

Tecnología Digital V: Diseño de Algoritmos

Trabajo Práctico 2

Gestión eficiente de recursos en sistemas ferroviarios



Iván Mondrzak, Federico Peitti y Benjamín Toledo

Junio, 2024

Índice de contenidos

1	Introducción	2
2	Modelo	3
2.1	Escenario adicional	5
3	Implementación	5
3.1	Ejecución	5
4	Experimentación	6
4.1	El modelo en acción	6
4.1.1	Resultados	6
4.2	Demandas	7
4.2.1	Resultados	8
4.3	Capacidades	9
4.3.1	Resultados	9
4.4	Método manual	10
5	Escenario adicional	13
6	Conclusiones	13

1 Introducción

El presente informe tiene como objetivo abordar el problema de circulación de material rodante (*rolling stock*) para poder aplicarlo al caso de la empresa *Lovejoy Railways Services LLC* y mejorar sus servicios.

El problema presentado en el enunciado apunta a minimizar la cantidad de vagones necesarios para ejecutar un servicio diario de trenes (satisfaciendo una demanda dada para cada viaje), determinando cuántos vagones deben utilizarse en cada viaje. El problema toma como datos de entrada un cronograma entre cabeceras del servicio, la demanda asociada a cada servicio en el cronograma (la estimación de pasajeros que harán uso de dicho servicio), y la capacidad de pasajeros que soporta una unidad de material rodante (es decir, un vagón).

Se tienen los siguientes supuestos (recuperados del enunciado):

- Todos los servicios utilizan el mismo tipo de unidad, y éste es único.
- Un servicio puede traspasar unidades a uno o más servicios posteriores, no necesariamente inmediatos, que ocurren en la misma estación.
- Los trasposos entre servicios pueden ser parciales.
- Las unidades no utilizadas en una estación se guardan temporalmente en el patio de maniobras. Inicialmente, asumimos que el mismo tiene capacidad infinita.
- La cantidad de unidades asignadas a un servicio debe satisfacer la demanda. Es posible enviar unidades de más para su reubicación.
- Para incorporar restricciones de infraestructura, los servicios tienen también una limitación respecto al máximo de unidades que pueden circular.
- En el contexto de la empresa, las unidades sólo pueden traspasarse en las estaciones cabecera.

2 Modelo

El servicio de trenes fue modelado a partir de un grafo dirigido con cotas inferiores, superiores y costos en sus arcos, con el objetivo resolver el problema como uno de circulación sobre un digrafo. El modelo utilizado es el que sigue (recuperado del enunciado, donde se toma como referencia un modelo propuesto por Schrijver¹. Dado un cronograma, con demandas y capacidades dadas (demandas para cada servicio, restricciones de la cantidad de vagones que pueden trasladarse por servicio o sobre la capacidad del patio de maniobras), se construye un digrafo $G = (V, A)$ que modele los servicios en una red espacio-tiempo. En G se cumple que:

- Los vértices de V se corresponden con cada **partida** o **llegada** de un tren en una estación.
- Para un servicio que sale del vértice i (partida) y llega al vértice j (llegada), el **arco** $(i, j) \in A$ y se tiene que su cota inferior l_{ij} (unidades para satisfacer la demanda del servicio) y su capacidad u_{ij} (máxima cantidad de unidades que puede llevar un servicio) son los dados por el *input* y su costo $c_{ij} = 0$.
- Para dos eventos i y j (partidas o llegadas) que ocurren uno después del otro en la misma estación (i es el evento anterior a j en la estación dada), el **arco** $(i, j) \in A$ y se tiene que $l_{ij} = 0$, $u_{ij} = \infty$ y $c_{ij} = 0$.
- El último evento (i) del cronograma en una estación se conecta con el primer evento (j) del día en dicha estación, tal que el **arco** $(i, j) \in A$ con $l_{ij} = 0$, $u_{ij} = \infty$ y $c_{ij} = 1$.
- En el contexto de circulación que se busca modelar, los inbalances b son 0 para cualquier vértice.

¹ Alexander Schrijver. Flows in railway optimization.
Nieuw Archief voor Wiskunde, 9(2):126, 2008.

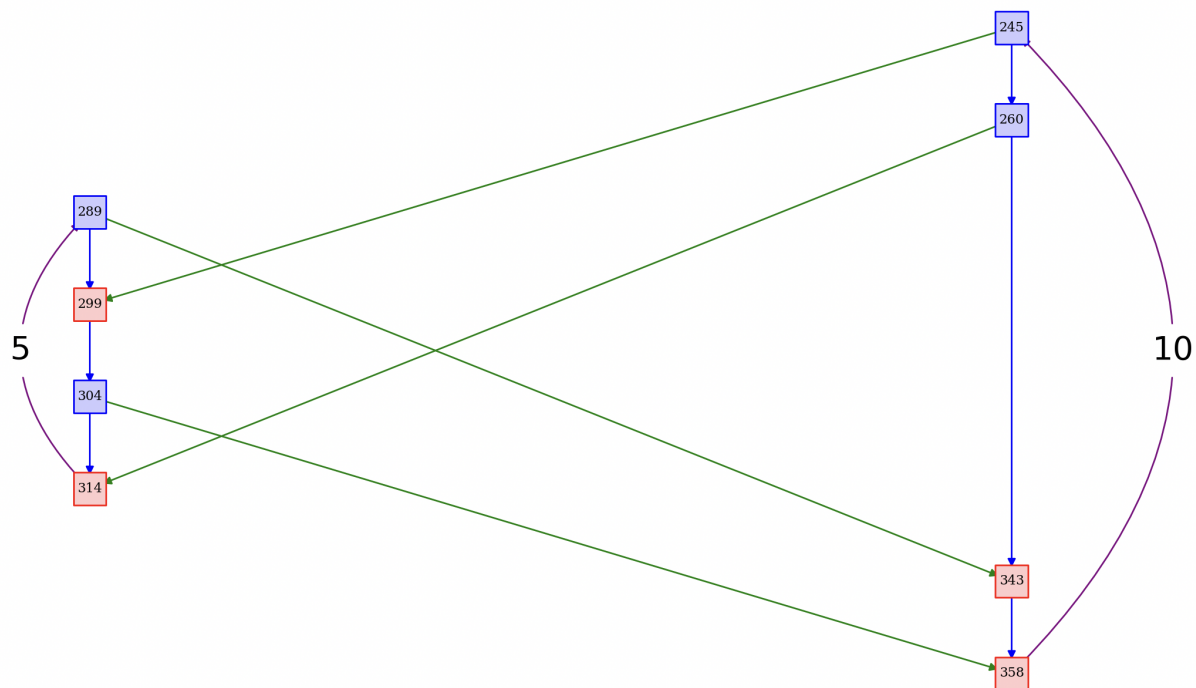


Figura 1: Construcción del modelo para una instancia de ejemplo.

Una vez definido el modelo, para poder resolver el problema de circulación planteado, resulta necesario convertir el problema a uno de flujo de costo mínimo. Para esto, se realiza un cambio de variable en los arcos y vértices del grafo:

- Los arcos (i, j) con capacidad u_{ij} y cota inferior l_{ij} , pasan a tener capacidad $u'_{ij} = u_{ij} - l_{ij}$ y cota inferior $l'_{ij} = 0$.
- El vértice i correspondiente al arco (i, j) con inbalance b_i pasa a tener inbalance $b'_i = b_i - l_{ij}$.
- El vértice j correspondiente al arco (i, j) con inbalance b_j pasa a tener inbalance $b'_j = b_j + l_{ij}$.

Luego, una vez resuelto el problema de flujo de costo mínimo (se tenga un flujo óptimo

x'^* , basta con reconstruir el flujo del problema de circulación tal que $x_{ij}^* = x'_{ij} + l_{ij}$, donde x_{ij}^* es el flujo máximo de costo mínimo que pasa por el arco (i, j) .

2.1 Escenario adicional

En el hipotético caso donde la compañía debe realizar obras sobre alguna de las cabeceras, por lo que la cantidad de unidades que pueden almacenarse durante la noche se ven limitadas, se propone un enfoque posible para abordar la situación. En esta sección solamente se presenta la idea tras el cambio realizado, más adelante en la **sección 5** se discute acerca de su comportamiento y las consecuencias de utilizarlo.

Una forma de encarar el problema es desplazando las unidades que no puedan permanecer en la estación en obras con un servicio adicional previo a la noche y otro a la mañana que las devuelva.

3 Implementación

Se implementó el modelo explicado previamente en Python, utilizando *NetworkX* tanto para construir el grafo (a partir de un archivo JSON con un cronograma) y obtener el flujo de costo mínimo con la función `min_cost_flow()`.

3.1 Ejecución

A continuación se detallan instrucciones de ejecución del modelo implementado. Hay dos variantes posibles que pueden probarse.

- Para el modelo original: `python main.py -i instancia`.
- Para la versión con limitación en una cabecera: `python main.py -i instancia -l indice cota`.

Donde **instancia** es el nombre de alguno de los conjuntos de datos disponibles (**toy** para **toy-instance**, **ret-tig** para Retiro-Tigre ó **mor-onc** para Moreno-Once), e **indice** es 0 o 1 para indicar cuál estación tiene la **cota** (número) que restringe el patio de maniobras.

Índices de referencia: para **toy** y Retiro-Tigre 0 es Tigre y 1 es Retiro; para Moreno-Once 0 es Once y 1 es Moreno.

4 Experimentación

En esta sección se proponen experimentos variando aspectos como la demanda de cada servicio, las condiciones de la empresa y diferentes conjuntos de datos con el fin de estudiar el resultado obtenido en cada caso.

4.1 El modelo en acción

El objetivo de este experimento es estudiar cómo se comporta el modelo con diferentes cronogramas. En particular, se utilizaron tres: los proporcionado en el enunciado, uno tomado a partir del servicio de Retiro a Tigre y otro de ejemplo (*toy*); y finalmente uno real de Buenos Aires. El conjunto utilizado para este último considera los servicios desde Once hasta Moreno y viceversa (solamente considerando las cabeceras), y fue adaptado para poder ser modelado como la instancia de Retiro-Tigre.

4.1.1 Resultados

Instancia	#Servicios	Cabecera 1	Unidades	Cabecera 2	Unidades	Unidades Totales
Toy	4	Tigre	10	Retiro	5	15
Retiro-Tigre	140	Tigre	48	Retiro	0	48
Moreno-Once	500	Once	111	Moreno	65	176

Tabla 1: Unidades necesarias por estación según la instancia.

En cada caso, se cuentan con demandas y capacidades diferentes. Gracias al algoritmo, la empresa *Lovejoy Railways Services LLC* puede saber que para cada uno de estos cronogramas necesita asignar esa cantidad de unidades por estación con el modelo utilizado para poder operar regularmente. Tras esto, el problema cambia a uno de evaluar costos y disponibilidad de mano de obra o material rodante.

4.2 Demandas

Tras ver cómo se comporta el modelo utilizado en diferentes casos, resulta de interés ver su flexibilidad ante diferentes contextos de demanda. Para ello, se proponen tres escenarios diferentes: en primer lugar, uno con demandas aleatorias uniformemente distribuidas entre 0 y 1000; luego, se comparan resultados para demandas constantes; finalmente, tomando como referencia el comportamiento estudiado por Amaya Torres², se construye una distribución de demanda tratando de emular la distribución real del uso del transporte público (las horas pico), como puede observarse a continuación.

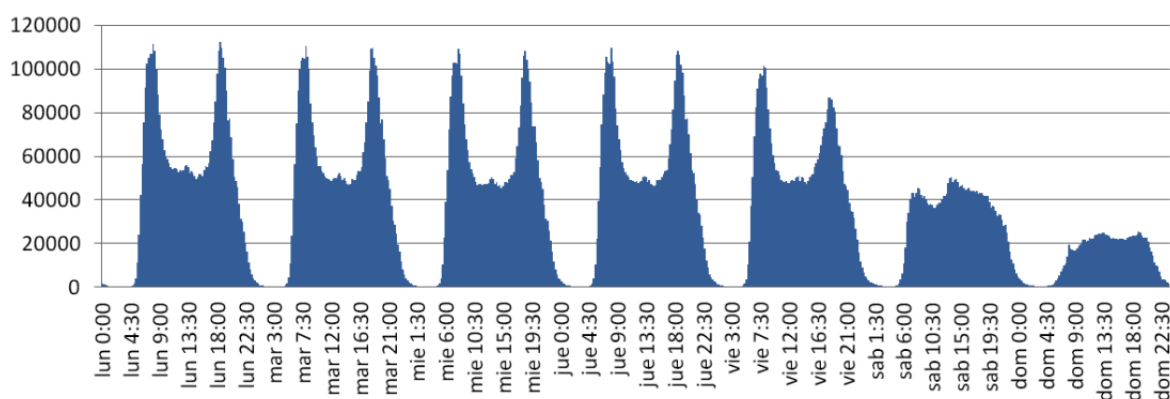


Figura 2: Distribución de la demanda según el día de la semana.

En este caso, se utilizan las instancias de ejemplo (*toy*) y la del servicio Moreno-Once. Tener en cuenta que la capacidad de una unidad rodante es siempre 100 y el máximo de

unidades por viaje es siempre 25. También se considera el caso en el servicio Once-Moreno donde la demanda es muy alta en Once y muy baja en Moreno ("Estación Popular").

4.2.1 Resultados

Instancia	Demanda	Cabecera 1	Unidades	Cabecera 2	Unidades	Total
Toy	Uniforme [0;1000]		15		8	23
	100		2		1	3
	500	Tigre	10	Retiro	5	15
	1000		20		10	30
	Horas Pico		2		1	3
Moreno-Once	Uniforme [0;1000]		153		93	246
	100		23		13	36
	500	Once	115	Moreno	65	180
	1000		230		130	360
	Horas Pico		108		52	160
	Estación Popular		111		69	180

Tabla 2: Unidades necesarias por estación según la demanda.

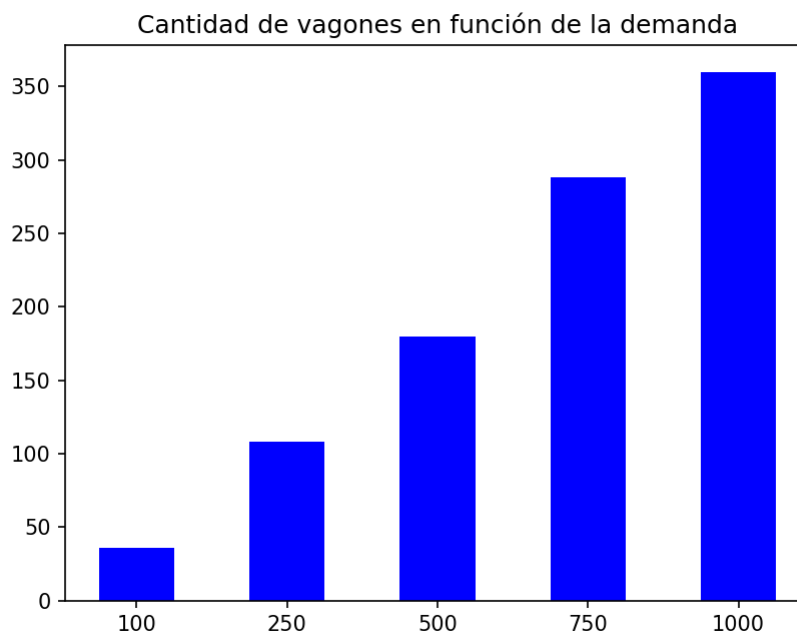


Figura 3: Cantidad de unidades necesarias (servicio Moreno-Once) según la demanda.

Es claro a partir del gráfico que a mayor demanda, se necesitarán más unidades para satisfacerla. Tanto el caso de las horas pico como el de la estación popular parecen (en el caso del servicio Moreno-Once) no impactar mucho en la solución obtenida, seguramente por la robustez del cronograma que debe tener en cuenta estos factores a la hora de construirse.

4.3 Capacidades

En este experimento, se analiza cómo impacta en las soluciones las condiciones de la empresa (capacidad de los vagones y límite de unidades de material rodante por servicio). Las demandas de cada servicio siguen una distribución uniforme en el rango $[100,700]$.

4.3.1 Resultados

Con límite 50:

Instancia	Capacidad	Cabecera 1	Unidades	Cabecera 2	Unidades	Total
Moreno-Once	50	Once	208	Moreno	123	331
	500		33		19	52

Tabla 3: Unidades necesarias por estación según la capacidad de un vagón.

Con capacidad 100:

Instancia	Límite	Cabecera 1	Unidades	Cabecera 2	Unidades	Total
Moreno-Once	7	Once	119	Moreno	57	176
	70		111		65	176

Tabla 4: Unidades necesarias por estación según el límite de vagones por servicio.

Como puede observarse, por más que sean muy pocas observaciones, tiene sentido que cuanto más gente entre en un vagón, menos vagones se necesiten para transportar cierta cantidad de personas.

En el caso del límite de vagones, ya es más delicado intentar llegar a una conclusión en base a tan pocas observaciones. En general, en principio el factor limitante depende

de restricciones físicas y de seguridad previamente definidas. Además, en este sentido la demanda también está "acotada" en el modelo ya que si se requieren más vagones de los permitidos no hay nada que pueda hacerse (por más que se tengan almacenados infinitos vagones, si se necesitan 2 y sólo pueden transportarse de a una unidad no hay nada que pueda hacerse).

4.4 Método manual

Actualmente *Lovejoy Railways Services* utiliza un sistema manual para resolver este problema. Su diseño está explicado en el enunciado. A continuación se lo analiza para estudiar su eficiencia y se presenta un caso a modo de ejemplo que evidencia la ineffectividad del método. En cambio, este mismo caso ejecutado con el algoritmo implementado en **sección 2** presenta una mejor y óptima solución.

Téngase el siguiente cronograma:

service id	hora	origen	tipo	hora	destino	tipo	demanda
1	100	Retiro	D	154	Tigre	A	100
2	160	Tigre	D	214	Retiro	A	400
3	215	Retiro	D	269	Tigre	A	200
4	270	Tigre	D	324	Retiro	A	500

Donde la capacidad de pasajeros del material rodante es de 100 y el máximo de material rodante a enviar en un viaje es 25.

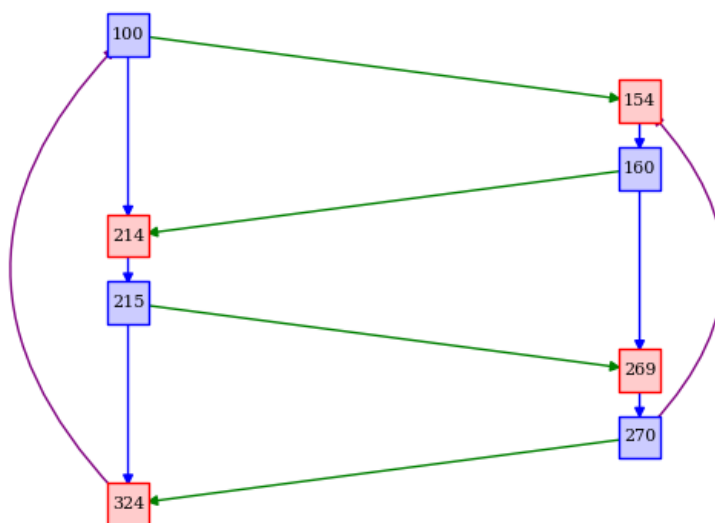


Figura 4: Grafo correspondiente al cronograma de ejemplo.

Dadas estas condiciones, utilizando el método manual, se obtiene el siguiente resultado:

1. (1): se envía 1 vagón desde Retiro a Tigre. Aumentando las unidades nuevas en 1.
2. (2): se envían 4 vagones desde Tigre a Retiro. Utilizando 1 vagón del stock de Tigre y aumentando las unidades nuevas en 3.
3. (3): se envían 2 vagones desde Retiro a Retiro. Utilizando 2 vagones del stock de Retiro.
4. (4): se envían 5 vagones desde Tigre a Retiro. Utilizando 2 vagones del stock de Tigre y aumentando las unidades nuevas en 3.

De este modo, la cantidad total de vagones necesarios para completar todos los servicios es 6 unidades en Tigre y 1 unidad en Retiro.

Sin embargo, utilizando el modelo propuesto en **sección 2** se obtiene un resultado mejor, utilizando solamente 5 unidades que parten desde Tigre. Lo cual indica que claramente el método manual que venía utilizando la empresa no es óptimo.

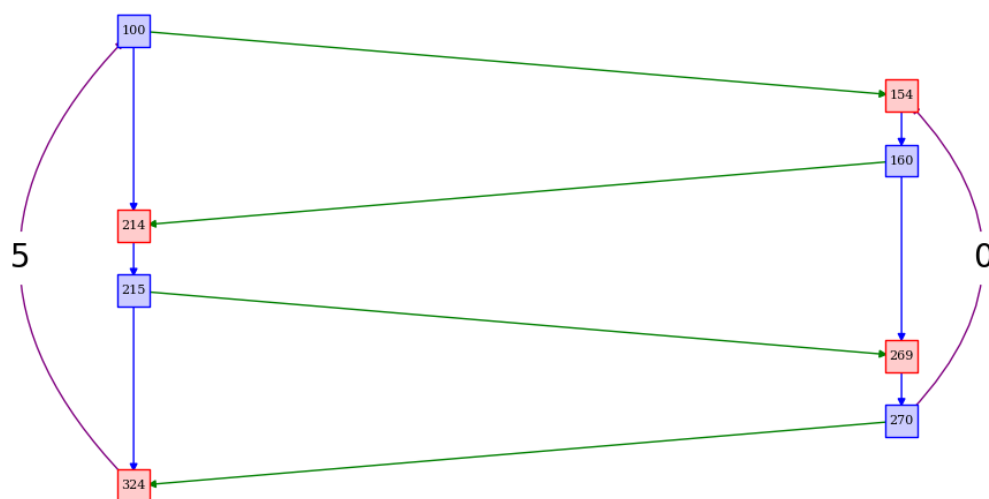


Figura 5: Solución utilizando el modelo implementado.

Notar que esto ocurre porque el servicio 3 envía la mínima cantidad de vagones para cumplir la demanda (2 unidades), lo que resulta un punto diferencial clave entre ambos enfoques. Por lo que ahora hay 2 unidades en el stock de Tigre y el servicio 4 debe agregar 3 unidades para satisfacer la demanda.

Una decisión más inteligente conociendo todo el cronograma hubiera sido enviar todos los vagones disponibles en el servicio 3 de forma que el servicio 4 solo tuviera que agregar 1 unidad nueva en lugar de 3.

En general, en situaciones en las que un servicio del tiempo i tiene stock que supera su demanda y menor demanda que un servicio de la estación opuesta de tiempo j tal que $j > i + t$ donde t es el tiempo que demora en completarse un servicio, suceden situaciones como la ilustrada previamente, en las que no se aprovecha al máximo el material rodante.

Adicionalmente, se cuenta con una implementación en Python del enfoque manual para probar sobre las instancias reales (`metodo_manual.py`). En ambos casos de los cronogramas reales, el método manual requiere más unidades y también servicios adicionales para reubicar material rodante.

5 Escenario adicional

En esta sección se retoma el estudio de la implementación del escenario adicional con el enfoque explicado en la **sección 2**. Este incorpora nuevos servicios a la noche y a la mañana para aprovechar que la otra estación cabecera sí tiene capacidad infinita. De esta forma, la cantidad total de vagones necesarios sigue siendo la misma, sólo que la cantidad que sobre en la estación limitada se trasladará a la otra cabecera durante la noche, para ser regresada a la mañana y poder comenzar con el servicio regularmente. En la práctica, esto implica un costo al tener que operar servicios adicionales para la distribución de los vagones, pero permite que se pueda operar sin problemas mientras se realizan las obras por la noche.

6 Conclusiones

En síntesis, mediante un modelado inteligente que aproveche el contexto donde se sitúa el problema, su resolución resulta más sencilla de lo que parece. Una amplia cantidad de variantes del problema abordado en este trabajo pueden resolverse de forma similar, modificando ligeramente el modelo de acuerdo al enfoque que se busque.

En cuanto al algoritmo propuesto, este demostró ser óptimo por aprovechar al máximo los recursos frente a los distintos escenarios provistos, resultando en una clara mejora frente al método utilizado por la empresa originalmente.

Por otra parte, la experimentación abierta frente a este modelo pero sobre todo para el problema en general es extremadamente extensa, en particular en lo que respecta a generar un cronograma estudiando la demanda y el comportamiento de los usuarios del sistema de transporte. Así, en un trabajo de índole mucho más profunda podrían estudiarse estructuras de cronograma que no sólo minimicen costo satisfaciendo condiciones, sino que también puedan apuntar a mejorar aspectos más complejos como satisfacción y situaciones extraordinarias (por ejemplo, con feriados, eventos o épocas del año).