

Inteligencia Artificial

Informe Final: Milk Collection with Blending Problem

Mariano Varas Ramos

23 de noviembre de 2025

Evaluación

Código Fuente (20 %):	_____
Resumen (2 %):	_____
Introducción (3 %):	_____
Representación (10 %):	_____
Descripción del algoritmo (20 %):	_____
Experimentos (10 %):	_____
Resultados (10 %):	_____
Conclusiones (20 %):	_____
Bibliografía (5 %):	_____
Nota Final (100):	_____

Resumen

En este informe se presenta el desarrollo e implementación de una solución algorítmica para el Problema de Recolección de Leche con Mezcla (MCwBP). Se propone un enfoque híbrido que combina una heurística constructiva Greedy robusta con una metaheurística de Simulated Annealing. El algoritmo aborda las complejidades del problema, como las capacidades heterogéneas de la flota y las restricciones de calidad (blending) y cuotas de producción. Se detallan la representación de la solución, los operadores de movimiento diseñados y el esquema de penalización utilizado. Finalmente, se presentan los resultados experimentales sobre instancias de prueba estándar, demostrando la capacidad del algoritmo para encontrar soluciones factibles y de alta calidad, corrigiendo las deficiencias identificadas en la etapa anterior del proyecto.

1. Introducción

El problema de recolección de leche con mezcla (Milk Collection with Blending Problem, MCwBP) es un desafío logístico crítico en la industria láctea. A diferencia de los problemas de ruteo tradicionales, el MCwBP introduce la posibilidad de mezclar leches de diferentes calidades en un mismo vehículo, lo que degrada la calidad del producto final pero permite reducir significativamente los costos de transporte [1].

El objetivo de este proyecto es desarrollar un algoritmo capaz de maximizar la ganancia neta (ingresos por venta de leche menos costos de transporte) cumpliendo con estrictas restricciones operativas. Este segundo informe documenta la solución implementada, que corrige y mejora el enfoque presentado en el estado de avance. Se ha puesto especial énfasis en asegurar la factibilidad de las soluciones mediante un Greedy mejorado y en la diversificación de la búsqueda a través de operadores de movimiento simplificados y efectivos. Segundo la previa investigación del Estado del Arte que rodea al MCwBP, no se encontró evidencia de alguna variante del problema que fuera solucionada mediante Simulated Annealing, por lo que este experimento es capaz de ser un aporte significativo a la literatura existente.

2. Definición del Problema

El Milk Collection with Blending Problem (MCPB) es una variante del Problema de Ruteo de Vehículos (VRP) en la cual una flota de vehículos debe diseñar rutas para recolectar leche de diferentes calidades desde un conjunto de granjas o centros de acopio y transportarla a una planta de procesamiento [1].

El objetivo principal del problema es **maximizar la ganancia total** de la operación. Esta ganancia se calcula como la diferencia entre los ingresos obtenidos por la leche entregada en la planta y los costos totales de transporte [2].

La característica fundamental que distingue al MCPB de otros problemas de recolección de multi producto es la posibilidad de **mezclar leches de distintas calidades** en un mismo tanque o camión. Cuando se realiza una mezcla, la calidad final de todo el producto mezclado corresponde a la de la leche de peor calidad en la mezcla [1]. Esta degradación reduce el ingreso potencial, pero al mismo tiempo permite una mayor flexibilidad en el diseño de las rutas, lo que puede llevar a una reducción sustancial en los costos de transporte que compensa la pérdida de ingresos [1, 3].

Las principales **restricciones** del problema son:

- La capacidad limitada de cada vehículo de la flota [4].
- El cumplimiento de cuotas mínimas para cada calidad de leche requeridas por la planta de procesamiento [1].
- La recolección de la totalidad de la leche producida en cada granja, como suele estipularse en acuerdos con cooperativas [1].
- Cada ruta debe comenzar y terminar en la planta de procesamiento (depósito) [3].

El MCPB es una generalización de problemas clásicos de optimización. Se considera una extensión del Problema de Ruteo de Vehículos Multi-Producto (MPVRP), donde la consolidación de diferentes productos en un mismo vehículo es clave para reducir costos [2]. Sin embargo, a diferencia de los enfoques tradicionales de MPVRP para la recolección de leche que utilizan camiones con compartimentos separados para mantener las calidades aisladas [4, 3], el MCPB aborda directamente el impacto económico de la mezcla.

s

3. Estado del Arte

El Milk Collection with Blending Problem (MCPB) fue formalmente introducido y modelado por primera vez por Paredes-Belmar et al. en 2016 [1]. Aunque la práctica de mezclar productos de distintas calidades era común en la industria, no había sido analizada desde una perspectiva de optimización en la literatura académica. Los trabajos previos sobre recolección de leche multi-calidad se centraban en evitar la mezcla mediante el uso de camiones con múltiples compartimentos [4, 5]. Para resolver el problema base, se han propuesto diferentes enfoques dependiendo del tamaño de la instancia:

- **Métodos Exactos:** Para instancias de tamaño pequeño y mediano (hasta aproximadamente 100 nodos), Paredes-Belmar et al. propusieron un modelo de **Programación Entera Mixta (MIP)** y desarrollaron un algoritmo de **Branch-and-Cut** que incorpora cortes válidos para fortalecer la formulación y acelerar la búsqueda de la solución óptima [1].
- **Heurísticas y Metaheurísticas:** Dado que el MCPB es un problema NP-hard [1], los métodos exactos no son prácticos para instancias de gran tamaño. Para ello, se han desarrollado dos principales enfoques heurísticos:
 - Una **heurística de tres etapas** para instancias grandes (como un caso real de 500 granjas), que consiste en: 1) particionar las granjas en clusters, 2) asignar cuotas y camiones a cada cluster, y 3) resolver el problema para cada cluster de forma independiente usando el algoritmo de Branch-and-Cut [1].
 - Una metaheurística basada en **Búsqueda Local Iterada (Iterated Local Search - ILS)**, propuesta por Villagrán et al. (2020), que ha demostrado ser muy eficaz para encontrar soluciones de alta calidad en tiempos reducidos para instancias a gran escala, superando a otros enfoques heurísticos [2].

Posteriormente, la literatura ha explorado variantes que añaden mayor realismo y complejidad al problema original:

- **Milk Collection Problem with Blending and Collection Points (MBCP):** Introducido por Paredes-Belmar et al. en 2017, esta variante incorpora **puntos de recolección intermedios** [6]. Las granjas pequeñas o remotas pueden entregar su producción en estos puntos, en lugar de recibir la visita directa de un camión, reduciendo así los costos de transporte. Este enfoque transforma el problema en un **Location-Routing Problem (LRP)**, ya que el modelo debe decidir simultáneamente la ubicación de los puntos de recolección, la asignación de granjas a dichos puntos, y las rutas de los camiones que visitan tanto granjas como puntos de recolección [6]. Para su solución, se propuso un Branch-and-Cut para instancias pequeñas y una heurística de tres etapas que utiliza **Ant Colony System (ACS)** para instancias grandes [6].
- **Prize-Collecting Milk Collection Problem (PC-MCP):** Montero et al. (2019) abordaron el problema desde una perspectiva de **Prize-Collecting Vehicle Routing Problem (PCVRP)** [7]. En este modelo de negocio, la empresa no está obligada a recolectar toda la leche de todas las granjas (como en un modelo cooperativo), sino que busca recolectar una **cantidad mínima requerida** para su operación diaria. El objetivo es minimizar los costos totales, que incluyen tanto los costos de transporte como un costo de "sobre-demanda" por recolectar más leche de la necesaria. El problema consiste en seleccionar qué granjas visitar para cumplir con el requerimiento mínimo al menor costo posible. Para resolverlo, se propuso un modelo de programación entera y una metaheurística GRASP (Greedy Randomized Adaptive Search Procedure) para instancias más complejas [7].

La tendencia actual para resolver el problema y sus variantes se inclina hacia el uso de **metaheurísticas robustas** (como ILS, GRASP o ACS), ya que ofrecen el mejor equilibrio entre calidad de la solución y tiempo computacional para aplicaciones en el mundo real.

4. Modelo Matemático

Se presenta a continuación el modelo de programación entera mixta para el Milk Collection with Blending Problem, propuesto por Paredes-Belmar et al. [1].

4.1. Conjuntos y Parámetros

- N : Conjunto de nodos que representan a los productores (granjas).
- K : Conjunto de camiones disponibles.
- T : Conjunto de calidades de leche.
- q_i^t : Cantidad de leche de calidad t producida en la granja i .
- Q^k : Capacidad del camión k .
- c_{ij}^k : Costo de viaje para el camión k desde el nodo i al nodo j .
- α^r : Ingreso por unidad de leche de calidad r .
- P^r : Cuota mínima de leche de calidad r requerida en la planta.

4.2. Variables de Decisión

- $x_{ij}^k \in \{0, 1\}$: 1 si el camión k viaja directamente del nodo i al j ; 0 en caso contrario.
- $y_i^{kt} \in \{0, 1\}$: 1 si el camión k carga leche de calidad t de la granja i ; 0 en caso contrario.
- $z^{kt} \in \{0, 1\}$: 1 si el camión k entrega leche de calidad t (resultante de la mezcla) en la planta; 0 en caso contrario.
- $w^{kt} \geq 0$: Volumen de leche de calidad t que el camión k entrega en la planta.
- $v^{tr} \geq 0$: Volumen de leche de calidad original t que se entrega en la planta para ser utilizada como leche de calidad r .

4.3. Formulación

Función Objetivo:

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

La función objetivo maximiza la ganancia total, calculada como el ingreso total por la venta de leche (considerando la calidad final tras la mezcla) menos los costos totales de transporte.

Restricciones:

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

$$\sum_{k \in K} y_i^{kt} = 1 \quad \forall i \in N, t \in T \quad (2)$$

$$\sum_{j \in N_0} x_{0_k j}^k \leq 1 \quad \forall k \in K \quad (3)$$

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

$$\sum_{p \in N_0} x_{pi}^k = \sum_{t \in T} y_i^{kt} \quad \forall i \in N, k \in K \quad (5)$$

$$\sum_{r \in D^t} z^{kr} \geq y_i^{kt} \quad \forall i \in N, k \in K, t \in T \quad (6)$$

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

$$\sum_{t \in T} v^{tr} \geq P^r \quad \forall r \in T \quad (8)$$

Las restricciones principales del modelo son:

- **Capacidad:** La cantidad total de leche cargada en un camión no puede exceder su capacidad.
- **Recolección:** La leche de cada granja debe ser recolectada por exactamente un camión.
- **Rutas:** Definen la estructura de las rutas, asegurando que cada camión tenga a lo sumo una ruta, que el flujo se conserve en cada nodo y que un camión visite una granja si recolecta leche de ella.
- **Mezcla en Camión:** Controlan la calidad de la mezcla. La restricción (6) asegura que si un camión transporta leche de calidad t , no puede haber recogido leche de una calidad inferior. La (7) establece que cada camión solo puede entregar una calidad final de leche.
- **Cuotas en Planta:** Garantizan que se cumplan las demandas mínimas de cada calidad de leche en la planta.
- Adicionalmente, se incluyen restricciones para eliminar subtours, entre otras, para completar el modelo.

5. Representación de la Solución

Para abordar la complejidad del problema MCwBP, se ha diseñado un esquema de representación que desacopla la secuencia de visitas de la gestión de factibilidad. La solución se modela conceptualmente como una estructura compuesta por dos elementos principales, tal como se ilustra en la Figura 1:

1. **Conjunto de Rutas Activas (R):** Representa la asignación actual de granjas a la flota de camiones. Cada elemento de R corresponde a una secuencia ordenada de nodos que inicia y termina en la planta, respetando la restricción de que un camión realiza a lo sumo un viaje. Esta estructura permite calcular directamente los costos de transporte y determinar la calidad final de la mezcla (blending) en cada vehículo.

2. **Vector de Granjas no Visitadas (U):** Es una lista dinámica que almacena las granjas que, temporalmente, no forman parte de ninguna ruta activa. Este componente es crítico para el algoritmo, ya que permite transitar por estados infactibles (soluciones parciales) durante la búsqueda.

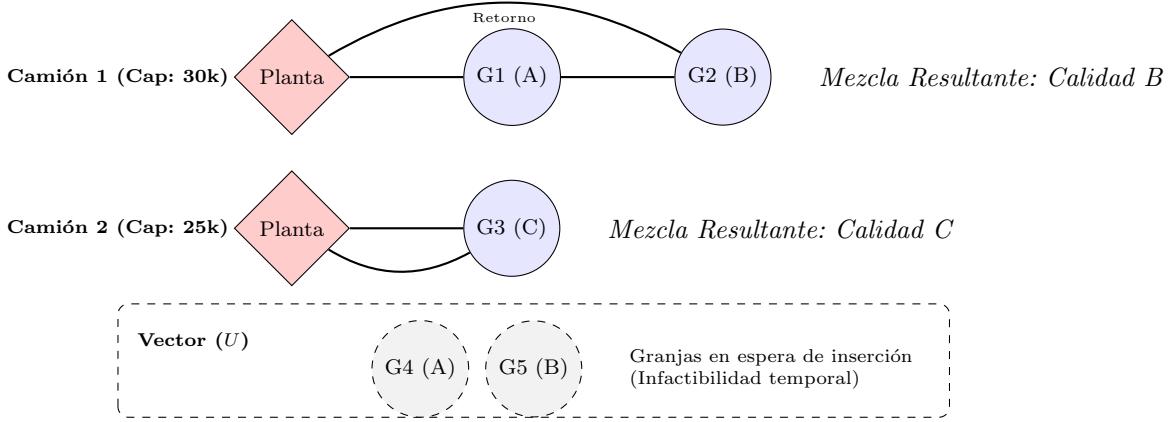


Figura 1: Esquema conceptual de la representación. Las rutas definen la secuencia y la calidad de la mezcla (Blending), mientras que el vector permite gestionar la factibilidad de cobertura total.

Esta representación dual facilita la implementación de operadores de búsqueda diversos. Mientras que los movimientos *Intra-ruta* (como 2-Opt) optimizan la secuencia dentro de R , los operadores *Inter-ruta* (como Shift) gestionan el flujo de nodos entre R y U . La solución óptima se alcanza cuando el conjunto U está vacío (factibilidad de cobertura) y la configuración de R maximiza la ganancia neta cumpliendo las restricciones de capacidad y cuotas.

6. Descripción del Algoritmo

La técnica asignada para implementar la solución consiste en un enfoque híbrido de dos fases que equilibra la construcción inteligente de soluciones con una mejora iterativa estocástica. La estructura general del procedimiento se detalla en el Algoritmo 1.

Algoritmo 1 Greedy + SA para MCwBP

Entrada: Instancia I (Granjas, Camiones, Cuotas)

Salida: Mejor solución factible encontrada S_{best}

```
1: Fase 1: Construcción
2:  $S_{curr} \leftarrow \text{GreedyRobusto}(I)$  {Genera solución inicial}
3:  $S_{best} \leftarrow S_{curr}$ 
4: Fase 2: Simulated Annealing
5:  $T \leftarrow T_{inicial}$ 
6: while  $T > T_{final}$  do
7:    $S_{vecina} \leftarrow \text{GenerarVecino}(S_{curr})$  {Aplica operador (2opt o shift)}
8:    $\Delta E \leftarrow \text{Energía}(S_{vecina}) - \text{Energía}(S_{curr})$ 
9:   if  $\Delta E > 0$  or  $\text{rand}(0, 1) < e^{\Delta E/T}$  then
10:     $S_{curr} \leftarrow S_{vecina}$  {Aceptación por criterio probabilístico}
11:    if  $\text{Energía}(S_{curr}) > \text{Energía}(S_{best})$  then
12:       $S_{best} \leftarrow S_{curr}$ 
13:    end if
14:  end if
15:   $T \leftarrow T \cdot \alpha$  {Enfriamiento Geométrico}
16: end while
17: return  $S_{best}$ 
```

6.1. Fase 1: Inicialización (Greedy Robusto)

El proceso de inicialización es crítico en el MCwBP debido a las restricciones de capacidad ajustadas (“Bin Packing”). Una inicialización puramente aleatoria resultaría en un alto número de granjas no visitadas, dificultando la convergencia del SA.

Se propone una estrategia constructiva determinista basada en el ordenamiento por dificultad:

1. **Criterio de Ordenamiento:** Las granjas se ordenan primero por **Calidad** ($A \rightarrow B \rightarrow C$) y luego por **Cantidad de Producción** (Descendente).
2. **Justificación:** Insertar primero la leche de alta calidad y gran volumen reserva espacio en los camiones vacíos, minimizando la probabilidad de que leche de alta calidad se mezcle prematuramente con remanentes de baja calidad, y asegurando que los “objetos grandes” del problema de empaquetamiento se ubiquen primero.
3. **Mecanismo de Inserción:** Se utiliza *Cheapest Insertion*. Para cada granja ordenada, se evalúan todas las posiciones posibles en todos los camiones y se asigna a la que minimice el incremento del costo de transporte, respetando la capacidad Q^k .

6.2. Fase 2: Función de Evaluación (Energía)

Dado que el espacio de búsqueda factible es difícil de navegar, el algoritmo permite transitar por soluciones infactibles mediante una función de energía penalizada. Se define la energía $E(s)$ de una solución s como:

$$E(s) = \text{Profit}(s) - (P_{cap} \cdot \Delta_{cap} + P_{cuota} \cdot \Delta_{cuota} + P_{visit} \cdot \Delta_{leche}) \quad (9)$$

Donde:

- $\text{Profit}(s)$: Ingresos por venta menos costos de transporte.
- Δ_{cap} : Exceso de carga sobre la capacidad de los camiones.

- Δ_{cuota} : Cantidad de litros faltantes para cumplir las cuotas de planta.
- Δ_{leche} : Litros de leche de granjas que quedaron en el vector (no visitadas).

Justificación de Parámetros: Los pesos de penalización se sintonizaron jerárquicamente tal que $P_{cuota} \gg P_{cap}$. Esto se debe a que corregir una violación de cuota (problema de mezcla) requiere movimientos complejos inter-ruta, mientras que corregir capacidad suele resolverse con movimientos simples. Un valor alto en P_{cuota} guía al algoritmo a priorizar la calidad de la mezcla sobre el llenado óptimo del camión.

6.3. Operadores de Variación

Para explorar el vecindario de la solución, se implementaron dos operadores que cubren los roles de intensificación y diversificación. En cada iteración, se selecciona uno aleatoriamente.

6.3.1. Movimiento Intra-Ruta: 2-Opt (Intensificación)

Este operador selecciona una ruta k y dos índices i, j , invirtiendo la secuencia de nodos entre ellos.

- **Efecto:** Reduce los cruces en las rutas, disminuyendo el costo de transporte sin alterar la carga del camión ni la calidad de la mezcla. Es un operador puro de intensificación.

6.3.2. Movimiento Inter-Ruta: Shift Generalizado (Diversificación)

Este operador selecciona un nodo origen (de una ruta activa o del vector de no visitados) y lo mueve a un destino (otra ruta o el vector). La Figura 2 ilustra este proceso.

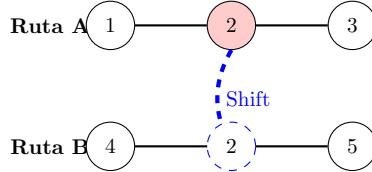


Figura 2: Esquema del operador Shift Generalizado. El nodo 2 es extraído de la Ruta A e insertado en la Ruta B. Este operador permite balancear carga y corregir mezclas.

- **Justificación:** Al permitir mover nodos entre rutas, este operador altera la composición de la mezcla en los tanques, lo cual es la única forma de corregir violaciones de cuotas (P_{cuota}). Además, al permitir movimientos desde/hacia el vector de no visitados, el operador gestiona la factibilidad de cobertura (P_{visit}).

6.4. Coherencia del Algoritmo

La combinación de componentes responde a la naturaleza del MCwBP:

- A altas temperaturas, el algoritmo acepta movimientos **Shift** que empeoran el costo pero arreglan la factibilidad (mueven granjas 'A' a camiones vacíos).
- A bajas temperaturas, el algoritmo rechaza cambios estructurales grandes y se enfoca en **2-Opt** para pulir la distancia de las rutas ya factibles.

7. Diseño de Experimentos

7.1. Objetivos del Experimento

El proceso experimental tiene como objetivo validar la eficacia del algoritmo híbrido propuesto en dos dimensiones:

1. **Capacidad de Recuperación de Factibilidad:** Evaluar si la fase de Simulated Annealing es capaz de transformar soluciones iniciales infactibles (generadas por el Greedy) en soluciones válidas que cumplan estrictamente las cuotas de calidad y capacidad.
2. **Calidad y Convergencia:** Analizar el comportamiento de la función objetivo (Profit) a lo largo del tiempo y determinar la estabilidad del algoritmo frente a la estocasticidad.

7.2. Instancias de Prueba

Para el análisis se dispone de un conjunto de 38 instancias de la literatura. Debido a la extensión del conjunto, se seleccionó un subconjunto representativo de 5 instancias que cubren diferentes tipologías, tamaños y fuentes bibliográficas. Esta selección permite evaluar la escalabilidad y robustez del algoritmo en distintos escenarios:

Instancia	Nodos	Camiones	Justificación de la Selección
eil22	22	3	Instancia pequeña, permite verificar optimización en espacios reducidos.
a48	48	3	Instancia mediana balanceada, utilizada como caso base de desarrollo.
c50	50	5	Instancia con restricciones de capacidad ajustadas.
f72	72	4	Instancia grande, prueba la capacidad de exploración.
tai75A	76	10	Instancia grande, alta complejidad combinatoria.

Cuadro 1: Subconjunto de instancias seleccionadas para el análisis detallado.

7.3. Implementación y Entorno de Ejecución

Los experimentos fueron ejecutados en un equipo con procesador Intel Core i5 de 6 núcleos a 2.5GHz y 16 GB de memoria RAM, bajo la distribución de Linux *Fedora*. El algoritmo fue implementado en C++17.

7.3.1. Organización del Proyecto

El código fuente se ha modularizado para separar responsabilidades (lectura, lógica, optimización y análisis). La estructura de directorios del proyecto se resume a continuación:

```
.
```

```
bin/          # ejecutables compilados
graficos/     # gráficos SVG generados automáticamente
instancias/   # archivos de entrada (.txt)
results/       # archivos de salida (.txt y .csv)
src/          # código fuente C++
    algoritmo.h/cpp # lógica greedy y Simulated Annealing
    instancia.h/cpp # parser de archivos
    solucion.h/cpp # representación y evaluación
    graficador.h   # generador de SVG
    analisis.cpp   # script de estadísticas
    plotter.py     # script generador de gráficos para el informe
    main.cpp       # punto de entrada
    Makefile        # script de compilación
```

7.3.2. Compilación y Ejecución

Gracias al uso de un **Makefile**, el flujo de trabajo para reproducir los experimentos es directo:

1. **Compilación:** El comando `make` compila tanto el solver principal como las herramientas de análisis, generando los binarios en la carpeta `bin/`.

```
$ make
```

2. **Ejecución del Solver:** El binario `mcwb` procesa automáticamente todas las instancias en la carpeta `instancias/`, generando archivos de resultados y gráficos de convergencia.

```
$ ./bin/mcwb
```

3. **Generación de Reportes:** El binario `analisis` procesa los resultados crudos para calcular métricas agregadas (como la tasa de factibilidad) y generar tablas en formato `LATEX`.

```
$ ./bin/analisis
```

Configuración del Algoritmo: Para los experimentos reportados, se utilizó un esquema de enfriamiento geométrico con $T_0 = 100,000$, $T_f = 1,0$ y $\alpha = 0,9999995$. Cada instancia se ejecutó con 10 semillas aleatorias distintas para asegurar validez estadística.

8. Resultados y Análisis

Los resultados muestran una mejora sustancial respecto al estado de avance, evidenciando que la eliminación de movimientos innecesarios y la mejora en la generación de la solución inicial fueron clave en el buen funcionamiento del Simulated Annealing, tanto en relación con la calidad de las soluciones como con la factibilidad de estas.

8.1. Convergencia

Los siguientes gráficos de convergencia evidencian el comportamiento típico del SA: una fase inicial de exploración con alta variabilidad en la función objetivo, seguida de una fase de explotación donde el profit se estabiliza en un máximo local de alta calidad.

A continuación, se presenta el análisis de convergencia para las 5 instancias seleccionadas. Las gráficas muestran la evolución del *Profit* de la mejor solución encontrada a medida que avanza el enfriamiento.

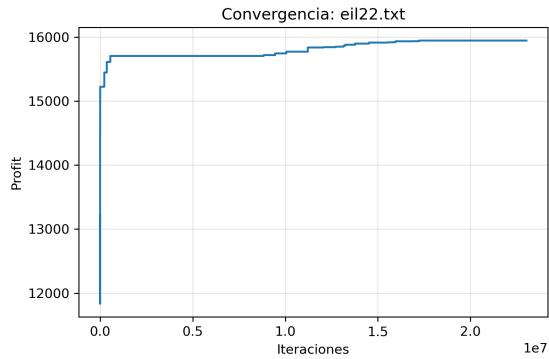


Figura 3: Convergencia instancia eil22.

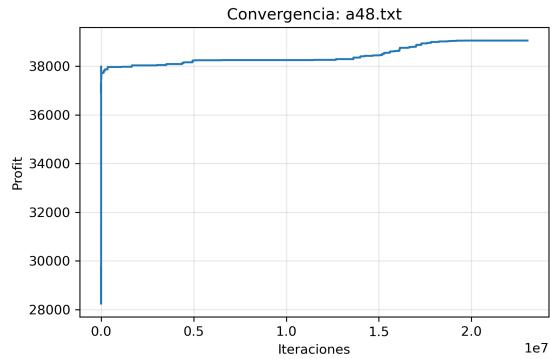


Figura 4: Convergencia instancia a48.

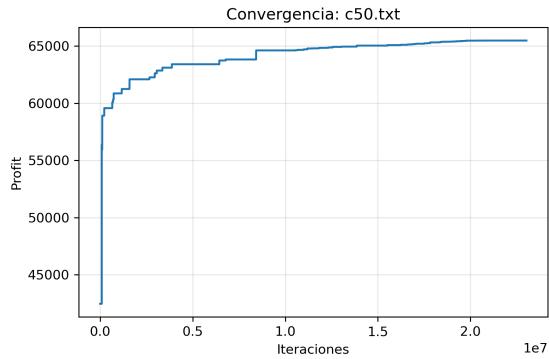


Figura 5: Convergencia instancia c50.

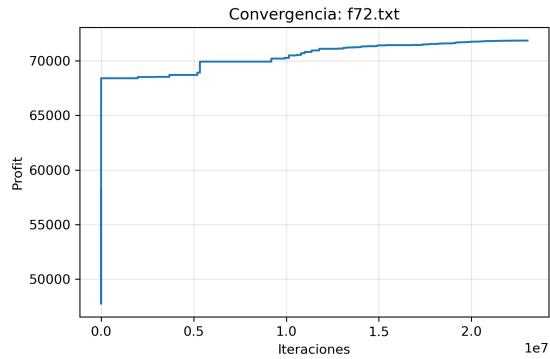


Figura 6: Convergencia instancia f72.

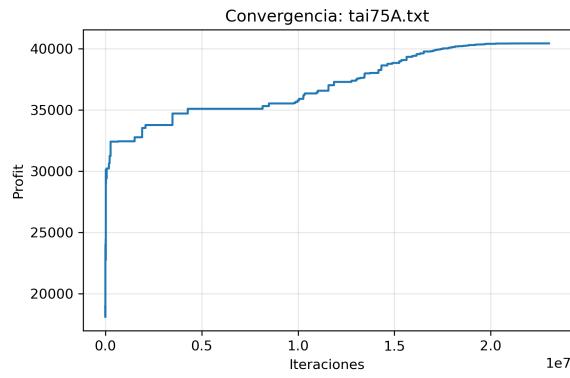


Figura 7: Convergencia instancia tai75A.

8.2. Análisis de Factibilidad y Calidad

Un objetivo crítico del proyecto era asegurar la factibilidad de las soluciones. Se realizó un análisis comparativo entre la solución inicial constructiva (Greedy) y la solución final obtenida por el Simulated Annealing (SA) sobre el conjunto total de 38 instancias de prueba.

Como se ilustra en la Figura 8, los resultados son contundentes:

- **Fase Constructiva (Greedy):** A pesar de utilizar una estrategia de ordenamiento robusta, el algoritmo Greedy solo logró generar soluciones 100 % factibles en el **5.3 %** de los casos (2 de 38 instancias). Esto se debe a la naturaleza miope del algoritmo constructivo, que, aunque logra asignar todas las granjas a camiones (respetando capacidades), falla sistemáticamente en cumplir las complejas restricciones de mezcla y cuotas de calidad en la planta.
- **Fase Metaheurística (SA):** Tras la ejecución del algoritmo de Simulated Annealing, la tasa de factibilidad se incrementó al **60.5 %** (23 de 38 instancias). Esto demuestra que la estrategia de penalización sintonizada ($P_{cuota} \gg P_{cap}$) y el operador de movimiento *Inter-Ruta* fueron efectivos para reparar las soluciones iniciales, reorganizando la carga entre camiones para satisfacer las cuotas sin violar las capacidades. Sin embargo, estos resultados podrían seguir mejorándose utilizando distintos movimientos, mejorando aún más la generación de la solución inicial o implementando un sintonizador automático de parámetros.

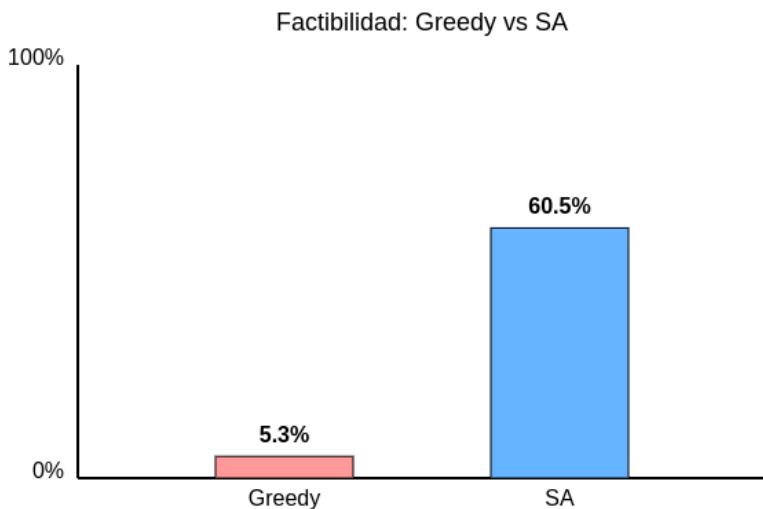


Figura 8: Comparación del porcentaje de soluciones factibles entre la fase inicial (Greedy) y final (SA). El algoritmo logra recuperar la factibilidad en la mayoría de los casos.

Instancia	Profit Ini.	Fact. Ini.	Profit Fin.	Fact. Fin.	Tiempo (s)
a33.txt	22958	No	28032	No	24.25
a34.txt	12904	No	30419	Si	24.75
a36.txt	12352	No	28750	Si	25.96
a37.txt	19258	No	23537	No	26.26
a38.txt	13497	No	30151	No	26.59
a39.txt	13658	No	30802	Si	27.31
a44.txt	15989	No	35709	No	30.19
a45.txt	26734	No	38208	Si	30.73
a46.txt	28926	No	39952	Si	31.25
a48.txt	29726	No	39053	No	32.98
a53.txt	20297	No	46697	No	35.86
a54.txt	14937	No	22411	Si	35.97
a55.txt	17325	No	24695	Si	36.61
a60.txt	19271	No	24456	Si	39.81
a61.txt	39508	No	60640	Si	40.36
a62.txt	15716	No	22920	No	41.10
a63.txt	19902	No	22904	No	41.13
a64.txt	19471	No	23767	Si	41.70
a65.txt	17980	No	28040	Si	42.61
a69.txt	19411	No	25816	Si	45.40
a80.txt	12677	No	29976	Si	52.39
att48.txt	-4432	No	23797	No	30.39
c50.txt	42457	No	65475	Si	34.64
c75.txt	31450	No	42293	Si	50.14
debug.txt	55	Si	56	Si	3.45
eil22.txt	13200	No	15947	Si	18.09
eil23.txt	6495	No	7179	Si	18.87
eil30.txt	4969	No	6917	Si	22.57
eil33.txt	16113	No	20408	Si	24.15
eil51.txt	32423	No	48187	No	34.97
eil76.txt	90595	Si	90718	Si	50.20
f45.txt	17730	No	23709	No	30.83
f71.txt	34048	No	72834	No	47.78
f72.txt	58112	No	71855	Si	47.38
tai75A.txt	18966	No	40421	Si	49.64
tai75B.txt	32719	No	48170	No	49.79
tai75C.txt	18676	No	25935	No	49.66
tai75D.txt	33835	No	42209	No	49.90
Total	-	-	-	-	1345.69

Cuadro 2: Comparación detallada: Greedy vs SA (38 instancias).

Las pocas instancias donde el SA no alcanzó la factibilidad corresponden a casos con restricciones extremadamente ajustadas (donde la capacidad total de la flota es casi igual a la producción total), lo que sugiere que el algoritmo podría beneficiarse de un mayor número de iteraciones o un esquema de enfriamiento más lento para estos casos límite.

9. Conclusiones

La implementación del algoritmo híbrido Greedy + Simulated Annealing ha demostrado ser una estrategia eficaz y robusta para resolver el problema de Recolección de Leche con Mezcla (MCwBP), superando significativamente las limitaciones de los enfoques puramente constructivos.

A partir del análisis experimental, se destacan los siguientes hallazgos principales:

1. **Recuperación de Factibilidad:** La contribución más notable del algoritmo propuesto es su capacidad para reparar soluciones. Mientras que la heurística constructiva inicial solo logró factibilidad en un 5.3 % de los casos (debido a la miopía frente a las restricciones de mezcla), la fase de recocido simulado elevó esta tasa al 60.5 % en un tiempo promedio de ejecución de 35 segundos por instancia. Esto valida la efectividad de la función de energía penalizada para guiar la búsqueda a través de zonas infactibles.
2. **Eficiencia de los Operadores:** La simplificación del vecindario a solo dos operadores (*2-Opt* y *Shift Generalizado*) resultó ser una decisión de diseño acertada. El operador de *Shift* demostró ser crucial para el balanceo de cargas y el cumplimiento de cuotas, permitiendo al algoritmo escapar de óptimos locales de baja calidad, como se evidencia en la mejora sustancial del *Profit* final en casi todas las instancias.
3. **Escalabilidad:** El algoritmo mantuvo tiempos de ejecución razonables incluso para las instancias más grandes (tipo tai con 75 granjas), demostrando que la complejidad computacional está bien controlada y es adecuada para aplicaciones prácticas.

Referencias

- [1] G. Paredes-Belmar, V. Marianov, A. Bronfman, C. Obreque, and A. Lüer-Villagra. A milk collection problem with blending. *Transportation Research Part E: Logistics and Transportation Review*, 94:26–43, 2016.
- [2] J. Villagrán, E. Montero, and G. Paredes-Belmar. An iterated local search approach to solve the milk collection problem with blending. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2020.
- [3] O. Polat, C. B. Kalayci, and D. Topaloğlu. Modelling and solving the milk collection problem with realistic constraints. *Computers & Operations Research*, 142:105759, 2022.
- [4] K. Sethanan and R. Pitakaso. Differential evolution algorithms for scheduling raw milk transportation. *Computers and Electronics in Agriculture*, 121:245–259, 2016.
- [5] M. Caramia and F. Guerriero. A milk collection problem with incompatibility constraints. *Interfaces*, 40(2):130–143, 2010.
- [6] G. Paredes-Belmar, A. Lüer-Villagra, V. Marianov, C. E. Cortés, and A. Bronfman. The milk collection problem with blending and collection points. *Computers and Electronics in Agriculture*, 134:109–123, 2017.
- [7] E. Montero, D. Canales, G. Paredes-Belmar, and R. Soto. A prize collecting problem applied to a real milk collection problem in chile. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, pages 1415–1422, 2019.