

Inteligencia Artificial

Informe Final: Milk Collection Problem

Ricardo Lorca De La Hoz

31 de octubre de 2023

Evaluación

Mejoras 2da Entrega (10 %):	_____
Código Fuente (10 %):	_____
Representación (5 %):	_____
Descripción del algoritmo (15 %):	_____
Experimentos (10 %):	_____
Resultados (40 %):	_____
Conclusiones (10 %):	_____
Nota Final (100):	_____

Resumen

El **Problema de Recolección de Leche con Mezcla (MCPB)** es una variante del **Problema de Enrutamiento de Vehículos (VRP)** que se centra en recoger leche de diferentes granjas y entregarla a una planta. El desafío principal es mezclar diferentes tipos de leche en los camiones, lo que resulta en la calidad de leche más baja. El objetivo es maximizar las ganancias y cumplir con restricciones como la capacidad del camión, cuotas mínimas, etc. A lo largo de la historia de VRP se han desarrollado algoritmos como Adaptive Large Neighborhood Search (ALNS) y Ant Colony Optimization (ACO) para dar una solución al problema de recolección de leche y mejorar la eficiencia en la recolección de leche. Este informe tiene como objetivo proporcionar una comprensión clara de **MCPB**, destacando su complejidad y las estrategias utilizadas para su resolución.

1. Introducción

Este informe tiene como objetivo abordar el Problema de Recolección de Leche con Mezcla (MCPB), una variante del Problema de Enrutamiento de Vehículos (VRP) que involucra la recolección de leche de granjas y su posterior entrega a una planta procesadora. El propósito principal es proporcionar una comprensión clara de MCPB, destacando su complejidad y las estrategias utilizadas para su resolución.

El informe se estructura en tres secciones principales. En primer lugar, se presenta una definición detallada del problema (**sección 2**), describiendo sus componentes esenciales y las restricciones asociadas. A continuación, se aborda el Estado del Arte (**sección 3**), donde se explora la relevancia y las soluciones propuestas en la literatura para MCPB. Luego, se presenta un Modelo Matemático (**sección 4**) que formaliza el problema y sus variables, junto con la función objetivo y las restricciones correspondientes. En las secciones Representación y Descripción del algoritmo (**sección 5 y 6**) se describen las estructuras utilizadas además de la forma en la que se implementa el algoritmo. Finalmente, en las **secciones 7 y 8**, se realizan experimentos con

diferentes parámetros del algoritmo y se obtienen resultados y conclusiones que pueden ayudar a una mejor implementación.

La motivación detrás de este estudio radica en la importancia de optimizar la recolección de leche, ya que esto puede conducir a la reducción de costos operativos y al aumento de las ganancias en la industria lechera.

2. Definición del Problema

El **problema de recolección de leche con mezcla** trata principalmente de un problema de enrutamiento de vehículos **VRP**, en el cual se cuenta con granjas/nodos, que producen cierta cantidad de leche de un tipo específico, y con una planta procesadora la cual envía camiones a recolectar esta leche para poder procesarla, para el caso estudiado, se cuenta con camiones que poseen un único compartimiento, por lo que al llevar 2 o más tipo de leche a la vez, estas se mezclan quedando con la calidad de la leche que peor calidad tenga al momento de la mezcla, donde, se sabe que la leche de tipo A es de mejor calidad que la leche de tipo B y esta a su vez es de mejor calidad que la leche de tipo C, tal como se puede observar en la figura 1, la cual muestra un ejemplo sencillo de un problema de recolección de leche, donde, aparte de los elementos ya mencionados, se puede apreciar que para ir de un nodo a otro hay un costo asociado, que puede ser debido al gasto de gasolina, al pago del conductor, etc. adicionalmente se debe tener en cuenta otros parámetros del problema, cómo la cuota mínima de cada tipo de leche, la ganancia que genera cada tipo de leche y la capacidad de los camiones para transportar cierta cantidad de leche.

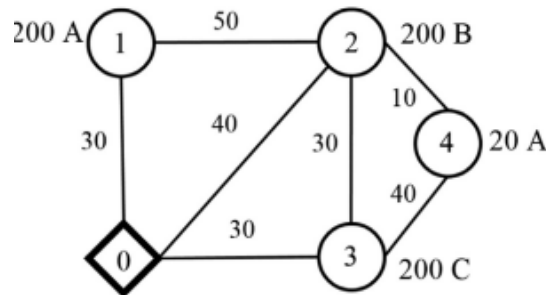


Figura 1: Red de un problema de recolección de leche
Fuente: A milk collection problem with blending[7].

Para identificar la/las variable(s) del problema se precisa saber que es lo que se puede decidir de tal forma que sea útil para la resolución, de este problema en específico se debe considerar que cada camión realiza un solo recorrido en el tiempo de simulación del problema (1 día), por lo que **se podría utilizar la ruta como variable**, ya que esto limitaría a los camiones a realizar solo una ruta, suponiendo también que cuando un nodo aparece en la ruta del camión entonces el camión cargó toda la leche de la granja correspondiente a ese nodo, se podría disminuir el espacio de búsqueda de forma que en el dominio de las variables se descartarían todas las rutas en las que se supere la capacidad del camión, para que la solución a este problema sea factible, se tienen que cumplir ciertas restricciones:

- Se debe cargar la leche de cada granja solo una vez: Solo un camión puede pasar por un nodo en específico y este camión no puede pasar otra vez por el mismo nodo.
- Se debe cargar toda la leche de la granja: cuando un camión pasa por una granja, este se lleva toda la leche que produjo la granja.

- No se debe superar la capacidad máxima de los camiones: como se explica arriba, los camiones tienen una cierta capacidad la cual debe ser mayor o igual a la cantidad de leche que transportan.
- Un camión sale de la planta procesadora y vuelve a ella al final del día: por lo que un camión no puede realizar más de un recorrido.
- Mezcla de leches: como se explica más arriba, al mezclarse la leche, queda con la calidad de la leche con peor calidad al momento de la mezcla, por ejemplo, si se mezcla leche de **tipo a** con leche de **tipo b**, el resultado será leche **tipo b**.
- Se debe superar la cuota mínima de leche: cada día hay una cuota mínima para cada tipo de leche, la cual debe ser cumplida para poder procesar la leche.

Finalmente, se puede notar que el problema requiere de un objetivo para ser resuelto, en este caso, al existir parámetros como la ganancia que genera cada tipo de leche y los costos asociados a ir de un nodo a otro, se hace evidente que el objetivo principal para una planta procesadora de leche debería ser el aumentar las ganancias, ya que, no tiene sentido hablar de objetivos como cumplir la cuota mínima, ya que estos están incluidos en las restricciones.

Se pueden encontrar algunas variantes para este problema:

- Los camiones pueden poseer distintos compartimentos para llevar la leche, por lo que se puede llevar más de un tipo de leche por camión.
- Existencia de puntos de recolección donde un grupo de granjas acercan la leche, por lo que no es necesario que un camión pase por las granjas, si es que estas dejaron su leche en un punto de recolección.
- Los camiones poseen un único compartimento y las leches de distinto tipo se mezclan (caso tratado en el informe).

3. Estado del Arte

En la literatura, el problema de enrutamiento de vehículos (Vehicle Route Problem) ha sido de sumo interés, para la investigación, desde el año 1959, cuando surge el problema de ruteo de vehículos, tras ser propuesto por George Dantzig y John Ramser, teniendo como objetivo el poder generar soluciones con las que se puedan cumplir las distintas restricciones que presenta el ruteo de vehículos en la vida real [4], teniendo sus raíces en la logística y distribución de bienes, el VRP nace como una variación del problema del vendedor viajero, propuesto en 1956 por Flood, evolucionando para adaptarse a diversas aplicaciones, entre las cuales se encuentra el caso de estudio de este informe, la recolección de leche, el cual surge en 1994 [9], siendo este un problema en el que se busca recolectar leche de granjas en distintas direcciones y entregarla a una planta de procesamiento mediante camiones, su representación más común es mediante un grafo, donde las granjas son nodos, los arcos entre nodos tienen costos asociados y cada nodo tiene atributos como la cantidad y el tipo de leche que produce la granja, la relevancia de Milk Collection Problem with Blending nace de la necesidad de mezclar diferentes tipos de leche con tal de poder reducir los costos de transporte, necesidad presente en el mundo real, aunque con una complejidad mucho mayor, como puede ser el caso de Chile, que se ha visto afectado debido a los costos, la calidad y los precios a los que se vende la leche que se produce en el país, por lo que han surgido investigaciones del problema de recolección de leche con mezcla gradual[6], donde, la leche no se mezcla de forma discreta, por lo que, al mezclar una leche de calidad mayor con una de calidad menor, la leche resultante será de una calidad intermedia, dando resultados distintos al problema con mezcla discreta.

Instance	Farms distribution 1						Farms distribution 2						Δ (%)			
	Gradual blending			Discrete blending			Gradual blending			Discrete blending						
	Z	IG (%)	T	Z	IG (%)	T	Z	IG (%)	T	Z	IG (%)	T				
a32	30,898.0	0.00	122	26,960.0	0.00	34	26,960.0	14.61	40,366.0	0.00	192	31,478.0	0.00	96	33,878.0	19.15
a33	36,287.0	0.00	9132	29,417.0	0.00	89	29,417.0	23.35	41,527.0	0.16	14,400	36,033.0	0.00	38	36,033.0	15.25
a34	35,466.0	0.07	14,400	30,536.0	0.00	66	30,536.0	16.14	44,646.0	0.00	2203	38,046.0	0.00	33	38,046.0	17.35
a36	33,476.0	0.69	14,400	29,513.0	0.00	138	29,513.0	13.43	40,246.0	0.10	14,400	32,455.0	0.00	56	32,455.0	24.01
a37	30,032.0	0.00	1720	24,837.0	0.00	38	24,837.0	20.92	38,346.0	0.00	2248	31,558.0	0.00	125	31,558.0	21.51
a38	35,998.0	0.16	14,400	28,596.0	0.00	704	28,596.0	25.88	44,610.0	0.00	2912	35,613.0	0.00	217	35,613.0	25.26
a39	35,720.0	0.25	14,400	31,067.0	0.00	73	31,067.0	14.98	45,220.0	0.17	14,400	35,773.0	0.00	460	35,773.0	26.41
a44	46,630.0	0.13	14,400	39,072.0	0.00	144	39,072.0	19.34	51,734.0	0.22	14,400	41,705.0	0.00	935	41,705.0	24.05
a45	47,144.0	0.37	14,400	40,922.0	0.00	188	40,922.0	15.20	54,246.0	0.13	14,400	42,437.0	0.00	908	47,705.0	12.64
a46	49,272.0	0.32	14,400	40,696.0	0.00	175	40,696.0	21.07	55,919.0	0.31	14,400	47,705.0	0.00	690	47,705.0	17.22
a48	45,360.0	0.44	14,400	39,947.0	0.00	2082	39,947.0	13.55	59,545.0	0.15	14,400	48,881.0	0.00	1642	48,881.0	21.82
a53	56,686.0	0.60	14,400	46,662.0	0.00	160	46,662.0	21.48	67,071.0	0.22	14,400	53,573.0	0.12	14,400	58,181.0	15.28
a54	27,030.0	0.93	14,400	22,414.0	0.00	217	22,414.0	20.59	31,666.0	0.42	14,400	24,421.0	0.00	607	24,421.0	29.67
a55	27,560.0	0.18	14,400	24,694.0	0.00	339	24,694.0	11.61	38,712.0	0.27	14,400	31,690.0	0.00	473	31,690.0	22.16
a60	30,490.0	0.41	14,400	24,934.0	0.00	438	24,934.0	22.28	38,268.0	0.40	14,400	32,101.0	0.00	366	32,101.0	19.21
a61	71,317.0	0.60	14,400	60,644.0	0.00	304	60,644.0	17.60	82,213.0	0.14	14,400	64,268.0	0.04	14,400	64,268.0	27.92
a62	27,216.0	1.87	14,400	22,949.0	0.00	340	22,949.0	18.59	31,869.0	0.69	14,400	26,507.0	0.19	14,400	26,507.0	20.23
a63	31,059.0	1.19	14,400	25,826.0	0.00	409	25,826.0	20.26	41,352.0	0.41	14,400	33,808.0	0.00	158	33,808.0	22.31
a64	32,388.0	0.75	14,400	26,117.0	0.00	2236	26,117.0	24.01	39,257.0	0.54	14,400	32,603.0	0.01	14,400	32,603.0	20.41
a65	29,900.0	0.46	14,400	28,046.0	0.00	467	28,046.0	6.61	38,784.0	0.60	14,400	28,687.0	0.66	14,400	34,227.0	13.31
a69	30,636.0	0.60	14,400	25,872.0	0.00	709	25,872.0	18.41	39,049.0	0.85	14,400	31,910.0	0.00	9361	31,910.0	22.37
a72	90,978.6	0.08	14,400	74,695.1	0.00	601	74,695.1	21.80	111,024.0	0.13	14,400	93,336.7	0.00	13,151	93,336.7	18.95
a80	37,346.0	1.28	14,400	29,997.0	0.00	4135	29,997.0	24.50	43,581.0	1.28	14,400	34,944.0	0.40	14,400	34,944.0	24.72
att48	44,475.0	1.82	14,400	20,402.0	0.00	2300	20,402.0	117.99	65,000.0	9.75	14,400	38,330.0	3.78	14,400	38,330.0	69.58
c50	59,065.0	0.34	14,400	50,128.0	0.00	145	50,128.0	17.83	73,331.0	0.14	14,400	59,241.0	0.04	14,400	59,241.0	23.78
c75	105,589.0	0.35	14,400	88,828.0	0.00	3631	88,828.0	18.87	124,185.0	0.59	14,400	101,765.0	0.00	1931	121,125.0	2.53
eii22	19,696.0	0.00	201	15,947.0	0.00	33	15,947.0	23.51	20,086.0	0.00	15	18,075.0	0.00	11	18,075.0	11.13
eii23	8,908.3	0.00	229	7,207.3	0.00	92	7,207.3	23.60	9,669.0	0.00	23	8,641.0	0.00	12	8,641.0	11.90
eii30	9,143.0	1.09	14,400	7,117.0	0.00	128	7,117.0	28.47	11,671.0	0.00	422	9,542.0	0.00	80	9,542.0	22.31
eii31	73,720.0	0.09	14,400	60,080.0	0.00	88	60,080.0	22.70	64,515.0	0.02	8733	59,222.0	0.00	82	59,222.0	8.94
eii33	23,916.0	0.00	4138	20,409.0	0.00	51	20,409.0	17.18	24,585.0	0.00	960	21,584.0	0.00	236	21,584.0	13.90
eii51	58,365.0	0.06	14,400	50,128.0	0.00	392	50,128.0	16.43	73,282.0	0.21	14,400	59,221.0	0.07	14,400	59,221.0	23.74
eii76	107,161.0	0.36	14,400	91,461.0	0.00	1180	91,461.0	17.17	124,154.0	0.38	14,400	101,767.0	0.00	2650	121,127.0	2.50
eii101	106,342.0	15.10	14,400	97,657.0	0.00	10,814	97,657.0	8.89	135,369.0	0.40	14,400	106,598.0	0.39	14,400	118,918.0	13.83
f45	28,535.5	0.01	14,400	23,705.0	0.00	141	23,705.0	20.38	33,783.5	0.12	14,400	29,668.0	0.00	12,584	29,668.0	13.87
f71	90,528.5	0.18	14,400	72,863.7	0.00	6775	72,863.7	24.24	109,965.0	0.19	14,400	89,137.2	0.00	9978	89,137.2	23.37
f135	8,006.4	45.58	14,400	7074.6	17.00	14,400	7074.6	13.17	9,704.8	35.78	14,400	9422.0	16.13	14,400	9,422.0	3.00
tai75A	48,311.5	0.77	14,400	41,349.5	0.00	4031	41,349.5	16.84	62,471.0	0.00	12,584	53,222.0	0.27	14,400	53,222.0	17.38
tai75B	56,270.5	1.92	14,400	49,461.0	0.00	1111	49,461.0	13.77	68,920.0	0.74	14,400	56,637.5	0.00	2764	66,359.5	3.86
tai75C	31,446.5	0.45	14,400	26,123.5	0.00	2559	26,123.5	20.38	43,775.0	4.42	14,400	37,749.5	0.49	14,400	37,749.5	15.96
tai75D	52,084.5	0.06	14,400	42,484.0	0.00	2559	42,484.0	22.60	67,851.0	0.83	14,400	56,540.0	0.50	14,400	56,540.0	20.01
tai100A	8,432.1	46.61	14,400	8,000.8	0.88	14,400	8,000.8	5.39	10,716.7	27.04	14,400	9,148.2	34.52	14,400	9,148.2	17.15
tai100B	11,879.1	21.87	14,400	10,115.8	1.94	14,400	10,115.8	17.43	14,912.8	12.80	14,400	12,991.0	22.93	14,400	12,603.2	18.33
tai100C	15,055.9	9.80	14,400	13,377.4	0.44	14,400	13,377.4	12.55	17,417.5	10.54	14,400	15,162.2	4.22	14,400	15,162.2	14.87
tai100D	8,170.0	34.75	14,400	6,958.7	0.65	14,400	6,958.7	17.41	10,625.6	14.35	14,400	8,173.2	33.42	14,400	8,173.2	30.01
mn101	742.0	43.55	14,400	208.0	1.54	14,400	208.0	256.73	1,067.0	65.76	14,400	878.0	17.64	14,400	878.0	21.53
mn121	285.5	111.28	14,400	-922.5	146.25	14,400	-922.5	130.95	434.4	75.57	14,400	294.7	115.15	14,400	294.7	47.40
Average	-	7.39	12,892	-	3.59	3211	-	27.89	-	5.68	11,981	-	5.34	7397	-	19.83

Figura 2: Comparación de estrategias de mezcla gradual y discreta en instancias de prueba. Fuente: Vehicle routing for milk collection with gradual blending: A case arising in Chile [6].

Donde se puede observar que, en promedio, para la primera distribución, la ganancia final Z del caso de mezcla gradual, es un 27,89 % más que en el caso de mezcla discreta, mientras que en la segunda distribución, la ganancia final del caso de mezcla gradual, fue solo de un 19,83 % más que en el caso de mezcla discreta.

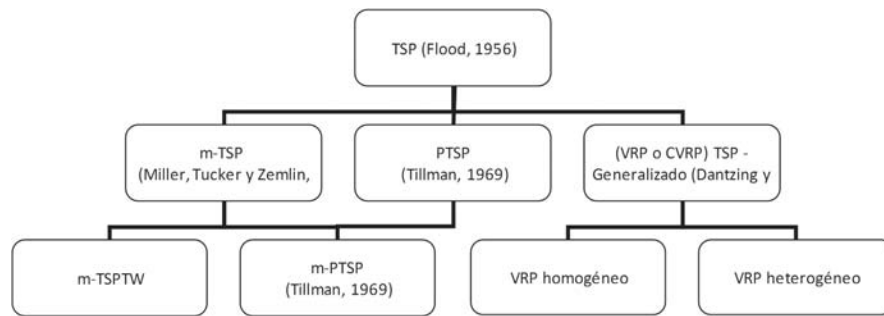


Figura 3: Modelos originarios del problema VRP Fuente: State of the art review of the vehicle routing problem: A historic account with solving methods[5].

Debido a la complejidad de que presentan problemas como VRP a la hora de obtener soluciones óptimas en un tiempo polinomial, para su resolución se ha optado por la utilización de métodos exactos, para cierto casos, técnicas de atomización combinatorial, como pueden ser las técnicas de optimización local convencional y las técnicas de optimización local inteligente, es

decir, heurísticas y metaheurísticas, respectivamente[3].

Entre los algoritmos mas destacados para la resolución de VRP y MCPB se encuentran:

- **ALNS (Adaptive Large Neighborhood Search):** Este algoritmo a demostrado ser eficiente para resolver instancias grandes de VRP, adaptando la exploración de vecindario según la situación. *El ALNS esta diseñado para tener en cuenta la naturaleza jerárquica del problema. Se demuestra empíricamente que este enfoque multinivel conduce a soluciones mucho mejores que el enfoque clásico basado en el cliente*[1].
- **ACO (Ant Colony Optimization):** Inspirado en el comportamiento de las colonias de hormigas, ha sido aplicado con éxito en la resolución de problemas de rutas, incluida la recolección de leche[2].

Tipo de método	Algoritmos
Metaheurísticas: métodos de solución capaces de escapar de los óptimos locales y realizar una búsqueda robusta del espacio de soluciones, a través de la combinación de procedimientos de mejora locales y estrategias de nivel superior (Glover y Kochenberger, 2003).	Búsqueda tabú: incluye una valoración de soluciones, tácticas de búsqueda local, criterio de finalización y elementos como tabúes en una lista y longitud de la misma (Jia et al., 2013).
	Optimización por colonia de hormigas (ACO): es un algoritmo de inteligencia colectiva inspirado en el comportamiento de alimentación de especies de hormigas (Ding et al., 2012).
	Algoritmos genéticos: técnica de búsqueda heurística adaptativa que opera sobre una población de soluciones (Pereira y Tavares, 2009).
	Búsqueda de vecindad variable (VNS): método de trayectoria que se basa en el principio de las vecindades que cambian de forma sistemática (Khouadjia et al., 2012).
Heurísticas: son métodos que realizan una exploración relativamente limitada del espacio de búsqueda y por lo general producen buenas soluciones dentro de tiempos de cálculo razonables. Además, la mayoría de estos métodos fácilmente se pueden adaptar a nuevas limitaciones encontradas en los contextos de la vida real. Las heurísticas se pueden agrupar en tres tipos: de dos fases, constructivas y métodos de mejoramiento (Toth y Vigo, 2002).	Búsqueda local iterativa (ILS): técnica basada en el principio de optimalidad próxima (Michallet et al., 2014).
	Heurísticas de dos fases: existen tres familias de métodos, los métodos de agrupar primero y ruteo después; los métodos de ruteo primero y agrupar después y los algoritmos de pétalo (Toth y Vigo, 2002). De los tres métodos el más común es de agrupar primero y ruteo después, los cuales pueden utilizar una gran variedad de métodos para agrupar los vértices a visitar, desde el clásico algoritmo de barido hasta la "Gran búsqueda adaptativa de vecindario propuesta en (Pisinger y Ropke, 2007).
	Heurísticas constructivas: el ejemplo más claro es el algoritmo de ahorros de Clarke y Wright, el cual va poco a poco construyendo una solución factible sin perder de vista el costo de la solución, sin embargo, no contienen una fase de mejora (Toth y Vigo, 2002).
Métodos Exactos: se clasifican en tres categorías: programación lineal entera, programación dinámica y búsqueda directa de árbol (Laporte y Nobert, 1987).	Heurísticas de mejoramiento: tratan de actualizar cualquier solución factible mediante la realización de una secuencia de intercambios de borde o vértice dentro o entre las rutas de los vehículos (Toth y Vigo, 2002).
	Programación lineal entera mixta (MILP): contiene tres tipos de formulaciones, partición de conjuntos, flujo de vehículos y flujo de mercancías asociado a flujos en los arcos (Laporte y Nobert, 1987).
	Programación dinámica. Consiste en encontrar el costo mínimo alcanzable mediante una función de costo recursiva (Laporte, 1992).
	Búsqueda directa de árbol: consiste en construir secuencialmente rutas de vehículos por medio de un árbol de ramificación y acotamiento (Laporte y Nobert, 1987).

Figura 4: Algoritmos de solución del VRP

Fuente: Modeling a vehicle routing problem with multiple depots, time windows and heterogeneous fleet of a courier service[8].

4. Modelo Matemático

Para el modelo matemático[7], se cuenta con los siguientes parametros:

- Grafo completo $G(N, A)$, donde N es el conjunto de nodos que representan a las granjas productoras, y A es el conjunto de arcos o caminos de un nodo a otro.

- N_0 Se define como $N \cup \{0\}$, donde 0 es el nodo que identifica la ubicación de la planta procesadora.
- A^0 Define el conjunto de arcos que conectan la planta con las granjas.
- K es el conjunto de camiones disponibles para hacer recorridos.
- T es el conjunto de las calidades de leche.
- N^t Identifica a las granjas que producen leche de calidad $t \in T$.
- D^t es el conjunto de calidades de leche r donde la mezcla de r y t resulta en r .
- IT es un conjunto de pares ordenados (i, t) de la granja i y calidad de leche t , ya que cada granja produce solo una calidad de leche.
- Q^k es la Capacidad del camión k .
- q_i^t es la cantidad de leche t producida por la granja i .
- c_{ij}^k es el costo de viaje del camión k para ir desde el nodo i al nodo j , donde $(i, j) \in A \cup A_0$.
- α^t es la ganancia por unidad de cierta calidad de leche t , es decir, para si vendo cierta cantidad de leche t , gano lo correspondiente a α^t .
- P^t es la cuota mínima diaria de leche de calidad t que debe entrar a la planta.

se definen las siguientes variables:

$$x_{ij}^k = \begin{cases} 1 & \text{Si el camión } k \text{ viaja directamente desde el nodo } i \text{ al nodo } j. \forall i, j \in \text{nodos} \\ 0 & \text{en caso contrario} \end{cases}$$

$$y_i^{kt} = \begin{cases} 1 & \text{Si el camión } k \text{ carga leche de calidad } t \text{ de la granja } i. \forall t \in A, B, C \text{ y } \forall i \in \text{nodos} \\ 0 & \text{en caso contrario} \end{cases}$$

$$z^{kt} = \begin{cases} 1 & \text{Si el camión } k \text{ entrega leche de calidad } t \text{ a la planta. } \forall t \in A, B, C \text{ y } \forall k \in \text{camiones} \\ 0 & \text{en caso contrario} \end{cases}$$

w^{kt} = Cantidad(en unidades volumétricas) de leche de calidad t que el camión k entrega a la planta

v^{tr} = Cantidad de leche de calidad t entregada a la planta, mezclada para su uso como leche de calidad r

La funcion objetivo para el problema tratado es la siguiente:

$$Z = \max_{t \in T} \sum_{r \in T} \alpha^r v^{tr} - \sum_{(i,j,k) \in AK} c_{ij}^k x_{ij}^k$$

Donde se busca maximizar la diferencia entre las ganancias de la leche que llega a la planta en contraposición a los costos asociados por ir a recoger la leche de las granjas, donde se puede observar que $\alpha^r v^{tr}$ toma en cuenta la calidad de la leche después de mezclarse. Esta función está sujeta a las siguientes restricciones:

$$1. \quad \sum_{t \in T} \sum_{i \in N: (i,t) \in IT} q_i^t y_i^{kt} \leq Q^k \quad \forall k \in K$$

2. $\sum_{k \in K_i} y_i^{kt} = 1$ $\forall i \in N, t \in T : (i, t) \in IT$

3. $\sum_{j: (0_k, j, k) \in AK} x_{0_k j}^k \leq 1$ $\forall k \in K$

4. $\sum_{i: (i, j, k) \in AK} x_{ij}^k = \sum_{h: (j, h, k) \in AK} x_{jh}^k$ $\forall k \in K, j \in N_0$

5. $\sum_{p: (p, i, k) \in AK} x_{pi}^k = y_i^{kt}$ $\forall k \in K_i, i \in N, t \in T : (i, t) \in IT$

6. $z^{kt} \leq 1 - \sum_{\substack{r \in D^t: r \neq t, \\ (i, r) \in IT}} y_i^{kr}$ $\forall k \in K_i, i \in N, t \in T$

7. $\sum_{t \in T} z^{kt} \leq 1$ $\forall k \in K$

8. $w^{kt} \leq z^{kt} Q^k$ $\forall k \in K, t \in T$

9. $w^{kt} \leq \sum_{r: t \in D^r} \sum_{h \in N^r} q_h^r y_h^{kr}$ $\forall k \in K, t \in T$

10. $\sum_{k \in K} \sum_{t \in T} w^{kt} = \sum_{(i, t) \in IT} q_i^t$

11. $\sum_{r \in D^t} v^{tr} = \sum_{k \in K} w^{kt}$ $\forall t \in T$

12. $\sum_{t \in T} v^{tr} \geq P^r$ $\forall r \in D^t$

13. $y_i^{kt} + y_i^{kr} \leq 1$ $\forall (t, r) \in PM, \forall (i, t), (j, t) \in IT$

14. $\sum_{i \in S} \sum_{j \in S} x_{ij}^k \leq |S| - 1$ $\forall S \subseteq N, \forall k \in K$

$$15. \quad y_i^{kt}, z^{kt} \in \{0, 1\} \quad \forall i \in N, \forall k \in K_i, \forall t \in T : (i, t) \in IT$$

$$16. \quad x_{ij}^k \in \{0, 1\} \quad \forall (i, j, k) \in AK$$

$$17. \quad w^{kt}, v^{tr} \geq 0 \quad \forall k \in K, \forall t, r \in T, \forall r \in D^t$$

- (1) Limita la cantidad de leche que carga el camión en base a su capacidad.
- (2) Requieren que la leche de cada granja sea recolectada por exactamente un camión.
- (3) Impone una ruta como máximo para cada camión, que debe comenzar en algún nodo 0_k para cada camión k .
- (4) Son ecuaciones de balance de flujo para cada nodo y cada camión.
- (5) La ruta del camión k , según las restricciones, debe detenerse en el nodo i si recoge la leche de ese nodo.
- (6) Evitan que un camión cargue leche de calidad inferior a t si está entregando leche de calidad t a la planta.
- (7) Requieren que un camión k contenga solo una calidad de leche, posiblemente mezclada.
- (8) Establecen la relación entre las variables continuas w_{kt} , o el volumen de leche t , y las variables binarias z_{kt} , o la asignación de calidad de leche t al camión k .
- (9) Mide el volumen de leche A, B o C que llega a la planta en el camión k .
- (10) Obliga el transporte de toda la leche producida a la planta.
- (11) Equilibran la cantidad de cada calidad de leche que llega a la planta y la cantidad de leche restante de cada calidad después de la mezcla en la planta.
- (12) Aplican las cuotas de la planta.
- (13) Evitan mezclas prohibidas.
- (14) Previenen sub-rutas no deseadas para cada camión.
- (15)-(17): Establecen el dominio de las variables.

5. Representación

Para el manejo de la problemática presentada, se utilizan 3 estructuras 'Truck', 'Milk' y 'Node', que representan un camión, un tipo de leche y una granja o planta procesadora, de esta forma, el acceso y modificación de los datos, correspondientes a cada camión, nodo y tipo de leche, se facilita.


```

struct Milk {
    float minimumQuota{};
    float profit{};
    char typeOfMilk{};
};

struct Node {
    float amountOfMilk{};
    int coordinateX{};
    int coordinateY{};
    int number{};
    Milk typeOfMilk{};
    bool visited{};
};

struct Truck {
    float amountOfMilk{};
    float capacity{};
    double cost{};
    int number{};
    vector<Node> route{};
    Milk typeOfMilk{};
    bool used{};
};

```

■ Milk:

- minimumQuota: representa la cuota mínima de leche que se debe cumplir, se utiliza para la restricción relacionada a las cuotas de la planta (12).
- profit: representa la ganancia que da la leche por unidad de volumen o cantidad, que se utiliza para la obtención de la ganancia final.
- typeOfMilk: representa el tipo de leche (A, B o C) al que corresponde, se utiliza para una mejor comparación e identificación de la leche que se encuentra en los nodos y en el camión.

■ Node:

- number: representa el número de identificación del nodo de entrega.
- coordinateX: representa la coordenada X de la ubicación del nodo, se utiliza para obtener el costo.
- coordinateY: representa la coordenada Y de la ubicación del nodo, se utiliza para obtener el costo.
- amountOfMilk: representa la cantidad de leche presente en el nodo, la cual se cargara en un camión una vez la granja sea visitada.
- typeOfMilk: representa el tipo de leche en forma de estructura Milk, lo que facilita el obtener información de la leche además de solo el tipo (A, B o C).
- visited: indica si el nodo ha sido visitado o no durante el recorrido de alguno de los camiones, se usa para restricciones como el que una granja solo pueda ser visitada por uno de los camiones y solo una vez, además es clave para saber cuando debe terminar el algoritmo de la solución inicial.

■ Truck:

- number: representa el número de identificación del camión.
- capacity: representa la capacidad máxima de leche que puede transportar el camión, se utiliza para no pasarse de la capacidad en relación con la cantidad de leche que se lleva en el camión.
- amountOfMilk: representa la cantidad de leche actualmente cargada en el camión.
- typeOfMilk: representa el tipo de leche en forma de estructura Milk, lo que facilita el obtener información de la leche además de solo el tipo (A, B o C).
- route: es una lista que contiene los nodos de entrega visitados por el camión durante su ruta.

- **cost:** representa el costo acumulado asociado a la ruta del camión.
- **used:** indica si el camión ha sido utilizado o no durante el recorrido de alguno de los camiones, se usa para restricciones como el que un camión solo puede hacer un recorrido, también, es esencial para saber cuando hay que cambiar el orden de los camiones en la solución inicial.

La instanciación de cada tipo de leche, nodo y camión, se almacena en vectores 'milks', 'nodes' y 'trucks' respectivamente, de forma que sea más fácil el recorrer cada uno de estos vectores a la hora de buscar una solución inicial.

- **Solución:** La solución corresponde a un vector de camiones(Truck), en los cuales se guarda la siguiente información importante:
 - **Ruta:** la ruta final del camión, desde que sale de la planta hasta que vuelve.
 - **Costo:** el costo final de recorrer la ruta.
 - **Tipo de leche:** el tipo de leche que lleva al final del recorrido.
 - **Cantidad de leche:** el volumen de leche con la que regresa a la planta.

Teniendo toda esta información se procede a calcular la ganancia por leche de cada camión, para posteriormente calcular la suma tanto de las ganancias por camión como del costo de cada ruta, para finalmente calcular la ganancia final, que corresponde a la suma de las ganancias por camión menos la suma de los costos de las rutas.

6. Descripción del algoritmo

Estructura General del Algoritmo:

- **Inicialización de Datos:** El algoritmo comienza leyendo datos desde un archivo de entrada, que contiene información sobre camiones, tipos de leche, nodos y sus coordenadas, capacidad de los camiones, cuotas mínimas de leche, y ganancias por tipo de leche. Estos datos se almacenan en estructuras de datos, como vectores.
- **Bucle Principal:** El algoritmo tiene un bucle principal que se ejecuta hasta que todos los nodos hayan sido visitados.
- **Selección de Camión:** En cada iteración del bucle principal, se selecciona aleatoriamente un camión que aún no ha sido utilizado. Este camión se usará para realizar rutas de entrega, siendo marcado como utilizado, para que no pueda realizar más de una ruta.
- **Greedy:** Se utiliza un enfoque "greedy" para determinar la ruta que tomará el camión seleccionado. Comenzando desde un nodo inicial (la planta procesadora), el algoritmo selecciona el siguiente nodo que maximiza la ganancia (ganancia por leche entregada menos el costo de viaje) y cumple con las restricciones de capacidad y cuota mínima de leche. Esta búsqueda continúa hasta que no se pueden agregar más nodos a la ruta, posteriormente, el camión regresa a la planta desde el nodo en que quedó en su ruta, sumándose también el costo de traslado.
- **Hill-Climbing AM:** Se utiliza la técnica reparadora Hill Climbing con alguna mejora para obtener una mejor solución a partir de la solución inicial entregada por Greedy, los movimientos utilizados para esta técnica serán:
 - 2-OPT para una ruta.
 - Swap entre nodos de una misma ruta, no necesariamente nodos adyacentes.

Se implementa de forma secuencial, es decir, primero se aplica Hill Climbing con 2-OPT y luego se aplica Hill Climbing con Swap. Se comienza con una solución inicial, se aplica el movimiento a la solución, hasta que el movimiento genere una solución mejor, posteriormente, se le aplica el movimiento a esa nueva solución, repitiéndose el proceso hasta que ya no se encuentren soluciones mejores, o hasta que se acaben las iteraciones.

- **Actualización de Datos:** Durante el enrutamiento, se actualiza la cantidad de leche en el camión, el tipo de leche transportada y el costo acumulado.
- **Marcado de Nodos Visitados:** Los nodos visitados durante el enrutamiento se eliminan del vector de nodos, para que no vuelvan a ser visitados.
- **Reinicio de Camiones y Nodos:** Cuando todos los camiones se han utilizado o no se pueden agregar más nodos a las rutas, se reinician los camiones y los nodos, listos para la siguiente iteración del bucle principal, es decir, se "vacía" la leche de todos los camiones y se marcan todos los nodos como **no visitados**.
- **Resultados y Estadísticas:** Después de que se han completado todas las rutas, se calculan estadísticas, como el costo total, la ganancia total y la diferencia entre ellas, lo que posteriormente se muestra en pantalla.

Componentes Específicos Implementados:

- **Selección de Tipo de Leche y calculo de ganancia:** El algoritmo selecciona el tipo de leche que maximiza la ganancia para cada nodo visitado. Si el camión ya lleva un tipo de leche, se verifica si cambiar a otro tipo sería más rentable.
- **Cálculo de Costo:** El costo entre dos nodos se calcula utilizando la distancia Euclidiana entre sus coordenadas. Este costo se utiliza para evaluar si agregar un nodo a la ruta es beneficioso.
- **Restricción de Capacidad:** Se verifica si agregar un nodo a la ruta no excede la capacidad del camión.
- **Restricción de Cuota Mínima de Leche:** Se verifica si agregar un nodo a la ruta cumple con las cuotas mínimas de leche para todos los tipos de leche, tomando en cuenta que las leches se pueden mezclar en la planta.

Algorithm 1: Greedy con orden aleatorio de camiones

Data: archivo
Result: camionesActualizados

```
1 LeerDatosDesdeArchivo(archivo);
2 InicializarCamiones();
3 InicializarTiposDeLeche();
4 InicializarNodos();
5 while ExistanNodosSinVisitar() do
6   | camiónActual ← SeleccionarCamiónAleatoriamente();
7   | InicializarRuta(camiónActual);
8   | MarcarNodoComoVisitado(nodoInicial);
9   | while PuedenAgregarNodosAlCamión(camiónActual) do
10  |   | nodoSiguiente ← SeleccionarMejorNodo(camiónActual);
11  |   | if CumpleTodasLasRestricciones(nodoSiguiente) then
12  |   |   | AgregarNodoARuta(camiónActual, nodoSiguiente);
13  |   |   | ActualizarCamión(camiónActual, nodoSiguiente);
14  |   |   | MarcarNodoComoVisitado(nodoSiguiente);
15 CalcularYMostrarResultados();
16 CalcularTiempoDeEjecución();
```

Algorithm 2: Hill Climbing con alguna mejora

Data: SoluciónInicial
Result: MejorSoluciónActual

```
1 MejorSoluciónActual ← SoluciónInicial
2 while QuedaVecindarioPorVisitar() do
3   | SoluciónCandidata ← Movimiento(MejorSoluciónActual);
4   | if SoluciónCandidata.Calidad > MejorSoluciónActual.Calidad then
5   |   | MejorSoluciónActual ← SoluciónCandidata
```

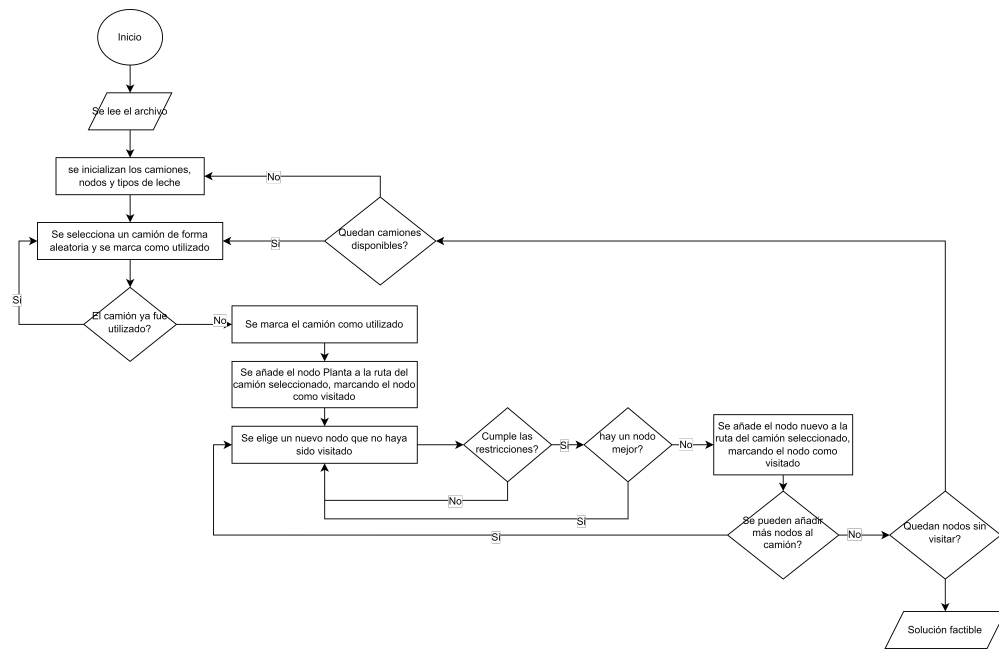


Figura 5: Diagrama de flujo Greedy

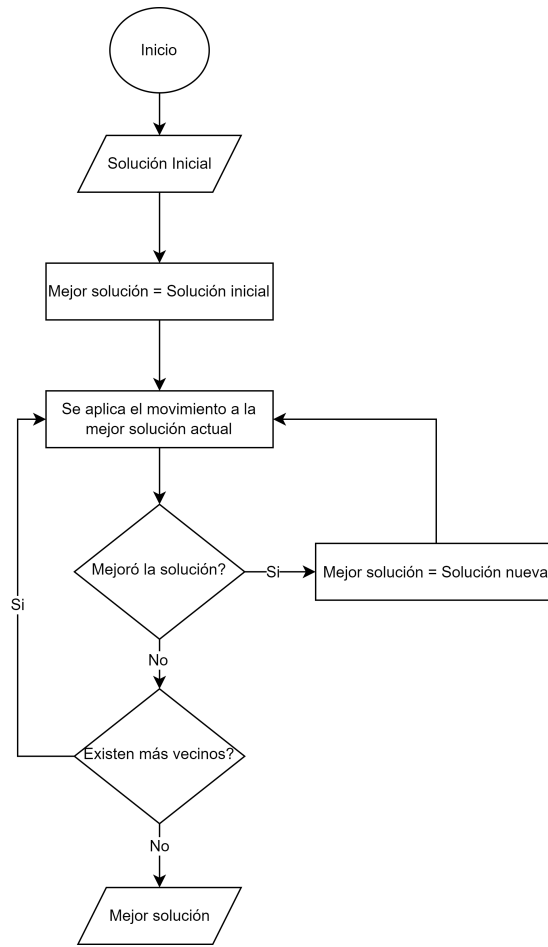


Figura 6: Diagrama de flujo Hill Climbing AM

7. Experimentos

Antes de experimentar, se necesita conocer los elementos o parámetros presentes en los algoritmos de búsqueda local, que se puedan modificar, el programa utilizado actualmente consta del orden de salida de los camiones, el criterio o función miope a la hora de encontrar una solución inicial con greedy y los movimientos que se pueden realizar al pasar la solución inicial por un algoritmo Hill Climbing con alguna mejora, empezando por el orden de los camiones, el cual, de forma simple, no es modificable, ya que, en el programa final se decidió utilizar todos los ordenes posibles, es decir, cuando hay cierto numero de camiones, mediante una función, se obtienen todas las permutaciones de orden posibles, lo que nos permite generar hasta $n!$ soluciones iniciales, siendo n el numero de camiones, además, no suele aumentar demasiado la complejidad mientras el número de camiones no sea muy grande, ahora, pasando a los parámetros que se pueden modificar, se tiene la función miope, para el algoritmo realizado, se definieron 2 criterios posibles, que la ganancia en leche de ir a un nodo sea la mayor entre todos los otros nodos que aún no se visitan, y que la diferencia entre la ganancia y el costo de ir a un nodo sea la mayor entre todos los nodos no visitados, considerando para ambos criterios, que se cumplan todas las restricciones planteadas con anterioridad, por lo que. En cuanto al Hill Climbing, los parametros a modificar serán los movimientos posibles, para lo cual se definieron 3 movimientos:

- 2-OPT para 2 rutas, es decir que un arco entre el nodo i y el nodo j de la ruta del camión k , y el arco entre el nodo h y el nodo g de la ruta del camión l , pasaran a cruzarse, quedando un arco entre el nodo i y g , y entre el nodo h y j , pasando el primera nueva ruta al camión k y la segunda al camión l .
- 2-OPT en una ruta, es decir que un arco entre el nodo i y el nodo j , y el arco entre el nodo h y el nodo g , todos de la misma ruta, pasaran a cruzarse, quedando un arco entre el nodo i y g , y entre el nodo h y j .
- Swap entre nodos de una ruta, se intercambian nodos de una ruta, no necesariamente tienen que ser nodos adyacentes.

Los experimentos se realizan con las instancias **a36**, **a44**, **eil33** y **eil76**, siendo está ultima la que más tiempo suele tomar, estos experimentos consistirán básicamente en ir modificando el criterio del algoritmo Greedy, a la vez que se modifican los movimientos utilizados para Hill Climbing, para los primeros 6 experimentos se utiliza el criterio mayor (Ganancia - Costo), es decir que el algoritmo greedy mira donde está parado, y se va al siguiente nodo que le asegure una mayor (Ganancia - Costo) en base al volumen de leche de la granja, al tipo de leche y a la distancia a la que está del nodo en el que se está actualmente.

- Experimento 1: Se utiliza un movimiento 2-OPT en 2 rutas y un Swap
- Experimento 2: Se utiliza un movimiento 2-OPT en 2 rutas y un 2-OPT en una ruta
- Experimento 3: Se utiliza un movimiento 2-OPT en 2 rutas
- Experimento 4: Se utiliza un movimiento 2-OPT en una ruta y un Swap
- Experimento 5: Se utiliza un movimiento 2-OPT en una ruta
- Experimento 6: Se utiliza un movimiento Swap

```

rilodeh@Rilodeh: /mnt/c/users/ / X + v
rilodeh@Rilodeh:/mnt/c/users/rilod/onedrive/escriptorio/MCPwB$
./MCPwB instances/a36.txt 1000
./MCPwB instances/a44.txt 1000
./MCPwB instances/eil33.txt 1000
./MCPwB instances/eil76.txt 1000

Archivo: instances/a36.txt

Solución inicial:

20211.8 1928.15 30140
1-11-35-8-32-26-29-23-5-17-14-20-2-1 765.311 16000 A
1-21-6-30-15-3-27-12-33-36-9-24-10-1 622.242 14200 B
1-13-19-10-4-22-25-31-28-16-7-34-1 540.6 14000 C

Solución después de Hill-Climbing:

29157.1 982.946 30140
1-23-14-32-20-5-35-29-26-2-17-11-8-1 372.271 16000 A
1-33-18-30-12-6-15-24-3-36-9-21-27-1 325.72 14200 B
1-25-34-31-19-22-28-13-4-7-10-16-1 284.955 14000 C

Tiempo total de ejecución: 0.191522[s]

Archivo: instances/a44.txt

Solución inicial:

37821.5 2058.46 39880
1-26-17-38-32-5-20-44-35-29-11-41-14-23-8-2-1 829.582 23400 A
1-28-31-7-37-34-43-12-4-13-40-27-19-39-10-16-25-22-1 667.563 17600 C
1-3-15-24-10-9-36-21-30-6-33-42-1 561.318 16000 B

Solución después de Hill-Climbing:

38620.7 1259.25 39880
1-32-20-35-5-26-41-11-38-17-2-44-14-23-29-8-1 466.995 23400 A
1-13-31-19-12-43-22-34-27-40-4-7-10-39-37-25-28-16-1 437.524 17600 C
1-3-42-15-30-24-36-21-33-6-18-9-1 354.735 16000 B

Tiempo total de ejecución: 0.059145[s]

Archivo: instances/eil33.txt

Solución inicial:

19713.5 1613.49 21327
1-8-32-26-23-5-29-11-2-17-14-20-1 649.488 11200 A
1-27-15-12-33-9-24-18-30-3-21-6-1 487.608 11690 B
1-31-25-19-10-20-16-4-13-22-7-1 476.393 6400 C

Solución después de Hill-Climbing:

20354 972.997 21327
1-29-17-26-23-20-2-32-14-11-8-5-1 355.806 11200 A
1-3-6-9-21-24-18-27-30-15-33-12-1 328.074 11690 B
1-31-28-25-22-19-16-10-7-13-4-1 289.117 6400 C

Tiempo total de ejecución: 0.136072[s]

Archivo: instances/eil76.txt

Solución inicial:

90094.6 2265.38 92360
1-67-10-22-34-25-31-46-49-76-7-28-55-13-40-58-61-37-19-52-4-64-43-70-16-73-1 669.767 42900 C
1-41-68-5-29-65-47-11-32-59-35-53-17-2-44-62-8-20-38-26-23-14-71-56-74-50-1 755.806 46800 A
1-12-15-33-3-57-39-60-54-51-21-6-18-48-63-27-45-9-42-30-75-69-36-66-24-72-1 839.804 46700 B

Solución después de Hill-Climbing:

91229.7 1130.28 92360
1-10-19-25-4-52-7-31-46-28-55-67-73-40-13-37-61-70-22-43-64-34-49-16-58-76-1 404.08 42900 C
1-41-17-50-26-56-32-11-59-14-38-71-62-29-5-47-8-20-53-35-68-2-44-65-23-74-1 389.8 46800 A
1-54-39-66-12-60-15-36-9-21-72-48-75-63-42-57-24-3-6-30-69-18-45-51-33-27-1 336.4 46700 B

Tiempo total de ejecución: 2.933947[s]

rilodeh@Rilodeh:/mnt/c/users/rilod/onedrive/escriptorio/MCPwB$

```

Figura 7: Experimento 1


```

rilodeh@Rilodeh: /mnt/c/users/rilod/onedrive/escritorio/MCPw$
./MCPwB instances/a36.txt 1000
./MCPwB instances/a44.txt 1000
./MCPwB instances/eil33.txt 1000
./MCPwB instances/eil76.txt 1000

Archivo: instances/a36.txt

Solución inicial:

28211.8    1928.15    30140
1-11-35-8-32-26-29-23-5-17-14-20-2-1    765.311    16000 A
1-21-6-30-15-3-27-12-33-36-9-24-18-1    622.242    14200 B
1-13-19-10-4-22-25-31-28-16-7-34-1    540.6    14000 C

Solución después de Hill-Climbing:

29113    1027.05    30140
1-11-8-26-29-35-5-20-32-14-23-2-17-1    342.315    16000 A
1-27-6-21-9-36-3-24-15-12-33-30-18-1    332.612    14200 B
1-16-10-28-25-22-19-31-13-4-7-34-1    352.124    14000 C

Tiempo total de ejecución: 0.150633[s]

Archivo: instances/a44.txt

Solución inicial:

37821.5    2058.46    39880
1-26-17-38-32-5-20-44-35-29-11-41-14-23-8-2-1    829.582    23400 A
1-28-31-7-37-34-43-12-4-13-40-27-19-39-10-16-25-22-1    667.563    17600 C
1-3-15-24-18-9-36-21-30-6-33-42-1    561.318    16000 B

Solución después de Hill-Climbing:

38592    1288.04    39880
1-32-5-35-23-14-44-26-41-2-17-38-11-20-29-8-1    416.09    23400 A
1-39-10-37-7-4-13-40-31-19-43-12-27-34-16-28-25-22-1    439.554    17600 C
1-3-15-30-36-21-24-18-9-6-33-42-1    432.392    16000 B

Tiempo total de ejecución: 0.051639[s]

Archivo: instances/eil33.txt

Solución inicial:

19713.5    1613.49    21327
1-31-25-19-10-28-16-4-13-22-7-1    476.393    6480 C
1-8-32-26-23-5-29-11-2-17-14-20-1    649.488    11200 A
1-27-15-12-33-9-24-18-30-3-21-6-1    487.608    11690 B

Solución después de Hill-Climbing:

20360    967.034    21327
1-4-13-7-10-19-22-25-28-16-31-1    276.676    6480 C
1-5-8-11-14-2-32-29-17-26-23-20-1    353.194    11200 A
1-3-12-33-9-18-15-30-27-24-21-6-1    337.164    11690 B

Tiempo total de ejecución: 0.120707[s]

Archivo: instances/eil76.txt

Solución inicial:

90094.6    2265.38    92360
1-67-10-22-34-25-31-46-49-76-7-28-55-13-40-58-61-37-19-52-4-64-43-70-16-73-1    669.767    42900 C
1-12-15-33-3-57-39-60-54-51-21-6-18-48-63-27-45-9-42-30-75-69-36-66-24-72-1    839.804    46700 B
1-41-68-5-29-65-47-11-32-59-35-53-17-2-44-62-8-20-38-26-23-14-71-56-74-50-1    755.806    46800 A

Solución después de Hill-Climbing:

91384.1    975.951    92360
1-76-46-28-55-58-16-61-37-70-22-49-31-7-52-4-34-64-43-25-19-10-40-73-67-13-1    325.689    42900 C
1-69-3-75-63-42-57-24-45-51-33-18-27-39-66-12-60-15-54-36-9-30-6-48-21-72-1    306.173    46700 B
1-5-68-41-26-56-32-11-59-8-20-47-35-53-14-38-71-62-29-23-65-44-2-17-74-50-1    344.089    46800 A

Tiempo total de ejecución: 2.664625[s]

rilodeh@Rilodeh: /mnt/c/users/rilod/onedrive/escritorio/MCPw$

```

Figura 8: Experimento 2

```
rilodeh@Rilodeh: /mnt/c/users/rilod/onedrive/escritorio/MCPw$
./MCPwB instances/a36.txt 1000
./MCPwB instances/a44.txt 1000
./MCPwB instances/eil33.txt 1000
./MCPwB instances/eil76.txt 1000

Archivo: instances/a36.txt

Solución inicial:

28211.8    1928.15    30140
1-11-35-8-32-26-29-23-5-17-14-20-2-1    765.311    16000 A
1-21-6-30-15-3-27-12-33-36-9-24-18-1    622.242    14200 B
1-13-19-10-4-22-25-31-28-16-7-34-1    540.6    14000 C

Solución después de Hill-Climbing:

28211.8    1928.15    30140
1-11-35-8-32-26-29-23-5-17-14-20-2-1    765.311    16000 A
1-21-6-30-15-3-27-12-33-36-9-24-18-1    622.242    14200 B
1-13-19-10-4-22-25-31-28-16-7-34-1    540.6    14000 C

Tiempo total de ejecución: 0.093139[s]

Archivo: instances/a44.txt

Solución inicial:

37821.5    2058.46    39880
1-26-17-38-32-5-20-44-35-29-11-41-14-23-8-2-1    829.582    23400 A
1-28-31-7-37-34-43-12-4-13-40-27-19-39-10-16-25-22-1    667.563    17600 C
1-3-15-24-18-9-36-21-30-6-33-42-1    561.318    16000 B

Solución después de Hill-Climbing:

37821.5    2058.46    39880
1-26-17-38-32-5-20-44-35-29-11-41-14-23-8-2-1    829.582    23400 A
1-28-31-7-37-34-43-12-4-13-40-27-19-39-10-16-25-22-1    667.563    17600 C
1-3-15-24-18-9-36-21-30-6-33-42-1    561.318    16000 B

Tiempo total de ejecución: 0.030516[s]

Archivo: instances/eil33.txt

Solución inicial:

19713.5    1613.49    21327
1-8-32-26-23-5-29-11-2-17-14-20-1    649.488    11200 A
1-27-15-12-33-9-24-18-30-3-21-6-1    487.608    11690 B
1-31-25-19-10-28-16-4-13-22-7-1    476.393    6480 C

Solución después de Hill-Climbing:

19713.5    1613.49    21327
1-8-32-26-23-5-29-11-2-17-14-20-1    649.488    11200 A
1-27-15-12-33-9-24-18-30-3-21-6-1    487.608    11690 B
1-31-25-19-10-28-16-4-13-22-7-1    476.393    6480 C

Tiempo total de ejecución: 0.075588[s]

Archivo: instances/eil76.txt

Solución inicial:

90094.6    2265.38    92360
1-67-10-22-34-25-31-46-49-76-7-28-55-13-40-58-61-37-19-52-4-64-43-70-16-73-1    669.767    42900 C
1-12-15-33-3-57-39-60-54-51-21-6-18-48-63-27-45-9-42-30-75-69-36-66-24-72-1    839.804    46700 B
1-41-68-5-29-65-47-11-32-59-35-53-17-2-44-62-8-20-38-26-23-14-71-56-74-50-1    755.806    46800 A

Solución después de Hill-Climbing:

90094.6    2265.38    92360
1-67-10-22-34-25-31-46-49-76-7-28-55-13-40-58-61-37-19-52-4-64-43-70-16-73-1    669.767    42900 C
1-12-15-33-3-57-39-60-54-51-21-6-18-48-63-27-45-9-42-30-75-69-36-66-24-72-1    839.804    46700 B
1-41-68-5-29-65-47-11-32-59-35-53-17-2-44-62-8-20-38-26-23-14-71-56-74-50-1    755.806    46800 A

Tiempo total de ejecución: 0.644706[s]

rilodeh@Rilodeh: /mnt/c/users/rilod/onedrive/escritorio/MCPw$
```

Figura 9: Experimento 3

```

rilodeh@Rilodeh: /mnt/c/users/rilod/onedrive/escritorio/MCPw$
./MCPwB instances/a36.txt 1000
./MCPwB instances/a44.txt 1000
./MCPwB instances/eil33.txt 1000
./MCPwB instances/eil76.txt 1000

Archivo: instances/a36.txt

Solución inicial:

28211.8    1928.15    30140
1-11-35-8-32-26-29-23-5-17-14-20-2-1    765.311    16000 A
1-21-6-30-15-3-27-12-33-36-9-24-18-1    622.242    14200 B
1-13-19-10-4-22-25-31-28-16-7-34-1    540.6    14000 C

Solución después de Hill-Climbing:

29204    935.997    30140
1-11-8-26-29-35-5-20-32-14-23-2-17-1    342.315    16000 A
1-27-6-21-9-36-3-24-15-30-18-33-12-1    298.99    14200 B
1-25-34-31-19-22-28-10-13-4-7-16-1    294.693    14000 C

Tiempo total de ejecución: 0.083749[s]

Archivo: instances/a44.txt

Solución inicial:

37821.5    2058.46    39880
1-26-17-38-32-5-20-44-35-29-11-41-14-23-8-2-1    829.582    23400 A
1-28-31-7-37-34-43-12-4-13-40-27-19-39-10-16-25-22-1    667.563    17600 C
1-3-15-24-18-9-36-21-30-6-33-42-1    561.318    16000 B

Solución después de Hill-Climbing:

38659    1221.02    39880
1-32-5-35-23-14-44-26-41-2-17-38-11-20-29-8-1    416.09    23400 A
1-39-10-37-7-4-13-40-31-19-12-43-22-28-16-25-34-27-1    415.533    17600 C
1-42-15-30-36-21-24-18-33-6-9-3-1    389.4    16000 B

Tiempo total de ejecución: 0.033707[s]

Archivo: instances/eil33.txt

Solución inicial:

19713.5    1613.49    21327
1-31-25-19-10-28-16-4-13-22-7-1    476.393    6480 C
1-27-15-12-33-9-24-18-30-3-21-6-1    487.608    11690 B
1-8-32-26-23-5-29-11-2-17-14-20-1    649.488    11200 A

Solución después de Hill-Climbing:

20381.8    945.234    21327
1-4-13-7-10-19-22-25-28-16-31-1    276.676    6480 C
1-3-15-30-27-10-24-21-9-33-12-6-1    318.959    11690 B
1-8-11-20-23-26-17-29-32-2-14-5-1    349.599    11200 A

Tiempo total de ejecución: 0.066074[s]

Archivo: instances/eil76.txt

Solución inicial:

90094.6    2265.38    92360
1-67-10-22-34-25-31-46-49-76-7-28-55-13-40-58-61-37-19-52-4-64-43-70-16-73-1    669.767    42900 C
1-41-68-5-29-65-47-11-32-59-35-53-17-2-44-62-8-20-38-26-23-14-71-56-74-50-1    755.806    46800 A
1-12-15-33-3-57-39-60-54-51-21-6-18-48-63-27-45-9-42-30-75-69-36-66-24-72-1    839.804    46700 B

Solución después de Hill-Climbing:

91410.5    949.52    92360
1-76-46-28-55-58-16-61-37-70-22-49-31-7-52-4-34-64-43-25-19-10-40-73-67-13-1    325.689    42900 C
1-5-68-41-26-56-32-11-59-8-20-47-35-53-14-38-71-62-29-23-65-44-2-74-50-17-1    322.587    46800 A
1-3-63-75-48-72-21-6-30-9-36-54-15-60-12-66-39-27-18-33-51-45-24-57-42-69-1    301.245    46700 B

Tiempo total de ejecución: 2.247425[s]

rilodeh@Rilodeh: /mnt/c/users/rilod/onedrive/escritorio/MCPw$

```

Figura 10: Experimento 4

```

rilodeh@Rilodeh: /mnt/c/users/rilod/onedrive/escriptorio/MCPwB$
./MCPwB instances/a36.txt 1000
./MCPwB instances/a44.txt 1000
./MCPwB instances/eil33.txt 1000
./MCPwB instances/eil76.txt 1000

Archivo: instances/a36.txt

Solución inicial:

28211.8    1928.15    30140
1-11-35-8-32-26-29-23-5-17-14-20-2-1    765.311    16000 A
1-21-6-30-15-3-27-12-33-36-9-24-18-1    622.242    14200 B
1-13-19-10-4-22-25-31-28-16-7-34-1    540.6    14000 C

Solución después de Hill-Climbing:

29113    1027.05    30140
1-11-8-26-29-35-5-20-32-14-23-2-17-1    342.315    16000 A
1-27-6-21-9-36-3-24-15-12-33-30-18-1    332.612    14200 B
1-16-10-28-25-22-19-31-13-4-7-34-1    352.124    14000 C

Tiempo total de ejecución: 0.061640[s]

Archivo: instances/a44.txt

Solución inicial:

37821.5    2058.46    39880
1-26-17-38-32-5-20-44-35-29-11-41-14-23-8-2-1    829.582    23400 A
1-28-31-7-37-34-43-12-4-13-40-27-19-39-10-16-25-22-1    667.563    17600 C
1-3-15-24-18-9-36-21-30-6-33-42-1    561.318    16000 B

Solución después de Hill-Climbing:

38592    1288.04    39880
1-32-5-35-23-14-44-26-41-2-17-38-11-20-29-8-1    416.09    23400 A
1-39-10-37-7-4-13-40-31-19-43-12-27-34-16-28-25-22-1    439.554    17600 C
1-3-15-30-36-21-24-18-9-6-33-42-1    432.392    16000 B

Tiempo total de ejecución: 0.025797[s]

Archivo: instances/eil33.txt

Solución inicial:

19713.5    1613.49    21327
1-31-25-19-10-28-16-4-13-22-7-1    476.393    6400 C
1-8-32-26-23-5-29-11-2-17-14-20-1    649.488    11200 A
1-27-15-12-33-9-24-18-30-3-21-6-1    487.608    11690 B

Solución después de Hill-Climbing:

20360    967.034    21327
1-4-13-7-10-19-22-25-28-16-31-1    276.676    6400 C
1-5-8-11-14-2-32-29-17-26-23-20-1    353.194    11200 A
1-3-12-33-9-18-15-30-27-24-21-6-1    337.164    11690 B

Tiempo total de ejecución: 0.041668[s]

Archivo: instances/eil76.txt

Solución inicial:

90094.6    2265.30    92360
1-67-10-22-34-25-31-46-49-76-7-28-55-13-40-50-61-37-19-52-4-64-43-70-16-73-1    669.767    42900 C
1-12-15-33-3-57-39-60-54-51-21-6-18-48-63-27-45-9-42-30-75-69-36-66-24-72-1    839.804    46700 B
1-41-68-5-29-65-47-11-32-59-35-53-17-2-44-62-8-20-38-26-23-14-71-56-74-50-1    755.806    46800 A

Solución después de Hill-Climbing:

91384.1    975.951    92360
1-76-46-28-55-58-16-61-37-70-22-49-31-7-52-4-34-64-43-25-19-10-40-73-67-13-1    325.689    42900 C
1-69-3-75-63-42-57-24-45-51-33-18-27-39-66-12-60-15-54-36-9-30-6-48-21-72-1    306.173    46700 B
1-5-68-41-26-56-32-11-59-8-20-47-35-53-14-38-71-62-29-23-65-44-2-17-74-50-1    344.089    46800 A

Tiempo total de ejecución: 2.038468[s]
rilodeh@Rilodeh: /mnt/c/users/rilod/onedrive/escriptorio/MCPwB$

```

Figura 11: Experimento 5

```

rilodeh@Rilodeh: /mnt/c/users/rilod/onedrive/escritorio/MCPwB$
rilodeh@Rilodeh: /mnt/c/users/rilod/onedrive/escritorio/MCPwB$
./MCPwB instances/a36.txt 1000
./MCPwB instances/a44.txt 1000
./MCPwB instances/eil33.txt 1000
./MCPwB instances/eil76.txt 1000

Archivo: instances/a36.txt

Solución inicial:
28211.8 1928.15 30140
1-11-35-8-32-26-29-23-5-17-14-20-2-1 765.311 16000 A
1-21-6-30-15-3-27-12-33-36-9-24-18-1 622.242 14200 B
1-13-19-10-4-22-25-31-28-16-7-34-1 540.6 14000 C

Solución después de Hill-Climbing:
29157.1 982.946 30140
1-23-14-32-20-5-35-29-26-2-17-11-8-1 372.271 16000 A
1-33-18-30-12-6-15-24-3-36-9-21-27-1 325.72 14200 B
1-25-34-31-19-22-28-13-4-7-10-16-1 284.955 14000 C

Tiempo total de ejecución: 0.102351[s]

Archivo: instances/a44.txt

Solución inicial:
37821.5 2058.46 39880
1-26-17-38-32-5-20-44-35-29-11-41-14-23-8-2-1 829.582 23400 A
1-28-31-7-37-34-43-12-4-13-40-27-19-39-10-16-25-22-1 667.563 17600 C
1-3-15-24-18-9-36-21-30-6-33-42-1 561.318 16000 B

Solución después de Hill-Climbing:
38620.7 1259.25 39880
1-32-20-35-5-26-41-11-38-17-2-44-14-23-29-8-1 466.995 23400 A
1-13-31-19-12-43-22-34-27-40-4-7-10-39-37-25-28-16-1 437.524 17600 C
1-3-42-15-30-24-36-21-33-6-18-9-1 354.735 16000 B

Tiempo total de ejecución: 0.031530[s]

Archivo: instances/eil33.txt

Solución inicial:
19713.5 1613.49 21327
1-8-32-26-23-5-20-11-2-17-14-20-1 649.488 11200 A
1-27-15-12-33-9-24-18-30-3-21-6-1 487.608 11690 B
1-31-25-19-10-28-16-4-13-22-7-1 476.393 6480 C

Solución después de Hill-Climbing:
20354 972.997 21327
1-29-17-26-23-20-2-32-14-11-8-5-1 355.806 11200 A
1-3-6-9-21-24-18-27-30-15-33-12-1 328.074 11690 B
1-31-28-25-22-19-16-10-7-13-4-1 289.117 6480 C

Tiempo total de ejecución: 0.070696[s]

Archivo: instances/eil76.txt

Solución inicial:
90094.6 2265.38 92360
1-67-10-22-34-25-31-46-49-76-7-28-55-13-40-58-61-37-19-52-4-64-43-70-16-73-1 669.767 42900 C
1-12-15-33-3-57-39-60-54-51-21-6-18-48-63-27-45-9-42-30-75-69-36-66-24-72-1 839.804 46700 B
1-41-68-5-29-65-47-11-32-59-35-53-17-2-44-62-8-20-38-26-23-14-71-56-74-50-1 755.806 46800 A

Solución después de Hill-Climbing:
91229.7 1130.28 92360
1-10-19-25-4-52-7-31-46-28-55-67-73-40-13-37-61-70-22-43-64-34-49-16-58-76-1 404.08 42900 C
1-54-39-66-12-60-15-36-9-21-72-48-75-63-42-57-24-3-6-30-69-18-45-51-33-27-1 336.4 46700 B
1-41-17-50-26-56-32-11-59-14-38-71-62-29-5-47-8-20-53-35-68-2-44-65-23-74-1 389.8 46800 A

Tiempo total de ejecución: 2.302419[s]
rilodeh@Rilodeh: /mnt/c/users/rilod/onedrive/escritorio/MCPwB$

```

Figura 12: Experimento 6

Para los ultimos 6 experimentos se utiliza el criterio mayor (Ganancia), es decir que el algoritmo greedy ya no se fija en la distancia a la que está de los nodos vecinos, simplemente se va al siguiente nodo que le asegure una mayor ganancia en base al volumen de leche de la granja y al tipo de leche, considerando que, si la leche de dicha granja es de menor calidad de la que lleva, entonces su leche disminuirá su calidad.

- Experimento 7: Se utiliza un movimiento 2-OPT en 2 rutas y un Swap
- Experimento 8: Se utiliza un movimiento 2-OPT en 2 rutas y un 2-OPT en una ruta

- Experimento 9: Se utiliza un movimiento 2-OPT en 2 rutas
- Experimento 10: Se utiliza un movimiento 2-OPT en una ruta y un Swap
- Experimento 11: Se utiliza un movimiento 2-OPT en una ruta
- Experimento 12: Se utiliza un movimiento Swap

```

rilodeh@Rilodeh: /mnt/c/asei
rilodeh@Rilodeh:/mnt/c/users/rilod/onedrive/escriptorio/MCPw$
./MCPwB instances/a36.txt 1000
./MCPwB instances/a44.txt 1000
./MCPwB instances/eil33.txt 1000
./MCPwB instances/eil76.txt 1000

Archivo: instances/a36.txt

Solución inicial:
28192.3 1947.69 30140
1-11-35-8-32-26-29-5-23-17-14-20-2-1 709.366 16000 A
1-21-6-30-3-15-27-12-33-36-9-24-18-1 635.891 14200 B
1-13-19-18-4-22-25-31-16-28-7-34-1 602.435 14000 C

Solución después de Hill-Climbing:
29113 1027.03 30140
1-11-17-2-32-5-20-14-23-26-29-35-8-1 416.353 16000 A
1-33-18-30-12-6-15-24-3-36-9-21-27-1 325.72 14200 B
1-25-34-31-19-22-28-13-4-7-10-16-1 284.955 14000 C

Tiempo total de ejecución: 0.204035[s]

Archivo: instances/a44.txt

Solución inicial:
37565.1 2314.95 39880
1-17-26-38-32-5-20-44-11-29-35-14-41-8-23-2-1 930.737 23400 A
1-28-31-7-37-43-34-4-12-13-19-27-40-16-42-10-25-22-1 806.645 17600 C
1-3-15-24-9-18-36-21-30-6-33-39-1 577.566 16000 B

Solución después de Hill-Climbing:
38589.4 1290.61 39880
1-35-41-2-44-14-23-26-17-38-11-20-29-8-32-5-1 480.662 23400 A
1-34-22-43-12-27-4-7-13-40-31-19-18-37-42-25-28-16-1 454.775 17600 C
1-3-15-39-30-24-36-21-33-6-18-9-1 355.176 16000 B

Tiempo total de ejecución: 0.071143[s]

Archivo: instances/eil33.txt

Solución inicial:
19696.2 1630.78 21327
1-8-32-26-23-5-29-11-2-17-14-20-1 649.488 11200 A
1-27-15-12-33-9-24-18-30-3-21-6-1 487.608 11690 B
1-31-25-19-10-28-16-4-22-13-7-1 493.681 6480 C

Solución después de Hill-Climbing:
20354 972.997 21327
1-29-17-26-23-20-2-32-14-11-8-5-1 355.806 11200 A
1-3-6-9-21-24-18-27-30-15-33-12-1 328.074 11690 B
1-31-28-25-22-19-16-10-7-13-4-1 289.117 6480 C

Tiempo total de ejecución: 0.141688[s]

Archivo: instances/eil76.txt

Solución inicial:
89816.5 2543.49 92360
1-67-10-22-25-34-31-46-49-76-7-28-13-40-55-58-19-61-37-52-4-43-64-16-70-73-1 756.193 42900 C
1-41-5-68-29-65-47-11-32-59-17-35-53-2-44-8-20-62-26-38-14-23-71-56-74-50-1 827.291 46800 A
1-12-15-33-3-57-39-60-21-51-54-6-18-48-27-63-45-9-42-30-36-69-75-66-24-72-1 960.002 46700 B

Solución después de Hill-Climbing:
91214.3 1145.74 92360
1-7-34-64-43-49-16-58-46-28-55-67-13-76-31-22-70-61-37-52-4-25-19-10-40-73-1 389.911 42900 C
1-41-17-50-26-56-32-11-59-8-20-53-47-35-68-5-29-62-71-38-14-2-44-65-23-74-1 374.702 46800 A
1-15-60-12-66-39-51-33-54-36-9-21-72-6-48-75-63-42-57-24-45-3-30-69-18-27-1 381.123 46700 B

Tiempo total de ejecución: 3.300414[s]
rilodeh@Rilodeh:/mnt/c/users/rilod/onedrive/escriptorio/MCPw$

```

Figura 13: Experimento 7

```

rilodeh@Rilodeh: /mnt/c/users/rilodeh/onedrive/escritorio/MCPw$
./MCPwB instances/a36.txt 1000
./MCPwB instances/a44.txt 1000
./MCPwB instances/eil33.txt 1000
./MCPwB instances/eil76.txt 1000

Archivo: instances/a36.txt

Solución inicial:

28192.3    1947.69    30140
1-11-35-8-32-26-29-5-23-17-14-20-2-1    709.366    16000 A
1-21-6-30-3-15-27-12-33-36-9-24-18-1    635.891    14200 B
1-13-19-10-4-22-25-31-16-28-7-34-1    602.435    14000 C

Solución después de Hill-Climbing:

29113    1027.05    30140
1-11-8-26-29-35-5-20-32-14-23-2-17-1    342.315    16000 A
1-27-6-21-9-36-3-24-15-12-33-30-18-1    332.612    14200 B
1-16-10-28-25-22-19-31-13-4-7-34-1    352.124    14000 C

Tiempo total de ejecución: 0.155963[s]

Archivo: instances/a44.txt

Solución inicial:

37565.1    2314.95    39880
1-17-26-38-32-5-20-44-11-29-35-14-41-8-23-2-1    930.737    23400 A
1-28-31-7-37-43-34-4-12-13-19-27-40-16-42-10-25-22-1    806.645    17600 C
1-3-15-24-9-18-36-21-30-6-33-39-1    577.566    16000 B

Solución después de Hill-Climbing:

38576    1303.99    39880
1-23-14-44-26-41-2-17-38-11-35-5-20-29-8-32-1    408.39    23400 A
1-42-10-37-7-4-13-40-31-19-43-12-27-34-16-28-25-22-1    434.34    17600 C
1-3-15-30-24-36-21-18-9-6-33-39-1    461.259    16000 B

Tiempo total de ejecución: 0.060502[s]

Archivo: instances/eil33.txt

Solución inicial:

19696.2    1630.78    21327
1-31-25-19-10-28-16-4-22-13-7-1    493.681    6480 C
1-8-32-26-23-5-29-11-2-17-14-20-1    649.488    11200 A
1-27-15-12-33-9-24-18-30-3-21-6-1    487.608    11690 B

Solución después de Hill-Climbing:

20360    967.034    21327
1-4-13-7-10-19-22-25-28-16-31-1    276.676    6480 C
1-5-0-11-14-2-32-29-17-26-23-20-1    353.194    11200 A
1-3-12-33-9-18-15-30-27-24-21-6-1    337.164    11690 B

Tiempo total de ejecución: 0.117638[s]

Archivo: instances/eil76.txt

Solución inicial:

89816.5    2543.49    92360
1-67-10-22-25-34-31-46-49-76-7-28-13-40-55-58-19-61-37-52-4-43-64-16-70-73-1    756.193    42900 C
1-12-15-33-3-57-39-60-21-51-54-6-18-48-27-63-45-9-42-30-36-69-75-66-24-72-1    960.002    46700 B
1-41-5-68-29-65-47-11-32-59-17-35-53-2-44-8-20-62-26-38-14-23-71-56-74-50-1    827.291    46800 A

Solución después de Hill-Climbing:

91378.3    981.685    92360
1-13-67-73-40-10-19-25-4-52-64-43-34-7-31-49-22-70-37-61-16-58-55-28-46-76-1    320.43    42900 C
1-69-3-75-63-42-57-24-51-33-45-18-27-39-66-12-60-15-54-36-9-30-6-48-21-72-1    305.632    46700 B
1-5-68-35-47-53-38-71-14-20-8-41-59-11-32-56-26-17-2-44-65-23-62-29-74-50-1    355.622    46800 A

Tiempo total de ejecución: 3.046844[s]

rilodeh@Rilodeh: /mnt/c/users/rilodeh/onedrive/escritorio/MCPw$

```

Figura 14: Experimento 8

```
rilodeh@Rilodeh: /mnt/c/users/ X + ~
rilodeh@Rilodeh: /mnt/c/users/rilod/onedrive/escriptorio/MCPwB$
./MCPwB instances/a36.txt 1000
./MCPwB instances/a44.txt 1000
./MCPwB instances/eil33.txt 1000
./MCPwB instances/eil76.txt 1000

Archivo: instances/a36.txt

Solución inicial:

28192.3 1947.69 30140
1-11-35-8-32-26-29-5-23-17-14-20-2-1 709.366 16000 A
1-21-6-30-3-15-27-12-33-36-9-24-18-1 635.891 14200 B
1-13-19-10-4-22-25-31-16-28-7-34-1 602.435 14000 C

Solución después de Hill-Climbing:

28192.3 1947.69 30140
1-11-35-8-32-26-29-5-23-17-14-20-2-1 709.366 16000 A
1-21-6-30-3-15-27-12-33-36-9-24-18-1 635.891 14200 B
1-13-19-10-4-22-25-31-16-28-7-34-1 602.435 14000 C

Tiempo total de ejecución: 0.093763[s]

Archivo: instances/a44.txt

Solución inicial:

37565.1 2314.95 39880
1-17-26-38-32-5-20-44-11-29-35-14-41-8-23-2-1 930.737 23400 A
1-28-31-7-37-43-34-4-12-13-19-27-40-16-42-10-25-22-1 806.645 17600 C
1-3-15-24-9-18-36-21-30-6-33-39-1 577.566 16000 B

Solución después de Hill-Climbing:

37565.1 2314.95 39880
1-17-26-38-32-5-20-44-11-29-35-14-41-8-23-2-1 930.737 23400 A
1-28-31-7-37-43-34-4-12-13-19-27-40-16-42-10-25-22-1 806.645 17600 C
1-3-15-24-9-18-36-21-30-6-33-39-1 577.566 16000 B

Tiempo total de ejecución: 0.030392[s]

Archivo: instances/eil33.txt

Solución inicial:

19696.2 1630.78 21327
1-8-32-26-23-5-29-11-2-17-14-20-1 649.488 11200 A
1-27-15-12-33-9-24-18-30-3-21-6-1 487.608 11690 B
1-31-25-19-10-28-16-4-22-13-7-1 493.681 6480 C

Solución después de Hill-Climbing:

19696.2 1630.78 21327
1-8-32-26-23-5-29-11-2-17-14-20-1 649.488 11200 A
1-27-15-12-33-9-24-18-30-3-21-6-1 487.608 11690 B
1-31-25-19-10-28-16-4-22-13-7-1 493.681 6480 C

Tiempo total de ejecución: 0.073582[s]

Archivo: instances/eil76.txt

Solución inicial:

89816.5 2543.49 92360
1-67-10-22-25-34-31-46-49-76-7-28-13-40-55-58-19-61-37-52-4-43-64-16-70-73-1 756.193 42900 C
1-41-5-68-29-65-47-11-32-59-17-35-53-2-44-8-20-62-26-38-14-23-71-56-74-50-1 827.291 46800 A
1-12-15-33-3-57-39-60-21-51-54-6-18-48-27-63-45-9-42-30-36-69-75-66-24-72-1 960.002 46700 B

Solución después de Hill-Climbing:

89816.5 2543.49 92360
1-67-10-22-25-34-31-46-49-76-7-28-13-40-55-58-19-61-37-52-4-43-64-16-70-73-1 756.193 42900 C
1-41-5-68-29-65-47-11-32-59-17-35-53-2-44-8-20-62-26-38-14-23-71-56-74-50-1 827.291 46800 A
1-12-15-33-3-57-39-60-21-51-54-6-18-48-27-63-45-9-42-30-36-69-75-66-24-72-1 960.002 46700 B

Tiempo total de ejecución: 0.640898[s]

rilodeh@Rilodeh: /mnt/c/users/rilod/onedrive/escriptorio/MCPwB$
```

Figura 15: Experimento 9


```

rilodeh@Rilodeh: /mnt/c/users/rilod/onedrive/escritorio/MCPwB$
./MCPwB instances/a36.txt 1000
./MCPwB instances/a44.txt 1000
./MCPwB instances/eil33.txt 1000
./MCPwB instances/eil76.txt 1000

Archivo: instances/a36.txt

Solución inicial:

28192.3    1947.69    30140
1-11-35-8-32-26-29-5-23-17-14-20-2-1    709.366    16000 A
1-13-19-10-4-22-25-31-16-28-7-34-1    602.435    14000 C
1-21-6-30-3-15-27-12-33-36-9-24-18-1    635.891    14200 B

Solución después de Hill-Climbing:

29210.8    929.221    30140
1-11-8-26-29-35-5-20-32-14-23-2-17-1    342.315    16000 A
1-16-10-7-4-13-28-25-22-19-31-34-1    287.917    14000 C
1-12-33-18-30-15-24-3-36-9-21-6-27-1    298.99    14200 B

Tiempo total de ejecución: 0.095346[s]

Archivo: instances/a44.txt

Solución inicial:

37565.1    2314.95    39880
1-17-26-38-32-5-20-44-11-29-35-14-41-8-23-2-1    930.737    23400 A
1-28-31-7-37-43-34-4-12-13-19-27-40-16-42-10-25-22-1    806.645    17600 C
1-3-15-24-9-18-36-21-30-6-33-39-1    577.566    16000 B

Solución después de Hill-Climbing:

38659.4    1220.56    39880
1-23-14-44-26-41-2-17-38-11-35-5-20-29-8-32-1    408.39    23400 A
1-42-10-37-7-4-13-40-31-19-12-43-22-28-16-25-34-27-1    410.319    17600 C
1-15-39-30-24-36-21-18-33-6-9-3-1    401.852    16000 B

Tiempo total de ejecución: 0.043555[s]

Archivo: instances/eil33.txt

Solución inicial:

19696.2    1630.78    21327
1-31-25-19-10-28-16-4-22-13-7-1    493.681    6480 C
1-27-15-12-33-9-24-18-30-3-21-6-1    487.608    11690 B
1-8-32-26-23-5-29-11-2-17-14-20-1    649.488    11200 A

Solución después de Hill-Climbing:

20381.8    945.234    21327
1-4-13-7-10-19-22-25-28-16-31-1    276.676    6480 C
1-3-15-30-27-18-24-21-9-33-12-6-1    318.959    11690 B
1-8-11-20-23-26-17-29-32-2-14-5-1    349.599    11200 A

Tiempo total de ejecución: 0.061481[s]

Archivo: instances/eil76.txt

Solución inicial:

89816.5    2543.49    92360
1-67-10-22-25-34-31-46-49-76-7-28-13-40-55-58-19-61-37-52-4-43-64-16-70-73-1    756.193    42900 C
1-12-15-33-3-87-39-60-21-51-54-6-18-48-27-63-45-9-42-30-36-69-75-66-24-72-1    960.002    46700 B
1-41-5-68-29-65-47-11-32-59-17-35-53-2-44-8-20-62-26-38-14-23-71-56-74-50-1    827.291    46800 A

Solución después de Hill-Climbing:

91481    959.017    92360
1-13-67-73-40-10-19-25-4-52-64-43-34-7-31-49-22-70-37-61-16-58-55-28-46-76-1    320.43    42900 C
1-69-3-75-63-42-57-24-51-33-45-18-27-39-66-12-60-15-54-36-9-38-48-72-21-6-1    298.161    46700 B
1-5-68-35-47-53-38-71-14-20-8-41-59-11-32-56-26-50-17-44-65-23-62-29-2-74-1    340.425    46800 A

Tiempo total de ejecución: 2.662531[s]
rilodeh@Rilodeh: /mnt/c/users/rilod/onedrive/escritorio/MCPwB$

```

Figura 16: Experimento 10

```

rilodeh@Rilodeh: /mnt/c/users/rilod/onedrive/escritorio/MCPw$
./MCPwB instances/a36.txt 1000
./MCPwB instances/a44.txt 1000
./MCPwB instances/eil33.txt 1000
./MCPwB instances/eil76.txt 1000

Archivo: instances/a36.txt

Solución inicial:

28192.3    1947.69    30140
1-11-35-8-32-26-29-5-23-17-14-20-2-1    709.366    16000 A
1-21-6-30-3-15-27-12-33-36-9-24-18-1    635.891    14200 B
1-13-19-10-4-22-25-31-16-28-7-34-1    602.435    14000 C

Solución después de Hill-Climbing:

29113    1027.05    30140
1-11-8-26-29-35-5-20-32-14-23-2-17-1    342.315    16000 A
1-27-6-21-9-36-3-24-15-12-33-30-18-1    332.612    14200 B
1-16-10-28-25-22-19-31-13-4-7-34-1    352.124    14000 C

Tiempo total de ejecución: 0.056257[s]

Archivo: instances/a44.txt

Solución inicial:

37565.1    2314.95    39880
1-17-26-38-32-5-20-44-11-29-35-14-41-8-23-2-1    930.737    23400 A
1-28-31-7-37-43-34-4-12-13-19-27-40-16-42-10-25-22-1    806.645    17600 C
1-3-15-24-9-18-36-21-30-6-33-39-1    577.566    16000 B

Solución después de Hill-Climbing:

38576    1303.99    39880
1-23-14-44-26-41-2-17-38-11-35-5-20-29-8-32-1    408.39    23400 A
1-42-10-37-7-4-13-40-31-19-43-12-27-34-16-28-25-22-1    434.34    17600 C
1-3-15-30-24-36-21-18-9-6-33-39-1    461.259    16000 B

Tiempo total de ejecución: 0.035195[s]

Archivo: instances/eil33.txt

Solución inicial:

19696.2    1630.78    21327
1-31-25-19-10-28-16-4-22-13-7-1    493.681    6480 C
1-8-32-26-23-5-29-11-2-17-14-20-1    649.488    11200 A
1-27-15-12-33-9-24-18-30-3-21-6-1    487.608    11690 B

Solución después de Hill-Climbing:

20360    967.034    21327
1-4-13-7-10-19-22-25-28-16-31-1    276.676    6480 C
1-5-8-11-14-2-32-20-17-26-23-20-1    353.194    11200 A
1-3-12-33-9-18-15-30-27-24-21-6-1    337.164    11690 B

Tiempo total de ejecución: 0.049885[s]

Archivo: instances/eil76.txt

Solución inicial:

89816.5    2543.49    92360
1-67-10-22-25-34-31-46-49-76-7-28-13-40-55-58-19-61-37-52-4-43-64-16-70-73-1    756.193    42900 C
1-12-15-33-3-57-39-60-21-51-54-6-18-48-27-63-45-9-42-30-36-69-75-66-24-72-1    960.082    46700 B
1-41-5-68-29-65-47-11-32-59-17-35-53-2-44-8-20-62-26-38-14-23-71-56-74-50-1    827.291    46800 A

Solución después de Hill-Climbing:

91378.3    981.685    92360
1-13-67-73-40-10-19-25-4-52-64-43-34-7-31-49-22-70-37-61-16-58-55-28-46-76-1    320.43    42900 C
1-69-3-75-63-42-57-24-51-33-45-18-27-39-66-12-60-15-54-36-9-30-6-48-21-72-1    305.632    46700 B
1-5-68-35-47-53-38-71-14-20-8-41-59-11-32-56-26-17-2-44-65-23-62-29-74-50-1    355.622    46800 A

Tiempo total de ejecución: 2.424380[s]

rilodeh@Rilodeh: /mnt/c/users/rilod/onedrive/escritorio/MCPw$

```

Figura 17: Experimento 11

```
rilodeh@Rilodeh: /mnt/c/users/ X + -
rilodeh@Rilodeh:/mnt/c/users/rilod/onedrive/escritorio/MCPwB$
./MCPwB instances/a36.txt 1000
./MCPwB instances/a44.txt 1000
./MCPwB instances/eil33.txt 1000
./MCPwB instances/eil76.txt 1000

Archivo: instances/a36.txt
Solución inicial:
28192.3 1947.69 30140
1-11-35-8-32-26-29-5-23-17-14-20-2-1 709.366 16000 A
1-21-6-30-3-15-27-12-33-36-9-24-18-1 635.891 14200 B
1-13-19-10-4-22-25-31-16-28-7-34-1 602.435 14000 C

Solución después de Hill-Climbing:
29113 1027.03 30140
1-11-17-2-32-5-20-14-23-26-29-35-8-1 416.353 16000 A
1-33-18-30-12-6-15-24-3-36-9-21-27-1 325.72 14200 B
1-25-34-31-19-22-28-13-4-7-10-16-1 284.955 14000 C

Tiempo total de ejecución: 0.114010[s]

Archivo: instances/a44.txt
Solución inicial:
37565.1 2314.95 39880
1-17-26-38-32-5-20-44-11-20-35-14-41-8-23-2-1 930.737 23400 A
1-28-31-7-37-43-34-4-12-13-19-27-40-16-42-10-25-22-1 806.645 17600 C
1-3-15-24-9-18-36-21-30-6-33-39-1 577.566 16000 B

Solución después de Hill-Climbing:
38589.4 1290.61 39880
1-35-41-2-44-14-23-26-17-38-11-20-29-8-32-5-1 480.662 23400 A
1-34-22-43-12-27-4-7-13-40-31-19-10-37-42-25-28-16-1 454.775 17600 C
1-3-15-39-30-24-36-21-33-6-18-9-1 355.176 16000 B

Tiempo total de ejecución: 0.042046[s]

Archivo: instances/eil33.txt
Solución inicial:
19696.2 1630.78 21327
1-8-32-26-23-5-29-11-2-17-14-20-1 649.488 11200 A
1-27-15-12-33-9-24-18-30-3-21-6-1 487.608 11690 B
1-31-25-19-10-28-16-4-22-13-7-1 493.681 6480 C

Solución después de Hill-Climbing:
20354 972.997 21327
1-29-17-26-23-20-2-32-14-11-8-5-1 355.806 11200 A
1-3-6-9-21-24-18-27-30-15-33-12-1 328.074 11690 B
1-31-28-25-22-19-16-10-7-13-4-1 289.117 6480 C

Tiempo total de ejecución: 0.067997[s]

Archivo: instances/eil76.txt
Solución inicial:
89816.5 2543.49 92360
1-67-10-22-25-34-31-46-49-76-7-28-13-40-55-58-19-61-37-52-4-43-64-16-70-73-1 756.193 42900 C
1-12-15-33-3-57-39-60-21-51-54-6-18-48-27-63-45-9-42-30-36-69-75-66-24-72-1 960.002 46700 B
1-41-5-68-29-65-47-11-32-59-17-35-53-2-44-8-20-62-26-38-14-23-71-56-74-50-1 827.291 46800 A

Solución después de Hill-Climbing:
91214.3 1145.74 92360
1-7-34-64-43-49-16-58-46-28-55-67-13-76-31-22-70-61-37-52-4-25-19-10-40-73-1 389.911 42900 C
1-15-60-12-66-39-51-33-54-36-9-21-72-6-48-75-63-42-57-24-45-3-30-69-18-27-1 381.123 46700 B
1-41-17-50-26-56-32-11-59-8-20-53-47-35-68-5-29-62-71-38-14-2-44-65-23-74-1 374.702 46800 A

Tiempo total de ejecución: 2.658432[s]
rilodeh@Rilodeh:/mnt/c/users/rilod/onedrive/escritorio/MCPwB$
```

Figura 18: Experimento 12

Los experimentos mostrados y en general, todo el algoritmo, no parece ser de una complejidad muy alta, pero el tiempo de ejecución de los resultados puede variar en base al equipo en el que se ejecuta el programa, por lo que, a modo de consideración, se dejan las especificaciones del equipo en el que se llevaron a cabo los experimentos.

Cuadro 1: Especificaciones del equipo

Componente	Especificaciones
Procesador	10th Generation Intel® Core™ i7-10750H
Memoria RAM	8GB DDR4 2933
	16GB DDR4-2933
Unidad de Disco Duro	1TB HDD 2.5. ^a 5400 RPM
	1TB SSD NVMe M.2

8. Resultados

De los experimento realizados, se obtuvieron los siguientes resultados:

Cuadro 2: Experimento 1

Criterio greedy: máxima (Ganancia - costo)

Movimientos HC-AM: 2-OPT entre 2 Rutas y Swap entre nodos por ruta

Instancia	Solución Inicial				Hill Climbing				Tiempo [s]
	Z	VA	VB	VC	Z	VA	VB	VC	
a36	28211.8	16000	14200	14000	29157.1	16000	14200	14000	0.156124
a44	37821.5	23400	17600	16000	38620.7	23400	17600	16000	0.059672
eil33	19713.5	11200	11690	6480	20354	11200	11690	6480	0.137555
eil76	90094.6	42900	46700	46800	91229.7	42900	46700	46800	2.974868

Cuadro 3: Experimento 2

Criterio greedy: máxima (Ganancia - costo)

Movimientos HC-AM: 2-OPT entre 2 Rutas y 2-OPT en una Ruta

Instancia	Solución Inicial				Hill Climbing				Tiempo [s]
	Z	VA	VB	VC	Z	VA	VB	VC	
a36	28211.8	16000	14200	14000	29113	16000	14200	14000	0.150633
a44	37821.5	23400	17600	16000	38592	23400	17600	16000	0.051639
eil33	19713.5	6480	11200	11690	20360	6480	11200	11690	0.120707
eil76	90094.6	42900	46700	46800	91384.1	42900	46700	46800	2.664625

Cuadro 4: Experimento 3

Criterio greedy: máxima (Ganancia - costo)

Movimientos HC-AM: 2-OPT entre 2 Rutas

Instancia	Solución Inicial				Hill Climbing				Tiempo [s]
	Z	VA	VB	VC	Z	VA	VB	VC	
a36	28211.8	16000	14200	14000	28211.8	16000	14200	14000	0.093139
a44	37821.5	23400	17600	16000	37821.5	23400	17600	16000	0.030516
eil33	19713.5	11200	11690	6480	19713.5	11200	11690	6480	0.075588
eil76	90094.6	42900	46700	46800	90094.6	42900	46700	46800	0.644706

Cuadro 5: Experimento 4

Criterio greedy: máxima (Ganancia - costo)

Movimientos HC-AM: 2-OPT en una ruta y Swap entre nodos por ruta

Instancia	Solución Inicial				Hill Climbing				Tiempo
	Z	VA	VB	VC	Z	VA	VB	VC	[s]
a36	28211.8	16000	14200	14000	29204	16000	14200	14000	0.083749
a44	37821.5	23400	17600	16000	38659	23400	17600	16000	0.033707
eil33	19713.5	6480	11690	11200	20381.8	6480	11690	11200	0.066074
eil76	90094.6	42900	46800	46700	91410.5	42900	46800	46700	2.247425

Cuadro 6: Experimento 5

Criterio greedy: máxima (Ganancia - costo)

Movimientos HC-AM: 2-OPT en una ruta

Instancia	Solución Inicial				Hill Climbing				Tiempo
	Z	VA	VB	VC	Z	VA	VB	VC	[s]
a36	28211.8	16000	14200	14000	29113	16000	14200	14000	0.061640
a44	37821.5	23400	17600	16000	38592	23400	17600	16000	0.025797
eil33	19713.5	6480	11690	11200	20360	6480	11690	11200	0.041668
eil76	90094.6	42900	46800	46700	91384.1	42900	46800	46700	2.038468

Cuadro 7: Experimento 6

Criterio greedy: máxima (Ganancia - costo)

Movimientos HC-AM: Swap entre nodos por ruta

Instancia	Solución Inicial				Hill Climbing				Tiempo
	Z	VA	VB	VC	Z	VA	VB	VC	[s]
a36	28211.8	16000	14200	14000	29157.1	16000	14200	14000	0.102351
a44	37821.5	23400	17600	16000	38620.7	23400	17600	16000	0.031530
eil33	19713.5	11200	11690	6480	20354	11200	11690	6480	0.070696
eil76	90094.6	42900	46700	46800	91229.7	42900	46700	46800	2.302419

Cuadro 8: Experimento 7

Criterio greedy: máxima (Ganancia)

Movimientos HC-AM: 2-OPT entre 2 Rutas y Swap entre nodos por ruta

Instancia	Solución Inicial				Hill Climbing				Tiempo
	Z	VA	VB	VC	Z	VA	VB	VC	[s]
a36	28192.3	16000	14200	14000	29113	16000	14200	14000	0.204035
a44	37565.1	23400	17600	16000	38589.4	23400	17600	16000	0.071143
eil33	19696.2	11200	11690	6480	20354	11200	11690	6480	0.141688
eil76	89816.5	42900	46800	46700	91214.3	42900	46800	46700	3.300414

Cuadro 9: Experimento 8

Criterio greedy: máxima (Ganancia)

Movimientos HC-AM: 2-OPT entre 2 Rutas y 2-OPT en una Ruta

Instancia	Solución Inicial				Hill Climbing				Tiempo [s]
	Z	VA	VB	VC	Z	VA	VB	VC	
a36	28192.3	16000	14200	14000	29113	16000	14200	14000	0.155963
a44	37565.1	23400	17600	16000	38576	23400	17600	16000	0.060502
eil33	19696.2	6480	11200	11690	20360	6480	11200	11690	0.117638
eil76	89816.5	42900	46700	46800	91378.3	42900	46700	46800	3.046844

Cuadro 10: Experimento 9

Criterio greedy: máxima (Ganancia)

Movimientos HC-AM: 2-OPT entre 2 Rutas

Instancia	Solución Inicial				Hill Climbing				Tiempo [s]
	Z	VA	VB	VC	Z	VA	VB	VC	
a36	28192.3	16000	14200	14000	28192.3	16000	14200	14000	0.093763
a44	37565.1	23400	17600	16000	37565.1	23400	17600	16000	0.030392
eil33	19696.2	6480	11200	11690	19696.2	6480	11200	11690	0.073582
eil76	89816.5	42900	46700	46800	89816.5	42900	46700	46800	0.640898

Cuadro 11: Experimento 10

Criterio greedy: máxima (Ganancia)

Movimientos HC-AM: 2-OPT en una ruta y Swap entre nodos por ruta

Instancia	Solución Inicial				Hill Climbing				Tiempo [s]
	Z	VA	VB	VC	Z	VA	VB	VC	
a36	28192.3	16000	14000	14200	29210.8	16000	14000	14200	0.095346
a44	37565.1	23400	17600	16000	38659.4	23400	17600	16000	0.043555
eil33	19696.2	11200	11690	6480	20381.8	11200	11690	6480	0.061481
eil76	89816.5	42900	46700	46800	91401	42900	46700	46800	2.662531

Cuadro 12: Experimento 11

Criterio greedy: máxima (Ganancia)

Movimientos HC-AM: 2-OPT en una ruta

Instancia	Solución Inicial				Solución después de Hill-Climbing				Tiempo [s]
	Z	VA	VB	VC	Z	VA	VB	VC	
a36	28192.3	16000	14000	14200	29113	16000	14000	14200	0.056257
a44	37565.1	23400	17600	16000	38576	23400	17600	16000	0.035195
eil33	19696.2	11200	11690	6480	20360	11200	11690	6480	0.049885
eil76	89816.5	42900	46700	46800	91378.3	42900	46700	46800	2.424380

Cuadro 13: Experimento 12

Criterio greedy: máxima (Ganancia)

Movimientos HC-AM: Swap entre nodos de una ruta

Instancia	Solución Inicial				Solución después de Hill-Climbing				Tiempo [s]
	Z	VA	VB	VC	Z	VA	VB	VC	
a36	28192.3	16000	14200	14000	29113	16000	14200	14000	0.114010
a44	37565.1	23400	17600	16000	38589.4	23400	17600	16000	0.042046
eil33	19696.2	11200	11690	6480	20354	11200	11690	6480	0.067997
eil76	89816.5	42900	46700	46800	91214.3	42900	46700	46800	2.658432

Cuadro 14: Comparativa Estado del arte vs Implementación propia

Instancia	Estado del arte					Implementación propia				
	Z	VA	VB	VC	T	Z	VA	VB	VC	T
a36	29233	16000	14200	14000	110	29210.8	16000	14200	14000	0.085890
a44	38771	23400	16000	17600	101	38659.4	23400	16000	17600	0.040021
a55	24694	11900	12250	17800	270	24653.1	11900	12250	17800	0.153425
a64	24100	11750	10650	20000	5395	23884.2	11750	10650	20000	0.091713
a80	29977	16250	14650	16200	5626	29882.6	16250	14650	16200	0.219013
eil22	15947	9800	7200	5500	12	15944.2	9800	7200	5500	0.010812
eil23	7207	6100	2009	2080	6	7198.13	6100	2009	2080	0.012485
eil30	7117	2400	6000	4350	99	7115.15	2400	6000	4350	0.053699
eil33	20409	11200	11690	6480	58	20381.8	11200	11690	6480	0.067387
eil76	91461	46800	46700	42900	1700	91410.5	46800	46700	42900	2.514882

De estos resultados se pueden sacar algunas conclusiones...

Criterios Greedy: Al parecer el algoritmo greedy utilizado es demasiado estricto cuando se utiliza un criterio de $\text{Max}(\text{ganancia} - \text{costo})$ lo que, si bien nos puede dejar más cerca de un óptimo local, no le da mucho margen de mejora al algoritmo Hill Climbing, un criterio más relajado como puede ser $\text{Max}(\text{Ganancia})$ puede ayudar a situarnos en otras regiones del espacio en las que, aunque la solución inicial sea peor, mediante el algoritmo Hill Climbing, se puedan encontrar nuevos óptimos locales mejores, considerando esto, y que, para algunas instancias, se obtiene mejor resultado al aplicar HC a la solución inicial creada con el greedy relajado y para otras se obtiene una mejor con el greedy estricto, se determinó el utilizar ambos criterios, a modo de crear todas las soluciones iniciales posibles y obtener un mejor resultado en Hill Climbing.

2-OPT entre 2 Rutas: Como se puede observar en los resultados, con este movimiento no se obtiene ningún cambio favorable, esto puede ocurrir debido a que es un movimiento complicado de realizar, ya que se pueden romper fácilmente las restricciones del modelo, teniendo en cuenta que, tanto el greedy utilizado como el Hill Climbing, se mueven en lo factible, por lo que, considerando que el greedy está muy pendiente de la ganancia y las restricciones, es probable que cualquier solución generada por 2-OPT entre 2 rutas, sea infactible, concluyendo en que para las próximas instancias, se debe descartar este movimiento.

2-OPT en una ruta y Swap: Ambos movimientos solo modifican el costo de la ruta, y cómo se muestra en los resultados, por separado funcionan bastante bien, entregando resultados mejores a los de la solución inicial, pero además, se puede observar que al utilizarlos juntos, es decir, al aplicar un Hill Climbing con 2-OPT de una ruta y luego

un Hill Climbing con Swap, los resultados suelen ser mejores que hacerlo solo con un movimiento, considerando que los tiempos al utilizar ambos no son muy diferentes que al utilizarlos de forma separada, se decide que la implementación final del algoritmo será con ambos movimientos.

Tiempo de ejecución: Como se puede observar en los resultados, la instancia **eil76** tiene un tiempo de ejecución mucho más alto que las demás instancias, esto es debido a 2 factores, y es que, al generar las soluciones iniciales con Greedy, cuando, con cierto orden de camiones, no se genera una solución factible, dicha solución se descarta, y no se utiliza Hill Climbing con esa solución. Resulta que, para la instancia **eil76**, con cualquier orden de camiones se genera una solución factible, es decir, que genera 6 soluciones con un criterio y 6 soluciones con el otro criterio, además, esta instancia cuenta con más nodos que la mayoría de otras instancias, lo que provoca que tanto el algoritmo Greedy como el Hill Climbing tengan que recorrer más nodos para encontrar una solución inicial y una solución mejor respectivamente.

Búsqueda Local: en base a la comparación entre los resultados del estado del arte con los de la implementación correspondiente al proyecto actual, se puede concluir que una técnica de búsqueda local, puede acercarse bastante al óptimo global en un tiempo considerablemente menor, lo que es muy útil a la hora de buscar soluciones a problemas complejos, aunque, para el caso particular que se está tratando, debido a que las instancias no son de gran tamaño, se observa que mediante una técnica completa se puede llegar a un óptimo global en un tiempo no tan grande.

9. Conclusiones

El problema de recolección de leche ha sido objeto de estudio en Chile y representa un desafío real para la industria láctea. Aunque se ha simplificado el problema en este proyecto, resulta crucial aproximarse a soluciones óptimas en un tiempo razonable. Las técnicas de búsqueda local pueden ser de gran ayuda para alcanzar este objetivo. No obstante, considerando el tamaño de las instancias proporcionadas y la simplificación del problema en comparación con problemas como el de recolección de leche con mezcla gradual, podría resultar más conveniente utilizar técnicas completas para su resolución. Aunque, el uso de técnicas incompletas ha sido un buen acercamiento al problema.

En lo que respecta a la implementación, uno de los mayores desafíos radicó en que, a medida que se avanzaba en el proyecto, se descubrían formas más eficientes de llevar a cabo la implementación de los algoritmos. Esto llevó a modificar grandes porciones de código, sin tener la certeza de que la nueva implementación proporcionaría resultados de calidad o si funcionaría en primer lugar. Cabe destacar que, en uno de los cambios realizados en la implementación, se utilizó un criterio de costo mínimo para el algoritmo **Greedy**. Pero, surgió un problema con una de las instancias, **eil30**, debido a que una de las granjas tenía un gran volumen de leche, pero el costo de llegar a esa granja era tan elevado que no se consideraba hasta casi el final, y cuando finalmente se consideraba, resultaba que ningún camión tenía la capacidad suficiente para transportar ese volumen de leche. Debido a esto se descartó el criterio de costo mínimo en la implementación final.

Para la versión final de la implementación, basándose en los resultados de los experimentos, se decidió incorporar ambos criterios para el algoritmo **Greedy**. En lo que respecta al **Hill Climbing**, se implementaron dos movimientos: **2-OPT en una ruta** y **Swap** entre nodos de una misma ruta. Esta implementación puede generar hasta 12 soluciones factibles en instancias como **eil76**, lo que puede llevar a un tiempo de ejecución considerable, por suerte, dado que se trata de un **Hill Climbing** con alguna mejora, es poco probable que se genere el vecindario completo antes de que finalice el algoritmo, lo que reduce bastante el tiempo de ejecución en

comparación con un **Hill Climbing** con mejor mejora. Además, contar con más soluciones iniciales puede ayudar a explorar el espacio de búsqueda y obtener soluciones de mayor calidad. Finalmente, en cuanto a la implementación del **Hill Climbing**, resultó práctico utilizar movimientos como **2-OPT en una ruta** y **Swap**, ya que ambos movimientos solo afectan al costo de la ruta, y, al no haber restricciones relacionadas con el costo, solo es necesario asegurarse de que la nueva solución generada sea mejor que la anterior.

Referencias

- [1] Nabila Azi, Michel Gendreau, and Jean-Yves Potvin. An adaptive large neighborhood search for a vehicle routing problem with multiple routes. *Computers Operations Research*, 41:167–173, 2014.
- [2] John E. Bell and Patrick R. McMullen. Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*, 18(1):41–48, 2004.
- [3] María Alejandra Enciso Caicedo; Wilfrido Arteaga Sarmiento; Nataly Lorena Guarín Cortés. Vehicle routing model as a transportation alternative, case of estudio: Umng sede campus. *Politécnica*, 14(27):45–56, 2018.
- [4] Simón; Goldberg, Bahamonde; Andrés. Una nueva heurística de construcción y mejoramiento basada en ahorros para instancias grandes del problema de ruteo de vehículos, 2021. Último acceso: 04 de septiembre de 2023.
- [5] Linda Bibiana; Linda Bibiana Rocha Medina; Elsa Cristina Gonzalez la Rotta. State of the art review of the vehicle routing problem: A historic account with solving methods. *Revista de la Facultad de Ingeniería*, 16:35–55, 2011.
- [6] Elizabeth; Lüer-Villagra Armin; Marianov Vladimir; Araya-Sassi Claudio; Paredes-Belmar, Germán; Montero. Vehicle routing for milk collection with gradual blending: A case arising in chile. *European journal of operational research*, 303(3):1403–1416, 2022.
- [7] Vladimir; Bronfman-Andrés; Obreque Carlos; Lüer-Villagra-Armin; Paredes-Belmar, Germán; Marianov. A milk collection problem with blending. *Transportation Research Part E: Logistics and Transportation Review*, 94:26–43, 2016.
- [8] William C; Rodríguez-Vásquez. Modeling a vehicle routing problem with multiple depots, time windows and heterogeneous fleet of a courier service. *CIT Informacion Tecnológica*, 31(1):207–214, 2020.
- [9] Rahul R; Sankaran, Jayaram K; Ubgade. Routing tankers for dairy milk pickup. *Interfaces*, 24(5):59–66, 1994.