithms and Their Application*. Hillsdale, New Jersey: L. Erlbaum Associates: 14–21.
18. Goldberg, D.E.; Lingle, R. (1985), Grefenstette, John J. (ed.), "Alleles, loci, and the traveling salesman problem", *Proceedings of the First International Conference on Genetic Algorithms and Their Applications (ICGA)*, Hillsdale, N.J.: Lawrence Erlbaum Associates, pp. 154–159.
19. Chen, S. H., Chen, M. C. (2011). Operators of the two-part encoding genetic algorithm in solving the multiple traveling salesmen problem. *Proceedings - 2011 Conference on Technologies and Applications of Artificial Intelligence, TAAI 2011*.
20. Carbon Neutral Homepage, <http://www.carbonneutralplus.com>

Apéndice

Distancias entre nodos [km], y cantidad de aceite a recolectar en cada nodo [L/semana]

	1	2	3	4	5	6	7	8	9	10	11
1	0	4,4	3,2	3,5	5,4	1,2	2,7	5,3	3,5	6,7	5,1
2	3,8	0	2,2	6,1	7,5	5,6	2,7	5,6	8,2	3,5	2,5
3	2,9	1,2	0	4,7	6,8	3,4	1,9	6,6	6,3	3,6	3,6
4	3,7	6,3	5,7	0	2,0	4,9	4,7	4,4	6,1	8,6	4,3
5	5,5	7,1	7,4	2,4	0	6,6	5,5	5,2	7,5	8,8	5,2
6	1,7	5,6	2,9	4,6	6,8	0	2,6	6,8	3,5	6,5	6,9
7	2,8	3,0	2,3	4,1	4,8	3,0	0	6,0	6,2	3,9	1,7
8	5,1	6,1	7,0	4,1	5,3	6,3	5,1	0	7,4	10,0	3,2
9	3,8	8,9	6,1	5,7	6,7	3,2	5,9	7,6	0	9,8	7,4
10	5,9	3,3	3,3	7,2	9,3	6,1	4,4	10,9	9,4	0	5,3
11	4,7	2,4	4,3	4,3	5,5	5,3	2,9	3,0	7,7	5,8	0
Q/L	—	1000	800	800	800	1000	2500	1500	2500	1500	1500