Informe Técnico

Análisis de Complejidad y Diseño de Algoritmos para el Problema del Transporte Discreto

Discrete Logistics

Autores:
Adrian Alejandro Souto Morales
Gabriel Herrera Carrazana

2025

Proyecto Final de Diseño y Análisis de Algoritmos

1. Introducción

En *Discrete Logistics*, operamos en un nicho de mercado extremadamente delicado: el transporte transfronterizo de productos de valor incalculable. Este informe aborda el problema computacional de distribuir un conjunto de artículos de alto valor entre un número determinado de transportistas ("mulas"), respetando estrictamente la capacidad de carga y minimizando el riesgo financiero global.

El objetivo es lograr una distribución del valor total de los artículos lo más equitativa posible, reduciendo el impacto de una pérdida catastrófica si un único transportista es comprometido.

2. Definición Formal del Problema

Dado un conjunto de N artículos $A = \{a_1, ..., a_n\}$ y M mulas $M = \{m_1, ..., m_m\}$, buscamos una partición tal que:

1. **Restricción de Capacidad:** $\sum_{a_i \in A_j} p_i \leq C_j$ para toda mula m_j.
2. **Función Objetivo (Equidad):** Minimizar K tal que $|V_j - V_k| \leq K$ para todo par de mulas (j, k), donde V_j es el valor total asignado a la mula j.

3. Análisis de Complejidad (Fase 2)

Para determinar la viabilidad computacional, analizamos la complejidad teórica del problema.

3.1. NP-Completitud

Demostramos que el problema pertenece a la clase **NP** y es **NP-Duro** mediante una reducción desde el **Problema de la Partición (PARTITION)**.

- **Construcción:** Dada una instancia de PARTITION (multiconjunto S), construimos una instancia de transporte con $M = 2$ mulas, capacidad $T/2$ (donde $T = \sum S_i$) y umbral $K = 0$.
- **Equivalencia:** Existe una solución al problema de transporte con diferencia 0 si y solo si existe una partición perfecta del conjunto S.

Conclusión: El problema es **NP-Completo**, lo que justifica el uso de heurísticas para instancias grandes.

4. Diseño de Algoritmos (Fase 3)

4.1. Enfoque Exacto: Fuerza Bruta

Este algoritmo garantiza encontrar la solución óptima explorando todo el espacio de búsqueda.

Algoritmo 1 Fuerza Bruta para Transporte Discreto

Entrada: Artículos A, Mulas M
Salida: Mejor Asignación S_{opt}

$MejorDiferencia \leftarrow \infty$
Para cada asignación S posible de A en M **do** ▷ Complejidad $O(M^N)$
  Si S cumple restricciones de peso **then**
    $Dif \leftarrow \text{máx}(V(m)) - \text{mín}(V(m))$
    Si $Dif < MejorDiferencia$ **then**
      $MejorDiferencia \leftarrow Dif$
      $S_{opt} \leftarrow S$
    end Si
  end Si
end Para
Retornar S_{opt}

Complejidad: $O(M^N \cdot (N + M^2))$. Inviable para $N > 12$.

4.2. Heurística Voraz (Greedy)

Estrategia eficiente que ordena los artículos por valor y los asigna a la mula con menor carga actual para balancear.

Algoritmo 2 Heurística Voraz (Greedy)

Entrada: Artículos A, Mulas M

Ordenar A descendentemente por valor
Para cada artículo art en A **do**
  $Candidata \leftarrow$ Mula válida con menor $Valor_{actual}$
  Si $Candidata$ existe **then**
    Asignar art a $Candidata$
  Sino
    Retornar Error (No hay solución factible)
  end Si
end Para
Retornar Asignación final

Complejidad: $O(N \log N + N \cdot M)$.

4.3. Metaheurística: Búsqueda Local

Mejora la solución voraz aplicando movimientos de *Relocate* y *Swap* entre la mula más cargada y la menos cargada.

Algoritmo 3 Búsqueda Local (Hill Climbing)

Entrada: Solución Inicial S

repeat
  $Mejora \leftarrow Falso$
  $M_{max} \leftarrow$ Mula con mayor valor
  $M_{min} \leftarrow$ Mula con menor valor
  Intento 1: Mover artículo de M_{max} a M_{min}
  Si movimiento reduce diferencia **then** $Mejora \leftarrow Verdadero$
  end Si
  Intento 2: Intercambiar artículos entre M_{max} y M_{min}
  Si intercambio reduce diferencia **then** $Mejora \leftarrow Verdadero$
  end Si
until $!Mejora$ o límite iteraciones

5. Análisis Experimental (Fase 4)

Los experimentos fueron realizados en Python comparando los tres enfoques.

5.1. Calidad de la Solución

Se probó con instancias pequeñas ($N = 4$ a 10, $M = 2$) para comparar contra el óptimo global.

Tabla 1: Diferencia promedio respecto al Óptimo (Gap en $)

N (Artículos)	Gap Greedy	Gap Búsqueda Local
4	0.0	0.0
5	9.6	0.0
6	0.8	0.8
7	7.2	0.0
8	48.0	4.8
9	12.0	2.8
10	15.2	8.4

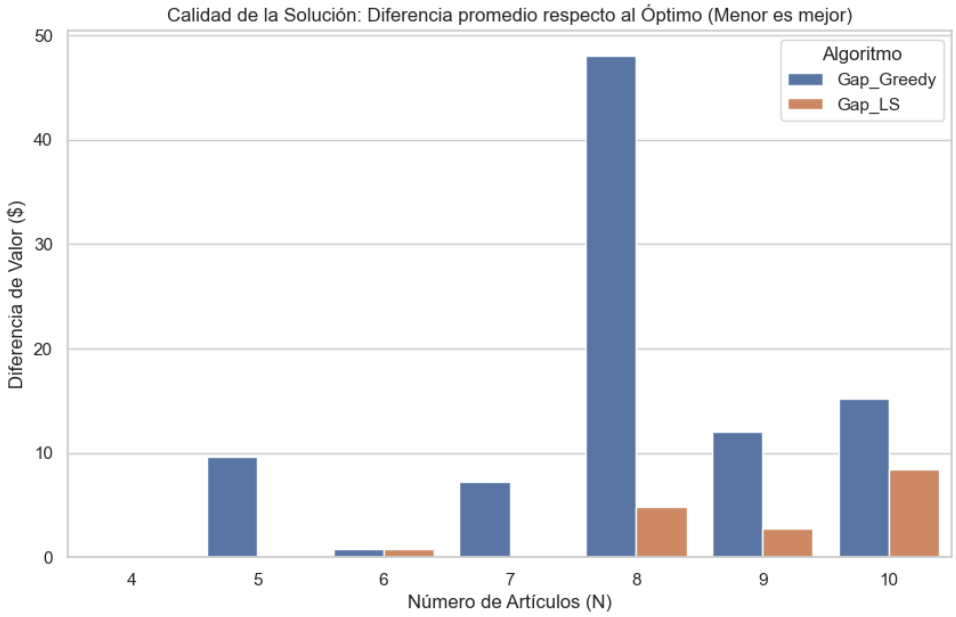

Figura 1: El algoritmo Greedy presenta picos de error altos (ej. N=8), mientras que la Búsqueda Local reduce drásticamente esta brecha, acercándose al óptimo.

5.2. Escalabilidad

Se realizaron pruebas de estrés con N hasta 500.

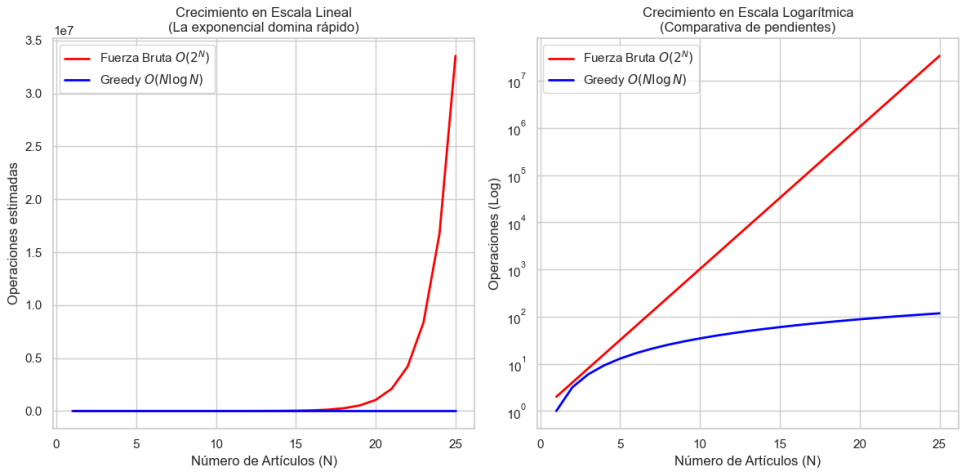

Figura 2: La Fuerza Bruta crece exponencialmente, volviéndose inútil para $N > 15$. Los algoritmos aproximados se mantienen por debajo de 1 segundo incluso con 500 artículos.

6. Conclusión

El análisis teórico y experimental confirma que el problema es intratable por métodos exactos para tamaños realistas. Sin embargo, la implementación de la **Metaheurística de Búsqueda Local** ha demostrado ser capaz de encontrar soluciones con un error promedio muy bajo ($< 5\,\%$ respecto al valor total en las pruebas) en

tiempos de ejecución insignificantes, cumpliendo así con los requisitos operativos de *Discrete Logistics*.