

TP2 - Gestión eficiente de recursos en sistemas ferroviarios

Olivia Nicholson, Sofia Ferrari y Milena Fuchs

Noviembre 2024

Contents

1	Introducción	3
2	Descripción del Problema	3
3	Modelo Implementado	4
3.1	Construcción del Grafo	4
3.2	Algoritmo Implementado	5
3.3	Justificación	5
3.4	Visualización	7
4	Datos	7
5	Escenario Adicional	7
5.1	Descripción de los Escenarios	8
5.2	Resultados y Comparación	8
5.3	Conclusión	8
6	Experimentación	9
6.1	Experimento demandas:	9
6.1.1	Metodología	9
6.1.2	Desarrollo	9
6.1.3	Tiempos de ejecución	10
6.1.4	Análisis de los resultados	10
6.2	Experimento capacidades:	10
6.2.1	Metodología	10
6.2.2	Desarrollo	11
6.2.3	Análisis de los resultados	11
6.3	Experimento Frecuencia:	11
6.3.1	Metodología	11
6.3.2	Desarrollo	11
6.3.3	Análisis de los resultados	12
7	Conclusión	12

1 Introducción

En este trabajo práctico nos centramos en la resolución de un problema clásico en la planificación táctica de sistemas ferroviarios: la circulación de material rodante. Este problema consiste en determinar la cantidad mínima de unidades de material rodante necesarias para satisfacer la demanda de pasajeros en un cronograma fijo de servicios.

El objetivo principal del trabajo es modelar y resolver el problema utilizando un enfoque basado en grafos y flujos de costo mínimo, siguiendo el modelo propuesto por Schrijver. Este enfoque permite representar la red espacio-tiempo de servicios ferroviarios y optimizar la utilización de las unidades disponibles mediante la reutilización eficiente y la minimización de movimientos innecesarios.

2 Descripción del Problema

Actualmente, la empresa utiliza un enfoque manual e intuitivo para abordar este problema. Sin embargo, este método tiene importantes limitaciones. No considera el cronograma completo para optimizar la reutilización de unidades, lo que resulta en un uso excesivo de material rodante. Por ejemplo, si un tren llega justo antes de que otro salga, podrían reutilizarse las mismas unidades, pero el enfoque manual a menudo asigna unidades nuevas, aumentando los costos operativos. Este problema motiva la necesidad de un modelo más eficiente que permita planificar la asignación de unidades de manera óptima.

En este trabajo, se propone un modelo basado en flujos en grafos que permite optimizar la reutilización de unidades disponibles, minimizando la cantidad total de material rodante necesario.

El objetivo del problema de circulación de material rodante es minimizar la cantidad total de unidades necesarias para satisfacer la demanda de pasajeros en un cronograma de servicios ferroviarios. Entonces, se puede formular como uno de optimización combinatoria mediante un modelo de flujos en grafos.

La función objetivo a minimizar es:

$$\min \sum_{(i,j) \in A_{\text{trasnoche}}} f_{ij}$$

sujeta a las siguientes restricciones:

$$\begin{aligned} \sum_{(v,j) \in \delta^+(v)} f_{vj} - \sum_{(i,v) \in \delta^-(v)} f_{iv} &= 0 \quad \forall v \in V \\ f_{ij} &\geq l_{ij} \quad \forall (i,j) \in A_{\text{tren}} \\ f_{ij} &\leq u_{ij} \quad \forall (i,j) \in A_{\text{tren}} \end{aligned}$$

Donde:

- V : Conjunto de eventos (partidas y llegadas) en el cronograma.
- A_{tren} : Conjunto de aristas que modelan los viajes de trenes entre estaciones.
- $A_{\text{trasnoche}}$: Conjunto de aristas que conectan el último evento del día con el primero del siguiente para mantener el stock inicial.
- l_{ij} : Cota inferior de flujo, correspondiente a la cantidad mínima de unidades necesarias para satisfacer la demanda.
- u_{ij} : Cota superior de flujo, correspondiente a la capacidad máxima de cada tren.
- f_{ij} : Flujo asignado a cada arista.

3 Modelo Implementado

Para resolver el problema de circulación de material rodante, implementamos un modelo basado en grafos dirigidos utilizando la biblioteca **networkx** en Python. Este modelo permite representar de manera estructurada los eventos de partida y llegada de los trenes, las transferencias entre estaciones, y las conexiones de trasnoche necesarias para garantizar que el sistema vuelva a un estado inicial equivalente al final del día.

3.1 Construcción del Grafo

El grafo $G = (V, A)$ se construyó con los siguientes elementos:

- **Nodos (V):** Cada nodo representa un evento en el cronograma, que puede ser una partida o llegada de un tren. Los nodos están caracterizados por su estación, tiempo de ocurrencia, y la demanda asociada.
- **Arcos (A):** Los arcos modelan las relaciones entre los nodos y se dividen en tres tipos:
 - **Arcos de tren:** Representan los viajes de los trenes entre estaciones, con una capacidad mínima igual a la demanda del servicio y una capacidad máxima basada en el material rodante disponible.
 - **Arcos de transferencia:** Conectan eventos consecutivos en la misma estación, permitiendo reutilizar material rodante entre trenes.
 - **Arcos de trasnoche:** Conectan el último evento del día con el primero del día siguiente en cada estación, modelando los costos asociados al movimiento nocturno de material rodante.

3.2 Algoritmo Implementado

El problema se resolvió utilizando el algoritmo de **escalamiento de capacidad** (**capacity scaling**) disponible en **networkx**. Este algoritmo es adecuado para problemas de flujo en redes con capacidades y costos asociados, ya que encuentra un flujo de costo mínimo que satisface las demandas y restricciones del sistema.

El flujo mínimo se calcula para:

- Satisfacer las demandas mínimas en los arcos de tren (**lower**) y no superar la capacidad máxima (**upper**).
- Cumplir la conservación de flujo en cada nodo, asegurando que el material rodante que entra y sale de cada estación esté balanceado.
- Minimizar el costo total asociado a los arcos de trasnoche, ya que estos representan el movimiento de material rodante adicional, que es más costoso.

3.3 Justificación

La elección de este enfoque se basa en su capacidad para representar de manera eficiente las restricciones operativas del sistema ferroviario. Cada decisión en la construcción del modelo tiene un propósito específico:

- **Nodos y arcos:** Permiten modelar los eventos temporales y las conexiones entre estaciones de forma granular, facilitando el análisis de la reutilización de material rodante.
- **Arcos de transferencia:** Promueven la reutilización de unidades entre trenes que operan en la misma estación, reduciendo la necesidad de unidades nuevas.
- **Arcos de trasnoche:** Introducen una penalización por el movimiento nocturno, incentivando soluciones que minimicen estos costos.
- **Algoritmo de escalamiento de capacidad:** Este método es eficiente en problemas de optimización de redes, garantizando que se encuentren soluciones óptimas en tiempo razonable incluso para cronogramas complejos. Utilizamos el algoritmo de *capacity scaling* de NetworkX, que es una técnica eficiente para resolver el problema de flujo máximo en redes con capacidades de arista. Este algoritmo permite encontrar la asignación de flujo óptima que minimiza el costo total bajo las restricciones de capacidad de cada arista.

Función de circulación de trenes

```
1: Entrada: Datos data, flag de dibujo dibujo
2: Cargar los datos utilizando la función cargar_datos(data)
3: Crear un grafo dirigido vacío G
4: Obtener parámetros generales:
5:   capacidad = datos["rs.info"]["capacity"]
6:   max_rs = datos["rs.info"]["max_rs"]
7:   estaciones = datos["stations"]
8:   costo_por_unidad = datos["cost_per_unit"]
9: Inicializar un diccionario flujo_por_estacion las estaciones y valores de 0
10: for cada servicio en los datos["services"] do
11:   Obtener las paradas y la demanda del servicio
12:   Crear eventos de salida y llegada con los nodos correspondientes
13:   Agregar nodo de salida al grafo G con la estación de la primera parada,
     el tiempo y la demanda ajustada
14:   Agregar nodo de llegada al grafo G con la estación de la segunda parada,
     el tiempo y la demanda ajustada negativa
15:   Agregar arista de tren entre el nodo de salida y el nodo de llegada:
16:     lower = demanda ajustada
17:     upper = max_rs
18:     costo = 0
19:     color = "azul"
20: end for
21: Crear un diccionario de eventos por estación
22: for cada nodo en el grafo G do
23:   Asignar el nodo a la estación correspondiente en eventos_station junto
     con el tiempo del evento
24: end for
25: for cada estación en eventos_station do
26:   Ordenar los eventos por tiempo
27:   for cada par de eventos consecutivos en la estación do
28:     Agregar arista de traspaso entre los eventos:
29:     lower = 0
30:     upper = infinito
31:     costo = 0
32:     color = "rojo"
33:   end for
34: end for
35: for cada estación con eventos en eventos_station do
36:   if hay eventos en la estación then
37:     Agregar arista de trasnoche desde el último evento al primero:
38:     lower = 0
39:     upper = 1e10
40:     costo = costo_por_unidad para esa estación
41:     color = "verde"
42:   end if
43: end for
44: Resolver el flujo mínimo utilizando el algoritmo de escalado de capacidades
   (capacity_scaling)
45: Mostrar el costo mínimo (flow_cost) y el flujo por arco
46: Si se solicita dibujo, llamar a la función modelo(G, estaciones,
   flow_dict)
```

3.4 Visualización

El modelo implementado incluye una funcionalidad opcional que permite visualizar el grafo de flujo de trenes si el usuario activa el parámetro correspondiente. El grafo resultante se visualiza para analizar las relaciones entre nodos y arcos. En la representación gráfica:

- Los nodos corresponden a eventos de partida y llegada de trenes, organizados por tiempo y estación.
- Los arcos de tren, transferencia y trasnoche se diferencian por colores (azul, rojo y verde, respectivamente) para facilitar su identificación.
- La disposición de los nodos sigue un eje temporal, con un eje vertical que indica las estaciones, proporcionando una visión clara de la circulación del material rodante.

4 Datos

Creamos una nueva instancia de circulación de trenes a partir de los datos proporcionados por la página del Tren Mitre. En primer lugar, se descarga el archivo PDF que contiene información relevante, como las horas de salida y llegada, los orígenes y destinos de las rutas, así como la demanda de cada tramo. Convertimos estos datos en un csv, lo que permite acceder fácilmente a cada uno de los elementos de la tabla y manipularlos según sea necesario.

Una vez cargados los datos, se emplea la función `create_instance`, la cual toma los valores de cada fila del DataFrame y crea una instancia que representa una ruta de tren con todos los parámetros relevantes. Este proceso se repite para cada fila del CSV, generando una lista de instancias que representan todas las posibles circulaciones de trenes basadas en los datos. Estas instancias pueden ser utilizadas posteriormente en el modelo de optimización para analizar la circulación de trenes y realizar los cálculos necesarios según los escenarios de demanda y otras variables definidas en el proyecto.

El archivo con los datos utilizados puede consultarse en el siguiente enlace:

[Enlace al documento PDF.](#)

5 Escenario Adicional

Analizamos un escenario adicional que plantea una decisión estratégica para la operación de los ramales "Retiro-Tigre" y "Victoria-Cardales" de la línea Mitre. La empresa busca determinar si es más eficiente operar ambos ramales con material rodante independiente, como sucede actualmente, o si conviene unificar el uso del material rodante en toda la línea, permitiendo que los trenes compartan

recursos y puedan quedar estacionados durante la noche en cualquiera de las cuatro estaciones involucradas (Retiro, Tigre, Victoria, Cardales).

5.1 Descripción de los Escenarios

Dos configuraciones operativas fueron evaluadas:

- **Escenario 1: Material rodante independiente.** Cada ramal opera de manera separada, utilizando su propio material rodante. Los problemas de circulación de material rodante se resuelven por separado para cada ramal. Esto implica que cada uno tiene su propio material rodante que no se comparte con el otro.
- **Escenario 2: Material rodante compartido.** Ambos ramales comparten el material rodante, funcionando como un único sistema. Se permite que los trenes realicen transferencias entre los ramales y queden estacionados en cualquiera de las estaciones al final del día. Esto fomenta una reutilización más eficiente del material rodante.

5.2 Resultados y Comparación

Para realizar la comparación, se construyó un ejemplo con horarios de trenes simplificados (ver imagen adjunta). Se calcularon las siguientes métricas:

- **Cantidad de material rodante necesario.** En el **Escenario 1**, cada ramal opera de manera independiente. Para el ramal "Tigre-Victoria" se requieren 10 vagones, y para el ramal "Victoria-Cardales" se necesitan 5 vagones, resultando en un total de 15 vagones.

En el **Escenario 2**, los ramales comparten material rodante. Esto permite reutilizar los mismos vagones entre ambos ramales, reduciendo la cantidad total necesaria a 10 vagones.

5.3 Conclusión

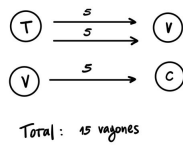
Los resultados indican que el **Escenario 2** es más eficiente en términos de material rodante, ya que permite reducir de 15 a 10 la cantidad total necesaria al reutilizar los recursos entre los ramales.

Cronograma

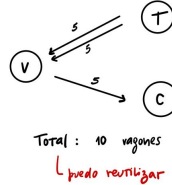
Tigre - Victoria			Victoria - Cardales		
8:30	9:00	5	9:30	10:00	
9:15	9:45	5			

→ uso los 5 vagones

Escenario 1: caso independiente



Escenario 2: caso conjunto



6 Experimentación

6.1 Experimento demandas:

6.1.1 Metodología

Se diseñaron tres escenarios para evaluar el modelo: demanda baja, alta y mixta, con el fin de analizar cómo optimiza el uso de recursos, maneja altos volúmenes de tráfico y se adapta a variaciones en la demanda. Los criterios de evaluación incluyen la relación directa entre demanda y vagones, la adaptabilidad del modelo ante cambios y la eficiencia en los tiempos de ejecución.

6.1.2 Desarrollo

Demanda Baja

Input: demanda 100 y capacidad máxima 100

Flujo por arco: *

Costo mínimo (vagones): 2

Vemos que igualando la demanda con la capacidad se necesitan solo 2 vagones para todo el trayecto, ya que solo necesito uno para la estación de llegada y otro para la de salida, de forma que puedan salir eficientemente. Al tener un vagón en la estación de salida y otro en la estación de llegada, se asegura que los trenes puedan operar de forma continua y fluida.

Demanda Alta Input: demanda 1000 y capacidad máxima 100

Costo mínimo (vagones): 20

Flujo por arco: *

Costo mínimo (vagones): 20

Vemos que igualando la demanda con la capacidad, se necesitan 20 vagones para todo el trayecto, ya que la demanda es de 1000 y la capacidad máxima de los vagones es 100.

Demanda Mixta

Input: demanda 1000 y capacidad máxima 100

Costo mínimo (vagones): 34

Flujo por arco: *

Vemos que para cuando la demanda cambia para cada estación la cantidad de vagones que se necesitan se ajusta para el caso en particular permitiendo una flexibilidad en el modelo. En este caso se necesitan 34 vagones para cubrir todo el trayecto y la distribución de los flujos por arco muestra que los vagones están siendo distribuidos eficientemente en el trayecto.

6.1.3 Tiempos de ejecución

Los tiempos de ejecución demuestran la eficiencia del modelo frente a distintos niveles de demanda. Para demanda baja, fue de **0.0136 segundos**, reflejando su rapidez con menor carga. En demanda alta, aumentó a **0.0269 segundos** debido al mayor volumen de datos, y en demanda mixta llegó a **0.0296 segundos**, influido por la complejidad de la distribución y la variabilidad de flujos.

6.1.4 Análisis de los resultados

El modelo demuestra una notable adaptabilidad frente a distintos niveles de demanda, ajustando eficientemente la cantidad de vagones necesarios en cada caso. En escenarios de baja demanda, optimiza el uso mínimo de recursos, mientras que frente a una demanda alta o mixta, distribuye los flujos de manera eficiente, asegurando la cobertura de todo el trayecto con tiempos de ejecución bajos incluso ante condiciones más complejas. Esto evidencia la flexibilidad y eficiencia del modelo en la gestión de recursos y cálculos operativos.

* el flujo por arco no lo agregamos en el informe ya que este tiene un límite de páginas y esta información puede verse en el código.

6.2 Experimento capacidades:

6.2.1 Metodología

Analizamos cómo afecta la cantidad necesaria para satisfacer la demanda al cambiar la capacidad de los trenes, evaluando un mismo *dataset* para vagones de capacidad para 200 personas y para vagones de capacidad para 400 personas, y así observar si se requiere más o menos frecuencias.

6.2.2 Desarrollo

Capacidad grande

Input: demanda 600 y capacidad máxima 400 Costo mínimo (vagones): 3

Vemos que al poner la capacidad de los vagones en 400, el costo mínimo necesario nos devuelve 3 vagones.

Capacidad chica

Input: demanda 600 y capacidad máxima 200 Costo mínimo (vagones): 2

Luego al disminuir la capacidad a una relativamente chica (de 200) y manteniendo la misma demanda que en el caso con capacidad grande. El costo mínimo devuelto pasa a ser de 2 vagones.

6.2.3 Análisis de los resultados

Lo que nosotras pensamos que pasaría antes de realizar el experimento es que si aumentamos la capacidad de cada tren, se necesitarán menos frecuencias. En cambio, Los resultados muestran que al aumentar la capacidad de los vagones de 200 a 400 personas, el costo mínimo pasa de 2 a 3 vagones. Esto indica que, aunque la capacidad por vagón sea mayor, el modelo ajusta los costos mínimos necesarios para satisfacer los horarios de la demanda.

6.3 Experimento Frecuencia:

6.3.1 Metodología

Evaluamos si es mejor operar trenes pequeños con mayor frecuencia o trenes grandes con menor frecuencia. Para ello comparamos dos escenarios: uno con trenes de 300 pasajeros cada 30 minutos y otro con trenes de 600 pasajeros cada 60 minutos; donde esperamos que los trenes grandes sean más eficientes en términos de costos, pero a la vez mayor tiempo de espera para los usuarios.

6.3.2 Desarrollo

Frecuencia alta

Input: Demanda: 300, Frecuencia: cada 30 minutos, Capacidad máxima: 200
Costo mínimo (vagones): 10

Vemos que al poner la demanda en 300, el costo mínimo necesario nos devuelve 10 vagones.

Frecuencia baja

Input: Demanda: 600, Frecuencia: cada 60 minutos, Capacidad máxima: 200
Costo mínimo (vagones): 9

Luego, al considerar la demanda acumulada por la disminución de la frecuencia, y teniendo en cuenta la misma capacidad, observamos que el costo mínimo disminuye a 9 vagones.

6.3.3 Análisis de los resultados

Los resultados muestran que, con trenes pequeños y mayor frecuencia, el costo mínimo es de 10 vagones, mientras que al operar con trenes grandes y menor frecuencia, el costo mínimo disminuye a 9 vagones. Esto evidencia que los trenes de mayor capacidad pueden ser más eficientes en términos de costos operativos, ya que requieren menos recursos totales, aunque este enfoque podría implicar un mayor tiempo de espera para los usuarios. El análisis destaca cómo el balance entre frecuencia y capacidad afecta la optimización de recursos en el sistema.

7 Conclusión

En este trabajo práctico desarrollamos un modelo de optimización para la gestión de material rodante en sistemas ferroviarios, basado en grafos y flujos de costo mínimo. Este enfoque permitió analizar y comparar distintos escenarios de operación, demostrando que la reutilización eficiente de recursos puede reducir significativamente la cantidad de vagones necesarios, optimizando costos y mejorando la planificación operativa.

A través de los experimentos realizados, evaluamos la capacidad del modelo para adaptarse a variaciones en la demanda, cambios en la capacidad de los trenes y diferencias en la frecuencia de servicio. Los resultados muestran que el modelo responde de manera eficiente a estas condiciones, asegurando un uso óptimo de los recursos en todos los casos.

En términos generales, el modelo proporciona una herramienta poderosa para la toma de decisiones en la gestión de sistemas ferroviarios, ofreciendo una solución eficiente para optimizar la asignación de recursos en contextos operativos diversos.