

TRABAJO FIN DE MÁSTER

LOCALIZACIÓN DE UNA LÍNEA DE METRO:
EFECTOS DE LA SOBRECUPACIÓN



María Aránzazu Manzanares Abásolo

Tutores: D. Juan Antonio Mesa López-Colmenar
D. Federico Perea Rojas-Marcos

Doble Máster en Enseñanza Secundaria y Matemáticas
Departamento de Matemática Aplicada II

Curso 2022-23

Resumen.

El problema de diseño de redes de tránsito rápido (metros, cercanías, autobuses en plataforma reservada y cerrada, etc) consiste en, conocida una demanda de movilidad en una zona urbana o metropolitana, determinar las estaciones y las conexiones entre las mismas. El tema suele ser muy complejo, ya que intervienen distintos especialistas (ingenieros, economistas, geólogos, etc.), autoridades y grupos sociales. Los distintos agentes que intervienen en la planificación se pueden agrupar en tres categorías: agencias de transporte, usuarios y empresas (constructoras y explotadoras). Esta gama de agentes a menudo implica la necesidad de tener en cuenta distintos criterios que deben coexistir dentro de un mismo proyecto. Las expectativas son distintas: así como las autoridades representadas por las agencias de transporte desean que la utilidad de las redes sea lo más amplia posible, los usuarios buscan minimizar el tiempo dedicado al viaje, y las empresas tienen como prioridad la optimización de costes de construcción y de operación para maximizar sus beneficios.

El método empleado por las constructoras para localizar el trazado de una línea de tránsito rápido consiste en determinar corredores por los que circula el tráfico, unirlos y considerar un pequeño número de alternativas para posteriormente compararlas desde distintos puntos de vista. Los modelos de programación matemática permiten considerar un gran número de opciones y, en particular, algunas que no reproducen los patrones de movilidad existentes pero que resultan ventajosos a los usuarios. En el diseño de redes de metro hay que tener en cuenta que el modo de tránsito rápido va a competir con otros modos de transporte como son los medios privados y públicos. Normalmente, el usuario escoge el medio que le reporte mayor utilidad, dentro de la cual se pueden considerar distintos atributos: el precio, la seguridad o el confort. No obstante, debido a que el usuario suele realizar el mismo viaje diario, el más importante suele ser el tiempo de recorrido. Por tanto, los problemas de diseño de redes de tránsito rápido son muy complejos al intervenir tanto distintos agentes como la conducta de los usuarios, dando lugar a problemas NP-duros desde el punto de vista computacional.

Por otra parte, la construcción de líneas de metro supone un coste muy elevado y no siempre es posible satisfacer completamente la demanda por lo que, incluso recién inauguradas, sufren la congestión y el amontonamiento de pasajeros. Los modelos matemáticos considerados hasta el momento asumen la competencia con otros modos de transporte en condiciones de no congestión. De hecho, la demanda real captada puede variar al degradarse la utilidad de los posibles usuarios debido a la congestión. En este sentido, se pueden clasificar en cuatro grupos los motivos de degradación de la utilidad esperada: i) congestión y amontonamiento en los accesos o en los andenes ii) espera de un segundo, tercer o sucesivos trenes iii) tiempos de parada en las estaciones mayores que los nominales debido a la gran afluencia de viajeros iv) sobreocupación en los vagones. Los efectos de estos motivos suelen aparecer en horas punta, con ocasión de eventos especiales, y cuando ocurren perturbaciones a la operativa nominal. Por otra parte, el intervalo entre dos trenes consecutivos y el tamaño de estos tiene una cota inferior (dos o tres minutos en el mejor de los casos, y de seis a diez vagones), por lo que las condiciones técnicas no permiten una ampliación de la capacidad por hora de la línea. Sin embargo, puede ser interesante, a la hora de realizar un análisis coste/beneficio, efectuar un estudio de la sensibilidad sobre la pérdida de usuarios respecto a la variación en la frecuencia. En áreas metropolitanas densamente pobladas, los estudios empíricos han determinado que la sobreocupación dentro de los vehículos de tránsito ha empeorado cada vez más en los últimos años. El hacinamiento crónico dentro de los vehículos no sólo es causado por la falta de infraestructura física, sino también por la provisión inadecuada de servicios o por una planificación basada en un modelo matemático inadecuado.

En este trabajo se va a considerar el problema de la localización del trazado de una línea de tránsito rápido que maximice el número de viajes captados, teniendo en cuenta los distintos efectos de la sobreocupación. A la hora de diseñar la localización de la línea, se van a tener en cuenta los condicionantes anteriormente descritos. El principal objetivo de un sistema de tránsito rápido es mejorar la movilidad de la población. Idealmente, una red debe estar diseñada para proporcionar tiempos de viaje cortos a un elevado número de pasajeros, respetando restricciones técnicas, legislativas y presupuestarias. Para ello, se van a introducir funciones que modelen cada uno de esos efectos. El modelado se va a efectuar mediante programación matemática entera mixta y, dada la complejidad computacional, se realizará una experiencia numérica de tamaño pequeño.

Abstract.

The design problem for rapid transit networks (metros, commuter trains, buses on a reserved and closed platform, etc.) consists of knowing a demand for mobility in an urban or metropolitan area, determining the stations and the connections between them. The issue is usually very complex, since different specialists (engineers, economists, geologists, etc.), authorities and social groups are involved. The different agents involved in planning can be grouped into three categories: transport agencies, users and companies (builders and operators). This range of agents often implies the need to take into account different criteria that must coexist within the same project. The expectations are different: just as the authorities represented by the transport agencies want the usefulness of the networks to be as wide as possible, the users seek to minimize the time spent on the journey, and the companies have the optimization of construction and operating costs as a priority to maximize the profits.

The method used by construction companies to locate the layout of a rapid transit line consists of determining corridors through which traffic circulates, joining them and considering a small number of alternatives to later compare them from different points of view. Mathematical programming models allow us to consider a large number of options and, in particular, some that do not reproduce existing mobility patterns but are advantageous to users. In the design of metro networks, it must be taken into account that the rapid transit methods will compete with other modes of transportation such as private and public means. Normally, the user chooses the mode that provides the greatest utility, within which different attributes can be considered: price, safety or comfort. However, since the user usually makes the same daily trip, the most important is usually the travel time. Therefore, rapid transit network design problems are very complex, as different agents and user behavior intervene, giving rise to NP-hard models from the computational point of view.

On the other hand, the construction of metro lines entails a very high cost and it is not always possible to fully satisfy the demand, which is why, even just inaugurated, they suffer from congestion and overcrowding of passengers. The models considered so far assume competition with other transportation modes under non-congested conditions. In fact, the real demand captured can vary as the usefulness of potential users is degraded due to congestion. In this sense, the reasons for the degradation of the expected utility can be classified into four groups: i) congestion and crowding at the entrances or on the platforms ii) waiting for a second, third or successive trains iii) stop times at the stations greater than nominal due to the large flow of travelers iv) overcrowding in wagons. The effects of these reasons usually appear at peak times, on the occasion of special events, and when disturbances to nominal operations occur. On the other hand, the interval between two consecutive trains and the size of these has a lower limit (two or three minutes in the best of cases, and from six to ten wagons), so the technical conditions do not allow an extension of the hourly capacity of the line. However, it may be interesting, when carrying out a cost/benefit analysis, to carry out a study of the sensitivity of the loss of users with respect to the variation in frequency. In densely populated metropolitan areas, empirical studies have found that overcrowding within transit vehicles has gotten worse in recent years. Chronic overcrowding inside vehicles is not only caused by the lack of physical infrastructure, but also by the inadequate provision of services or by planning based on an inadequate mathematical model.

In this work, the problem of locating the layout of a rapid transit line that maximizes the number of trips captured, taking into account the different effects of overcrowding, will be considered. When designing the location of the line, the conditions described above will be taken into account. The

main objective of a rapid transit system is to improve the mobility of the population. Ideally, a network should be designed to provide short travel times to a high number of passengers, while respecting technical, legislative and budgetary constraints. To do this, functions that model each of these effects will be introduced. The modeling will be done through integer or mixed-integer mathematical programming and, given the computational complexity, a small numerical experiment will be carried out.

Agradecimientos.

*A mis tutores
Juan Antonio Mesa y
Federico Perea
por su permanente apoyo
a lo largo de la realización
de este trabajo.
A Alejandro,
por estar siempre ahí.*

Índice general

1. Introducción.	9
2. Diseño de redes de tránsito rápido.	15
2.1. Construcción de la red: proyecto, objetivo y evaluación.	18
2.2. Modelado algorítmico para redes de tránsito rápido.	20
2.2.1. Modelos de localización de redes.	21
2.2.2. Modelos de localización de líneas.	24
2.3. Modelos de amontonamiento en redes de tránsito rápido	26
2.3.1. Contextualización.	27
2.3.2. Paradigma matemático.	28
2.3.3. Ideas principales sobre amontonamiento.	34
3. Modelo para el diseño de una red de tránsito rápido.	35
3.1. Modelo matemático sin sobreocupación.	35
3.1.1. Datos y notación.	35
3.1.2. Variables.	36
3.1.3. Función objetivo y restricciones.	36
3.2. Efectos del amontonamiento de pasajeros.	39
3.2.1. Amontonamiento en los vehículos.	40
3.2.2. Amontonamiento en los andenes.	43
3.2.3. Incremento de tiempos de espera.	44
3.2.4. Incremento del tiempo de parada.	47
3.3. Modelo matemático con sobreocupación.	48
3.3.1. Datos y notación.	48
3.3.2. Variables.	48
3.3.3. Función objetivo y restricciones.	49
4. Entorno tecnológico.	51
4.1. Lenguaje, interfaz y editor.	51
4.2. Software de optimización y estructura del programa.	53
4.2.1. Pyomo	53
5. Resultados experimentales.	55
5.1. Sin efecto de sobreocupación.	58
5.2. Considerando el efecto de la sobreocupación.	60
5.3. Red ampliada.	64
6. Ampliación del problema y posibles horizontes de estudio.	69
7. Resumen y conclusiones principales.	71
A. Anexos.	77
A.1. Método de los planos de corte de Gomory.	77

Índice general

A.2. Dominio de aplicación de diferentes solvers.	79
A.2.1. Herramientas de optimización alternativas.	80
B. CÓDIGOS EMPLEADOS EN LENGUAJE PYTHON	83

1. Introducción.

En los últimos años, se ha producido una tendencia hacia el incremento de las inversiones en proyectos de transporte público y de logística. Paralelamente, dichas inversiones han ido a la baja en proyectos de construcción de carreteras. Este cambio se ha producido para dar respuesta a las preocupaciones medioambientales por el hecho evidente de que el combustible fósil necesario para el transporte privado no es una opción sostenible.

En este contexto, muchas ciudades han construido nuevos sistemas de metro, han expandido los existentes o han modernizado los antiguos, incrementándose en más de un 50% las redes existentes en el mundo construidas en la primera década de los años 2000. Existen distintas definiciones para un sistema de metro, siendo un rasgo distintivo de este medio de transporte su independencia del tráfico por carretera o del tránsito de peatones. La más aceptada entre los especialistas en metro es: un metro propiamente dicho tiene plataforma reservada y cerrada, excepto las estaciones. Sin embargo, si hay tramos con cruces o acceso en tramos, se dice metro ligero¹. Aunque normalmente los metros circulan bajo tierra, en algunas ciudades como Londres la parte suburbana de la red se construye sobre la superficie, y en ocasiones comparte infraestructuras con los principales raíles de la red ferroviaria local. Es más, resulta frecuente encontrar sistemas en superficie y otras tipologías como metros luminosos, pre-metros, trenes de cercanías, sistemas de tránsito rápido², monorraíles, etc. En ciudades de Asia y América son frecuentes los tramos en viaducto. Por comodidad nos referiremos a todos estos sistemas como *redes de tránsito rápido*. Es en este marco donde se va a contextualizar el presente trabajo, consistente en un modelo para la localización del trazado de una línea de tránsito rápido.

Los proyectos de planificación de líneas de tránsito rápido a largo plazo son muy caros, llenos de incertidumbre y sujetos a calendarios que suelen sobrepasar los presupuestos. El número de actores involucrados incluye políticos, planificadores urbanos, ingenieros, consultores y grupos de ciudadanos. Es frecuente hallar conflictos de intereses entre ellos, dado que cada uno tiene sus prioridades. Si además introducimos el punto de vista matemático, el problema se complica, ya que es necesario imponer restricciones que con frecuencia resultan difíciles de formular.

Adicionalmente, los proyectos de diseño de redes de tránsito rápido están fuertemente influidos por el impacto urbano debido a las obras, los posibles cambios en los modelos de tráfico y el objetivo ecológico de reducir la polución en el aire. Por tanto, en este marco no resulta realista abordar la solución del problema mediante aplicaciones directas de modelos matemáticos estándar y algoritmos de optimización. Diversos autores (Laporte et al. [2011a]) han examinado recientemente decenas de proyectos de tránsito rápido, llegando a la conclusión de que en la mayoría de los casos los planificadores utilizaron poca o ninguna investigación operativa. Además, es destacable el hecho de

¹En España, son metros puramente dichos los de Madrid, Barcelona, Valencia, Bilbao, Sevilla y Palma de Mallorca. El de Málaga es un metro ligero, y los de Granada y Alicante se dice que son “metro-style”.

²El término *rapid transit systems* engloba todos los sistemas de transporte rápidos y masivos en aglomeraciones urbanas: metro, metro ligero, cercanías, BRT (Bus Rapid Transit), *light rails*, etc.

que la mayoría de la literatura orientada académicamente al análisis de sistemas y de operaciones falla en las aplicaciones propuestas, principalmente por utilizar modelos excesivamente teóricos y frecuentemente con una formulación incorrecta de las funciones objetivo (Laporte et al. [2011b]). Sin embargo, todo esto ilustra la necesidad y el potencial del uso de métodos de búsqueda de operaciones en planificación. El papel de dichos métodos es el uso de las técnicas de optimización de sistemas enfocado a construir y calcular soluciones potenciales que sean sometidas a una posterior toma de decisiones, teniendo en cuenta que la naturaleza del problema es multiobjetivo.

En el Capítulo 2 describiremos algunos métodos de búsqueda de operaciones desarrollados recientemente y que resultan de interés para configurar redes y localizar estaciones.

Para ello, es preciso introducir algunas nociones previas.

- **Teoría de grafos.**

En matemáticas, la representación clásica de las redes de transporte recae sobre la conocida *teoría de grafos*. Este campo de investigación nació en 1736, cuando Leonard Euler (1707-1783) trató de resolver el famoso problema de los puentes de Königsberg (ver Figura 1.1). Fue a partir de la década de 1980 cuando la teoría de grafos empezó a aplicarse al estudio y optimización de tránsitos urbanos.

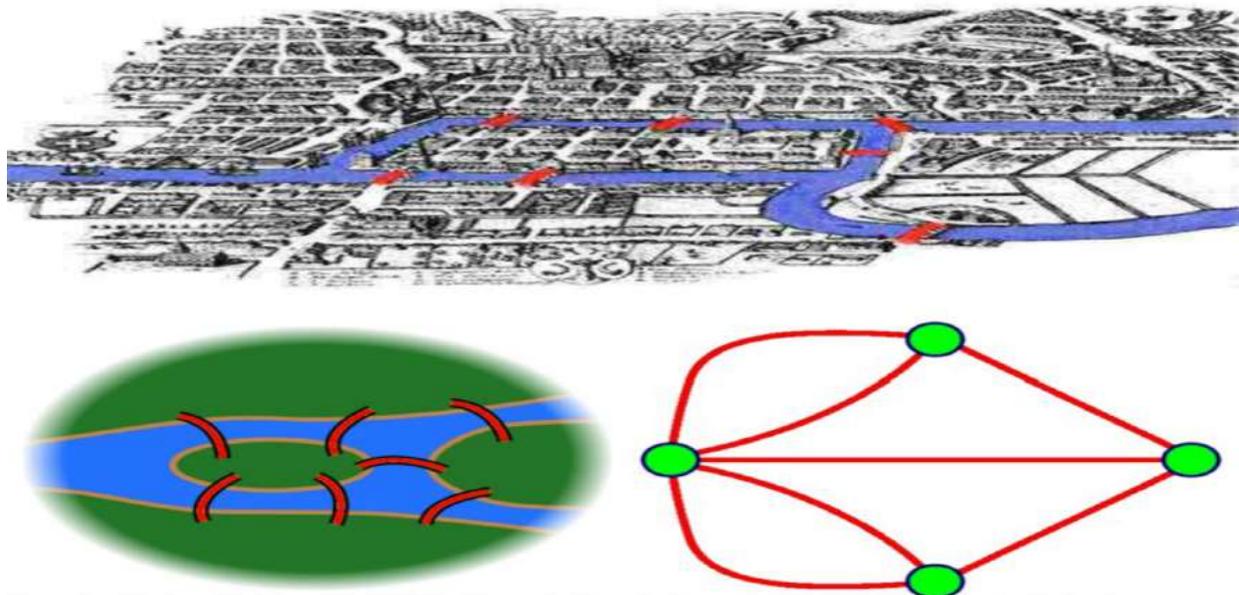


Figura 1.1.: Esquema del problema de los puentes de Königsberg, resuelto por Euler en 1736.
Fuente: Wikimedia.

Un grafo $G(N, E)$ constituye un objeto abstracto, utilizado para dar significado a dos conjuntos: el conjunto N , que representa un número finito de elementos denominados *nodos*, y el conjunto E formado por *aristas*, que constituyen las uniones sin dirección por pares de elementos de N . Podemos denominar n y m al número de elementos de N y E , respectivamente. Dependiendo del tipo de relaciones que se establezcan en el conjunto E , un grafo se puede clasificar como *dirigido* o *no dirigido*. En un grafo no dirigido, $G(N, E)$, cada elemento de E denominado arista representa una conexión entre dos nodos distintos i, j de N . Una arista se suele denominar por $\{i, j\}$ o e_{ij} .

Por otra parte, en un grafo dirigido $G(N, A)$, el orden en cada elemento de A resulta ser un aspecto importante: un elemento (i, j) de A se denomina *arco* y tiene un origen $i \in N$ y un destino $j \in N$.

Para estudios como los del presente trabajo, resulta muy útil asociar valores numéricos (*pesos* o *costes*) a las aristas o nodos de un grafo, sea éste dirigido o no. Los pesos definidos sobre aristas se pueden expresar como una función $W : E \rightarrow \mathbb{R}$, que asigna un peso $w(e)$ (y de manera equivalente, un peso $w(a)$ a cada arista $e \in E$). Según el contexto, estos pesos describirán diferentes aspectos, como costes económicos, tiempos de viaje, distancias, capacidades, etc.

• Tipos de algoritmos.

Los problemas de optimización se resuelven mediante algoritmos. Una posible clasificación de los algoritmos existentes es a partir del grado de certeza de optimalidad de sus soluciones. Así, tenemos:

- Algoritmos exactos, son aquellos que devuelven una solución óptima garantizada, es decir, se demuestra que no existe otra solución con mejor valor objetivo. Para problemas muy complejos no son una opción viable, dado que no son capaces de devolver soluciones en tiempos razonables.
- Algoritmos de aproximación, son aquellos que devuelven una solución factible, no necesariamente óptima, pero dan una cota de la distancia máxima entre el valor de esa solución factible encontrada y el valor de la solución óptima (la solución óptima es desconocida). En general, son algoritmos iterativos para los que se ha demostrado la convergencia a la solución. Suelen ser más rápidos que los algoritmos exactos.
- Algoritmos heurísticos, son aquellos que devuelven una solución factible, sin ningún tipo de estimación de la distancia entre dicha solución y la solución óptima. Aún así, si están bien construidos, experimentalmente se puede demostrar que devuelven soluciones de gran calidad. Suelen ser más rápidos que los algoritmos exactos y los algoritmos de aproximación.

Los algoritmos propuestos en este trabajo son exactos, más concretamente modelos de programación matemática lineal entera mixta.

A diferencia del algoritmo exacto que conduce a una solución óptima, o técnicas estadísticas cuyos resultados presentan una cierta convergencia, el propósito de una heurística es obtener una solución de alta calidad. En este contexto, una *metaheurística* es una heurística con propósito general, es decir, se puede aplicar a diferentes problemas. En otras palabras, se trata de una combinación de una heurística con un algoritmo exacto. Este término fue acuñado por Glover en la década de los 80' (Glover [1986]), cuando propuso el procedimiento de búsqueda tabú.

Algunas de las principales razones para usar heurísticas son:

- Permiten la resolución, total o parcialmente, de problemas NP-duros cuando la instancia es de tamaño medio o grande.
- En algunos casos, es mejor obtener una solución aproximada para un problema exacto que una solución exacta para un problema aproximado.
- Las heurísticas son más flexibles para objetivos o restricciones complejos.
- Los algoritmos heurísticos son competitivos cuando hay poco tiempo disponible, como sucede con los algoritmos en línea.

CAPÍTULO 1. INTRODUCCIÓN.

- El procedimiento general de búsqueda de un algoritmo consiste en partir de una solución inicial, y aplicar iterativamente una transformación hasta llegar a la situación buscada.
- Un procedimiento de búsqueda cuando se utiliza una heurística de trayectoria consiste en repetir la selección de un movimiento en el espacio de soluciones que transforma la situación actual hasta que se cumpla un criterio de parada.

• Diseño de redes.

El diseño de una red de tránsito rápido es un proceso realmente complejo. Dentro del ámbito de la investigación operativa, las numerosas técnicas tomadas de los campos del diseño de redes y las metaheurísticas pueden respaldar los procesos de planificación al proporcionar soluciones factibles que cumplan parcial o totalmente con los criterios de cobertura y coste del proyecto.

Las líneas de metro suelen ser subterráneas. No obstante, en muchas ciudades como Londres, Seúl o Sevilla, la parte suburbana de la red (es decir, aquellos tramos alejados de los núcleos principales de población) es superficial, compartiendo en ocasiones infraestructura con la red ferroviaria principal. Además, las redes ligeras de transporte que circulan completamente sobre la superficie son bastante habituales en urbes que ofrecen una amplia gama de medios de transporte, existiendo metros ligeros, monorraíles, trenes de cercanías, etc. Por conveniencia, en este trabajo nos referiremos casi exclusivamente a todos estos sistemas como *redes de tránsito rápido*.

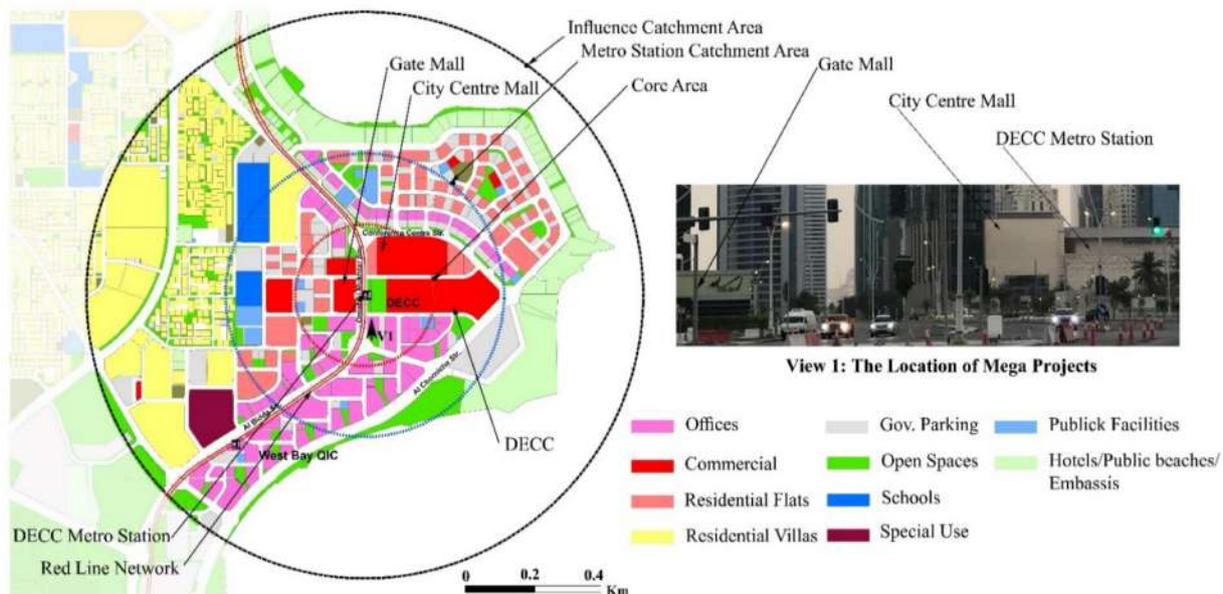


Figura 1.2.: Desglose de los distintos elementos de un área urbana. La ubicación de la parada de metro, previo estudio de su área de influencia, resulta fundamental para conocer el impacto que ésta causará sobre la población. Fuente: Jamaledin et al. [2022].

Habitualmente, cualquier proyecto para el diseño e implantación de redes de tránsito rápido se enfrenta a problemas de diversa índole, abarcando desde el posible impacto en el tejido urbano y en zonas periféricas, hasta cambios en patrones de tráfico, reducción de la contaminación en el aire, necesidad y disponibilidad de trabajadores cualificados, competencia con otros medios de transporte, normativa territorial, etc. Es por ello por lo que la solución a estos problemas va mucho más allá de una aplicación directa de modelos matemáticos y su resolución mediante cualquier herramienta

de optimización, constituyendo una de las líneas de trabajo con mayor dinamismo dentro de la investigación operativa en la planificación del tránsito en redes. Dados los numerosos actores y objetivos implicados en este tipo de problemas, es sensato orientar las técnicas de optimización conocidas hacia la construcción y evaluación de soluciones potenciales para los proyectos, con el fin de que, en última instancia, la administración responsable tome la decisión más adecuada disponiendo también de un riguroso criterio cuantitativo (Gendreau et al. [1995]). Los nodos de origen y destino suelen ser áreas densamente pobladas. En la planificación de estas redes, se esbozan líneas que cubran las principales zonas de movilidad, al mismo tiempo que proporcionan un grado suficiente de conectividad entre las líneas de la red. Un ejemplo lo encontramos en la Figura 1.2.

Para evaluar las configuraciones de tránsito rápido, se hace uso de diversos índices, a fin de disponer de una información cuantitativa completa y fidedigna. Si bien intuitivamente la extensión geográfica de una red viene dada por la longitud total de las líneas y el número de estaciones componentes, existen factores adicionales a tener en cuenta, como el área de influencia de cada estación. Este área no siempre estará definida exclusivamente por el tráfico de peatones, pues existe un amplio abanico casuístico, desde estaciones con servicios de estacionamiento para automóviles y bicicletas, hasta conexiones con autobuses o trenes de cercanías.

Numerosas medidas están relacionadas con la topología de la red $G = (N, E)$, siendo N es el conjunto de nodos y E el conjunto de aristas. Algunas de éstas son:

- Número de ciclos mínimos (sin incorporar ningún otro ciclo), C :

$$C = |E| - |N| + 1 \quad (1.1)$$

- Disponibilidad de ciclo, definida como la relación entre C y el valor más grande que podría tomar para una red con $|N|$ nodos y $|E|$ aristas, α :

$$\alpha = \frac{|E| - |N| + 1}{2|N| - 5} \quad (1.2)$$

- Complejidad de la red, β :

$$\beta = \frac{|E|}{|N|} \quad (1.3)$$

- Indicador de conectividad, relación de $|E|$ al número máximo de aristas que podrían existir en una red plana con nodos $|N|$, γ :

$$\gamma = \frac{|E|}{3(|N| - 2)} \quad (1.4)$$

- Sencillez del servicio, δ , definida como el número de rutas de origen/destino (O/D) que se pueden recorrer sin transbordos.

Además de los índices anteriores, autores como Laporte, Mesa, Ortega y Perea (Laporte and Mesa [2015], y las referencias contenidas) han hecho uso en sus trabajos de índices alternativos en el ámbito de las redes de transporte, como por ejemplo el *índice de efectividad pasajero/red* de una red y el *índice de efectividad pasajero/plano*.

Otra línea de investigación en el área de evaluación de redes de tránsito se relaciona con el concepto de robustez. La robustez en el diseño de redes se puede definir de varias formas. La mayoría de las referencias coinciden en definirla como la capacidad de una red para seguir operando en presencia de fallos en cualquier tramo o vehículo, o a posibles imprevistos externos. Se volverá a este concepto en el Capítulo 2.

El resto de esta memoria se estructura como sigue. El Capítulo 2 introduce de manera holgada y genérica los problemas de diseño de redes de tránsito rápido, desde la perspectiva más cualitativa (elaboración del proyecto, objetivos, diferentes modelos de localización), por un lado, hasta un modelo real cuyo objetivo es reflejar los efectos del amontonamiento de pasajeros y la sobreocupación en una red de metro. En el Capítulo 3 se introduce el modelo que da título a esta memoria. Aquí se formularán la función objetivo y las restricciones pertinentes. Se contemplarán dos casos: modelos sin sobreocupación, y modelos con sobreocupación. En el Capítulo 4 se estudiará el entorno tecnológico: repaso del lenguaje de programación escogido, las librerías empleadas, el solver³ seleccionado y la estructura del programa. Con todo ello, los resultados experimentales se presentan en el Capítulo 5, constituyendo un caso práctico basándonos en una red existente. A la vista de los resultados, se proponen algunos posibles horizontes de estudio y líneas de investigación futuras en el Capítulo 6. Finalmente, el Capítulo 7 realiza una retrospectiva de todo el texto, agrupando las principales conclusiones y exponiendo los resultados más relevantes.

³En este texto haremos uso del anglicismo «*solver*» para hacer referencia a las herramientas numéricas usadas para resolver los distintos problemas de optimización matemática planteados.

2. Diseño de redes de tránsito rápido.

Las redes de metro y otros medios de tránsito rápido aumentan notablemente la movilidad de la población dentro de las urbes, a la par que contribuyen a reducir la congestión del tráfico y la polución producida por un número masivo de vehículos. Dado el crecimiento poblacional al que en general se enfrentan todas las ciudades del mundo, sumado a la urbanización de nuevos terrenos y la expansión de los núcleos residenciales, numerosas áreas metropolitanas están optando por construir o extender sus redes de transporte.

Dentro de los distintos medios de desplazamiento, las redes de metro han mostrado ser más eficientes que otros sistemas de transporte como el autobús o los vehículos privados, debido a su menor consumo de energía y su capacidad para transportar una mayor densidad de pasajeros. A su vez, en la mayoría de los casos se opta por redes subterráneas, que no interfieren en las calzadas de la superficie y pasan desapercibidas en el día a día, causando un impacto mínimo tras su construcción y funcionando de manera independiente al resto de tráfico urbano. En numerosos textos, el concepto de *tránsito rápido* no sólo es referido a redes de metro. También se incluyen monorraíles y trenes ligeros, entre otros, teniendo todos ellos en común la capacidad de proporcionar un desplazamiento rápido y eficiente de pasajeros a lo largo de la geografía de las ciudades.

A la hora de planear la construcción de una red de tránsito rápido es necesario tener en cuenta numerosos agentes: políticos, urbanistas, agencias de movilidad, ingenieros, empresas de construcción, plataformas ciudadanas, etc. La conjugación de todos estos elementos da lugar a que las tareas de planificación resulten tremendamente complejas, a fin de satisfacer los requisitos y necesidades impuestos por cada parte. Habitualmente, la primera etapa consiste en un análisis exhaustivo de las áreas donde se pretende construir la nueva línea, considerando especialmente la densidad de población de las mismas y los patrones de desplazamientos exhibidos hasta entonces. Una vez almacenados estos patrones en matrices de flujo de origen-destino, se identifican y combinan algunos corredores de tráfico amplios, dando lugar a varios escenarios de red susceptibles de ser evaluados desde diferentes perspectivas (a menudo, haciendo uso de análisis de decisión multi-criterio). Se trata de un problema eminentemente estratégico, por lo que es habitual que los procesos se demoren en el tiempo.

En términos generales, la planificación de las redes de tránsito rápido se puede clasificar dependiendo de si la red se va a construir desde cero, o si por el contrario ésta va a crecer incorporando nuevas líneas o ampliando algunas de las existentes. Un excelente ejemplo lo encontramos en la Figura 2.1, donde se representa la red de metro de Moscú¹. En dicha figura, apreciamos a la izquierda un brazo adicional de la línea 4 (azul claro), de construcción más reciente que el resto de la línea, a modo de ampliación en un tramo intermedio. De igual manera, en la parte inferior vemos cómo la línea 6 (naranja) ha sido ampliada para unir su extremo con la red de trenes de cercanías *L1* (celeste). Desde un punto de vista

¹En los años 2016 y 2019 se incorporaron las nuevas líneas 14 y 15, respectivamente, así como la el comienzo de un segundo anillo exterior (ampliación de la línea 11) comenzado en 2018, cuya finalización está prevista para finales de 2022. Un mapa completo, actualizado e interactivo del metro de Moscú puede encontrarse en: <https://mosmetro.ru/metro-map/>

CAPÍTULO 2. DISEÑO DE REDES DE TRÁNSITO RÁPIDO.

estrictamente técnico, el proceso *secuencial* de planificación ferroviaria se basa en el conocimiento de los patrones de viaje y comienza con el diseño de la red. En contraposición, históricamente el resto de redes fueron construidas desde cero, manteniendo intacta su estructura hasta la actualidad. La consecuente planificación de líneas, la programación de horarios y la programación de recursos corresponden a las etapas posteriores de este proceso. Cuestiones adicionales a tener en cuenta son la fiabilidad de las líneas, su solidez, la capacidad para adaptarse a un horario predeterminado, la capacidad de maniobra ante imprevistos, el diseño de los andenes, etc.

Habitualmente, las vías en una red de metro no están interconectadas. No obstante, existen notables excepciones, como puede ser el caso de redes donde exista un tronco común para varias líneas (Los Ángeles o Bilbao, por ejemplo), o el caso de aquellas líneas que funcionan como un conjunto de líneas, funcionando la mayoría de ellas de manera independientes (Londres). Volviendo a la Figura 2.1, vemos cómo la línea 1 (rojo) actúa como eje central y referencia para las posteriores líneas. Por otra parte, llaman la atención las numerosas intersecciones entre líneas, destacando la intersección casi en el centro del mapa de las líneas 1, 3, 4 y 9 (rojo, azul, azul claro y gris, respectivamente). Destaca también la línea 5 (marrón), de diseño circular, cuya finalidad es interconectar periféricamente y en ambos sentidos todas las líneas de la ciudad, que de otra manera no siempre confluirían.

Cabe destacar que, en algunas ciudades, los sistemas de cercanías también comparten el sistema ferroviario, tanto en el centro de la ciudad como en zonas periféricas (*L1* en la Figura 2.1, y también las correspondencias con trenes y tranvías en numerosas zonas del círculo central —línea 5—). Esto implica que el diseño de la red y la planificación de la línea (excepto la configuración de la frecuencia) se consideran juntos en el proceso de modelado. Una segunda característica específica de los metros es que transportan un gran número de viajeros que recorren distancias cortas en comparación con los ferrocarriles de media y larga distancia, siendo 1 – 2 minutos el tiempo de viaje típico entre una estación de metro y la siguiente. Junto a lo anterior, conviene tener presente que el diseño de la red de tránsito rápido se compone de dos problemas entrelazados: la determinación de las alineaciones y la ubicación de las estaciones, teniendo en cuenta que pueden existir varios medios de transporte compitiendo entre sí en determinadas áreas urbanas.

En particular, determinar la localización de las estaciones es un problema típico de ubicación en redes de transporte para el cual se pueden aplicar diversos criterios, dependiendo de los objetivos de la administración a cargo del proyecto. Una estación ubicada en un área de alta densidad poblacional podría resultar poco eficiente debido a la dirección de la línea a la que pertenece: por ejemplo, si dicha la línea va de norte a sur, pero las personas ubicadas cerca de la estación trabajan al este de la misma, la estación tendrá escasa utilidad para los desplazamientos relacionados con el trabajo de los habitantes de dicha área. Por tanto, resulta fundamental analizar adecuadamente la orientación de las alineaciones de una red, y no sólo en centrarse en la ubicación de las estaciones. Desde un punto de vista matemático, dado que la instalación a ubicar es una red completa (y no una única estación), algo apreciablemente grande respecto a su entorno, el problema considerado constituye un problema de ubicación de instalación multidimensional (Laporte and Mesa [2015]).

2.1. Construcción de la red: proyecto, objetivo y evaluación.

Fijados estos precedentes, podemos afirmar cuál es, en última instancia, el principal objetivo de un sistema de transporte colectivo: **mejorar la movilidad de la población**. Dado que los sistemas de tránsito rápido suelen tener una gran capacidad, reducen en gran medida la congestión del tráfico, la contaminación del aire y el consumo de energía, proporcionando así una movilidad sostenible. Además, estos sistemas se encuentran entre los modos colectivos de transporte terrestre más rápidos y, por lo tanto, suelen proporcionar los tiempos de viaje más cortos. Otra importante característica es su influencia cartográfica en las ciudades, ya que vertebran el desarrollo de áreas residenciales, empresariales y comerciales. Los sistemas de tránsito rápido requieren inversiones donde se implican capitales muy elevados, tanto para la construcción como para el posterior mantenimiento. La inversión inicial está relacionada fundamentalmente con la construcción de túneles, los sistemas de comunicación, pago de tasas y la compra de material. Los agentes interesados en los procesos de planificación se pueden clasificar, a grandes rasgos, en tres grupos: la sociedad en general (representada por las agencias de transporte y los correspondientes delegados gubernamentales), los potenciales usuarios y la empresa que ofrece el servicio (que puede ser de carácter público o privado).

Una medida frecuentemente utilizada en la etapa de planificación es la población cubierta por el sistema, a menudo definida como la población que vive dentro de un cierto umbral (circular) de distancia de las estaciones. En las áreas menos pobladas, este umbral puede alcanzar hasta 1 km de radio, mientras que en ciudades grandes es habitual encontrar estaciones a pocos minutos a pie de cualquier lugar. Es preciso tener en cuenta que las distancias usadas bajo estos criterios no son euclídeas, por lo que sólo se trata de una aproximación. Una medida alternativa, más eficiente, es la cobertura de viaje pronosticada que se puede medir mediante encuestas de origen-destino, junto con modelos de equilibrio de tráfico. Los usuarios potenciales de la red están principalmente interesados en reducir su tiempo de viaje. A su vez, un objetivo secundario es el de hacer el menor número de transbordos posibles entre líneas. En lo que respecta a las empresas constructoras, son las principales implicadas en gestionar los costes e ingresos fijos y variables de construcción y explotación dentro del proyecto.

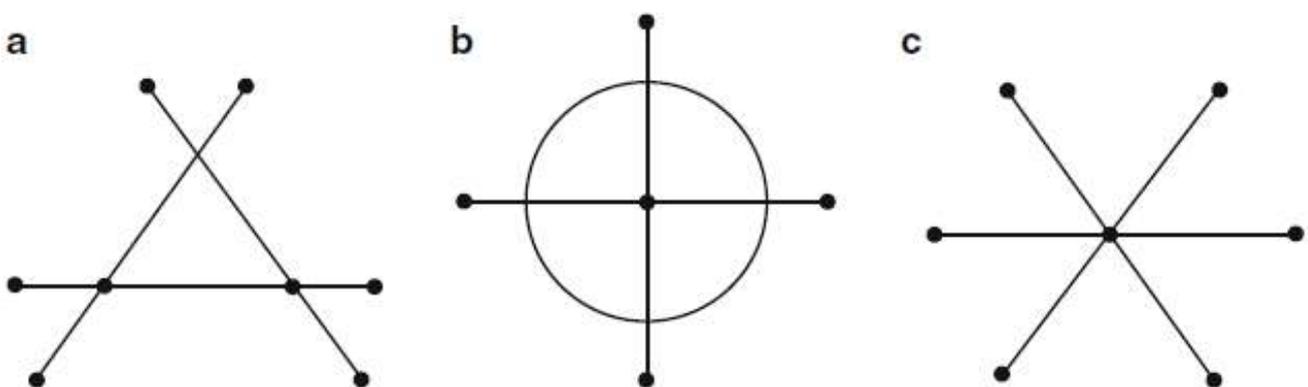


Figura 2.2.: Tres diseños básicos en redes de metro: (a) Red triangular. (b) Red en forma rueda de carro. (c) Red estrellada. Fuente: Laporte and Mesa [2015].

Una vez construida la red, ésta puede evaluarse mediante **medidas e indicadores** de red. No obstante las mismas medidas también pueden utilizarse para evaluar redes potenciales, en particular las resultantes del proceso de combinación de corredores. Para este fin, la teoría de grafos resulta

2.1. CONSTRUCCIÓN DE LA RED: PROYECTO, OBJETIVO Y EVALUACIÓN.

especialmente útil. Estas medidas, además, se pueden utilizar como funciones objetivo o como restricciones en modelos de programación matemática. Algunos autores (ver Laporte and Mesa [2015] y las referencias contenidas) han medido y estudiado la eficiencia de las redes de tránsito rápido a través de las medidas pasajeros/red y pasajeros/plano (Laporte et al. [1997]). Por ejemplo, se ha demostrado que en una ciudad circular, los diseños triangulares y de rueda de carro son preferibles a los diseños de tipo estrella (ver Figura 2.2), en términos de conectividad y sencillez de viaje.

En los últimos años, se han desarrollado nuevos indicadores basados en la evaluación y eficacia de las redes de transporte, enfocados desde la perspectiva de la teoría de grafos en redes complejas y la robustez de éstas. Podemos definir la robustez como el grado en que un sistema o componente puede funcionar correctamente en presencia de entradas no válidas o condiciones del entorno desfavorables. En el caso de la planificación de redes de tránsito rápido, el número de pasajeros futuros es una variable de entrada de incertidumbre que también dependerá de los tiempos de viaje de los modos de transporte alternativos. En este contexto, en las últimas décadas se ha incrementado el número de estudios relacionados con las propiedades estructurales de las redes, siendo las redes de metro el ejemplo más citado entre las redes de transporte. A menudo, las redes resultantes de estas investigaciones se hallan en un punto intermedio entre redes regulares (con nodos de igual grado) y redes aleatorias (generadas por borde por una probabilidad dada). Esto refuerza la necesidad de conocer qué resultados pueden ser válidos —más allá del aspecto puramente matemático— para su implementación en una zona geográfica.

Definamos un grafo $G = (V, E)$, y sea d_{ij} la distancia topológica entre los nodos $v_i, v_j \in V$ (i.e., el número mínimo de aristas en un camino entre v_i y v_j). Es habitual encontrar en algunos textos (Laporte and Mesa [2015]) definida la longitud del *camino característico* L y el *coeficiente de agrupamiento* C como:

$$L = \frac{1}{|V|(|V| - 1)} \sum_{i \neq j} d_{ij}, \quad (2.1)$$

y

$$C = \frac{1}{|V|} \sum_{v_i \in V} C_i, \quad (2.2)$$

donde C_i determina el número de aristas en $G_i = (V_i, E_i)$, el subgrafo de vecinos de v_i , dividido entre el máximo número posible $|V_i|(|V_i| - 1)/2$.

Las redes de mundo pequeño son tolerantes a las interrupciones: son robustas. Se ha demostrado que las redes de metro no son sólidas a nivel local. Sin embargo, las redes de conexiones directas, donde existe un límite físico entre todos los pares de estaciones para las cuales los pasajeros no necesitan transferirse a otra línea, pueden verse como redes de mundo pequeño.

2.2. Modelado algorítmico para redes de tránsito rápido.

Los proyectos de construcción de redes de transporte rápido se pueden agrupar en tres categorías: (i) Planificación desde cero de una línea sin la existencia previa de ninguna otra (por ejemplo, el metro de Granada en 2013). (ii) Planificación simultánea, desde el comienzo, de varias líneas (Sociedad del Metro de Sevilla, 2001). (iii) Aquellos en los que se pretende ampliar una red existente, lo que corresponde a un problema de diseño de red condicional (línea 4 del metro de Moscú, 2008).

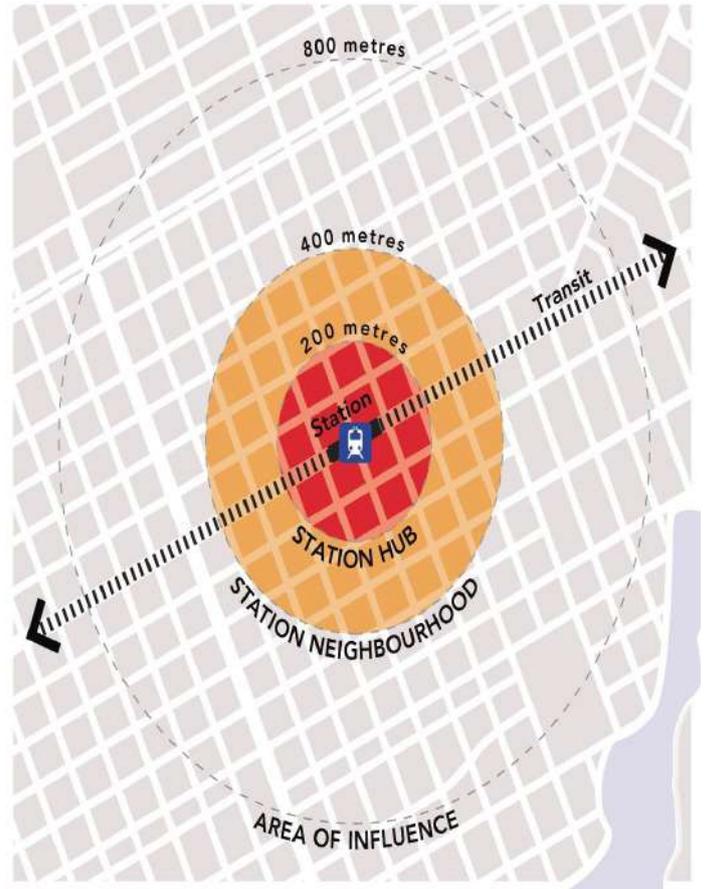
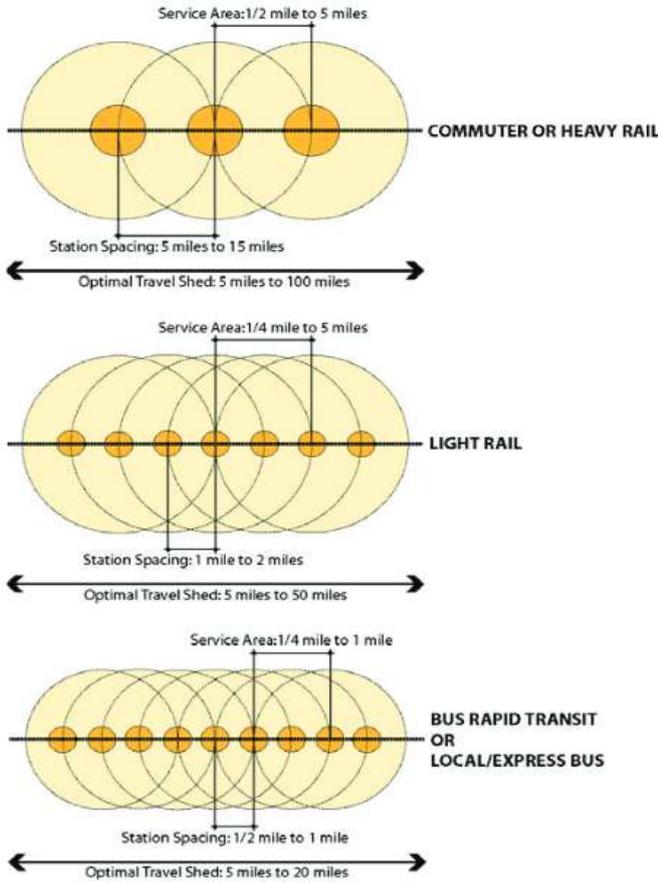
El reto de localizar una línea para un sistema de tránsito rápido recae dentro del área de ubicación de estructuras dimensionales, ya sea en un espacio discreto o continuo. Concretamente, se trata de un problema de ubicar caminos y redes. En el marco de la teoría de grafos, el problema consiste en seleccionar un camino entre dos nodos (que podrían ser fijos, a priori) y algunos de los nodos intermedios para que sean estaciones, con el fin de optimizar una función objetivo sujeta a ciertas restricciones. En una configuración continua, el problema consiste en seleccionar una línea recta, una línea fraccionada o un segmento curvo y algunos puntos en él. Si la línea de tránsito rápido va a situarse sobre la superficie, casi siempre será necesario trabajar con un entorno discreto. Sin embargo, si la red se va a construir bajo tierra, encajaría mejor un espacio mixto de red continua. Si se quiere modelar de manera realista el problema de ubicar una red de transporte, es preciso considerar varias características además de las que se encuentran en los problemas de cubrimiento de caminos, lo cual supone una extensión de los algoritmos descritos en la bibliografía, incluyendo restricciones de espacio entre estaciones, competencia con otros medios de transporte (o la combinación con éstos, para según qué destinos), demanda asignada a pares de puntos en lugar de puntos individuales, etc.

Supongamos que nuestro problema consiste en construir una red de tránsito rápido desde cero, así como el problema de extender una red ya ubicada. El primer paso consistirá en la selección de estaciones clave, respondiendo a la pregunta: ¿cuáles son los principales puntos de atracción? La respuesta a esta cuestión abarca un amplio espectro de casuísticas, desde áreas residenciales densamente pobladas, hasta centros comerciales, zonas turísticas, aeropuertos, hospitales, centros de negocios, etc. La ubicación de las estaciones se realizará (más allá de las posibles restricciones burocráticas) con la ayuda de herramientas analíticas. La etapa inmediatamente posterior consistirá en conectar las estaciones clave para formar una *red central*. Finalmente, las estaciones intermedias se ubicarán en el trazado resultante de la segunda etapa. El objetivo prioritario, desde el punto de vista de las administraciones implicadas, será generalmente atraer a la mayor cantidad de viajeros posible, minimizando el tiempo total de viaje de los pasajeros, a la par que se determina un espaciado óptimo entre estaciones, teniendo en cuenta los tiempos de acceso al servicio, la movilidad de los vehículos, los tiempos de permanencia y los tiempos de transferencia intermodal.

De manera estándar, este proceso puede realizarse usando modelos de programación matemática entera o entero-mixta, con el objetivo de maximizar la cobertura del viaje para resolver el problema de diseño de la red central en presencia de un modo de transporte alternativo. En lo que respecta a los estudios orientados a optimizar la ubicación de nuevas estaciones, existen modelos donde el objetivo es maximizar el ahorro en el tiempo de viaje de los pasajeros introduciendo nuevas estaciones en zonas clave. Otros modelos contemplan el doble criterio de maximizar la cobertura, minimizando a su vez el número de nuevas estaciones. También existen modelos que combinan el número de estaciones y las distancias a las mismas. Respecto a este primer caso, el objetivo consiste en minimizar el número de nuevas estaciones asumiendo que la demanda está cubierta dentro de una distancia predefinida. La segunda casuística constituye un problema NP-duro, y consiste en minimizar la suma de las distancias desde los puntos de demanda hasta la estación más cercana bajo la restricción de que el número de

2.2. MODELADO ALGORÍTMICO PARA REDES DE TRÁNSITO RÁPIDO.

estaciones nuevas esté acotado superiormente. Un interesante y exhaustivo análisis puede encontrarse en De los Santos [2014]. En este capítulo abordaremos superficialmente, y a modo introductorio para capítulos posteriores, algunos modelos para la localización de redes y la localización de líneas, quedando patente al final la necesidad de esta distinción. Los modelos propuestos a continuación analizan las ideas propuestas en Laporte et al. [2011b].



(a) Solapamiento de áreas de influencia para nodos en distintos medios de transporte. (b) Área de influencia alrededor de una estación.

Figura 2.3.: Esquema de radios de influencia para distintos tipos de trenes. Fuente: Renaissance Planning Group [2011].

2.2.1. Modelos de localización de redes.

Formalmente, el problema de diseño de una red de metro consiste en incorporar un conjunto de líneas de tránsito interconectadas dentro de una red no dirigida más grande, $G = (N, E)$, donde $N = \{1, \dots, n\}$ denota el conjunto de nodos y $E = \{(i, j) : i, j \in N, i < j\}$ denota el conjunto de aristas. Los nodos corresponden a centroides de población dentro de una zona urbana, mientras que las aristas corresponden a conexiones potenciales a construir entre pares de vértices. Sea p_i la población asociada al nodo i : la población del vértice normalmente vendrá definida como la población que vive dentro de una distancia razonable desde el propio vértice. Definiendo como c_{ij} el coste de construir una arista (unión) entre i y j , y como c_i el coste de construir una estación en el nodo i , los modelos matemáticos empleados para el diseño de redes de metro hacen uso de tres criterios principales:

1. Coste total de construcción.
2. Población total cubierta por la red.
3. Tráfico total de (O/D) capturado por la red.

Este problema puede plantearse desde el punto de vista de la optimización multicriterio. Alternativamente, es posible minimizar el coste sujeto a una restricción de cobertura de población o tráfico, o bien maximizar la cobertura ligada a una restricción presupuestaria. Dada una red de tránsito, un problema de cobertura de conjuntos puede resolverse más adelante para ubicar las líneas, pero es probable que haya varias superposiciones entre ellas.

Profundizando en la teoría de grafos, los modelos de localización de redes con coste mínimo o cuyo objetivo sea obtener una cobertura máxima se engloban dentro de los llamados problemas de *árbol de Steiner*² con ganancias (STPP, por sus siglas en inglés). Consideraremos aquí tres ejemplos típicos:

1. Los algoritmos «STPP de recolección de premios» (prize-collection STPP, PCSTPP) abordan problemas de optimización combinatoria, donde la función objetivo es una combinación lineal del coste de construcción y de la población cubierta por la red. Sea x_{ij} una variable binaria igual a 1, si y sólo si la arista $\{i, j\}$ pertenece a la red. Y sea y_i una variable binaria igual a 1 si y solamente si el vértice i pertenece a la red. A su vez, sea α un parámetro positivo definido por el usuario. Por último, se define $T \subseteq N$ como un conjunto de nodos que necesariamente deben pertenecer a la red (T puede ser vacío). El modelo será entonces:

Minimizar:

$$\sum_{(i,j) \in E} c_{ij}x_{ij} + \sum_{i \in N} c_i y_i - \alpha \sum_{i \in N} p_i y_i \quad (2.3)$$

s.a. las restricciones:

$$\sum_{(i,j) \in E} x_{ij} = \sum_{i \in N} y_i - 1 \quad (2.4)$$

$$\sum_{i,j \in S} x_{ij} \leq \sum_{i \in S/\{k\}} y_i \quad k \in S \subseteq N, |S| \geq 2 \quad (2.5)$$

$$y_i = 1 \quad i \in T \quad (2.6)$$

$$x_{ij} \in \{0, 1\} \quad (i, j) \in E \quad (2.7)$$

²Cuando se hace alusión a los problemas de árbol de Steiner, generalmente se abarca una clase de problemas en grafos que requieren conectar nodos específicos de un grafo (denominados *nodos terminales*), con un coste mínimo, y añadiendo algunos nodos extra. Interconectar nodos generalmente requiere seleccionar aristas de un grafo dado que forman un *árbol*, denominando a la solución árbol de Steiner.

2.2. MODELADO ALGORÍTMICO PARA REDES DE TRÁNSITO RÁPIDO.

$$y_i \in \{0, 1\} \quad i \in N. \quad (2.8)$$

La función objetivo tiene como fin minimizar el coste de construcción de la red, menos un coste proporcional a la población cubierta por la red. Las restricciones (2.4) y (2.5) fuerzan a la red a ser ramificada. La primera de ellas impone que el número de aristas seleccionadas sea igual al número de aristas requeridas para generar un árbol de expansión a raíz del subgrafo implicado, i.e., $y_i - 1$, mientras que la segunda garantiza que la solución no contenga ciclos (véase Lucena and Resende [2004]).

2. Por su parte, en los algoritmos STPP «de cuota» (Costa et al. [2005]) el objetivo es minimizar el coste mientras se asegura una cobertura mínima de población, β :

Minimizar:

$$\sum_{(i,j) \in E} c_{ij}x_{ij} + \sum_{i \in N} c_i y_i \quad (2.9)$$

s.a. las restricciones (2.4)-(2.8), y:

$$\sum_{i \in N} p_i y_i \geq \beta, \quad (2.10)$$

que indica que si el coste total recaudado excede la cobertura mínima de población, entonces es posible encontrar un valor de β que determine una cota inferior para dicho coste.

3. Una tercera variante la encontramos en los STPP fraccionarios, donde el objetivo es maximizar la relación población-coste. El modelo es:

Maximizar:

$$\alpha \sum_{i \in N} p_i y_i / \left(\sum_{(i,j) \in E} c_{ij}x_{ij} + \sum_{i \in N} c_i y_i \right) \quad (2.11)$$

s.a. las restricciones (2.4)-(2.8).

La función (2.11) es no lineal, lo cual va a aumentar la complejidad de la resolución del modelo considerado. Adicionalmente, se pueden desarrollar otra clase de modelos considerando explícitamente la población no cubierta por la red. Dichos modelos hacen uso de variables z_{ij} iguales a 1, si y sólo si al nodo i que no forma parte de la red se asigna el nodo j más cercano en la red. Definiendo como d_{ij}

el coste de llegar a j desde i , es posible modificar cualquiera de los modelos anteriores añadiendo un término $\gamma \sum_{(i,j) \in E} d_{ij} z_{ij}$ a la función objetivo, siendo γ un parámetro definido por el usuario positivo, o bien imponiendo un límite superior al sumatorio. En este tipo de modelos, es especialmente necesario definir cuidadosamente las restricciones para garantizar que un nodo no pueda estar al mismo tiempo en la red y fuera de ella.

2.2.2. Modelos de localización de líneas.

Más allá de la localización de redes, se distingue entre los modelos anteriores y modelos diseñados para localizar un conjunto L de líneas que cubran sólo una parte de la red, $G(N, E)$. De forma similar al primer STPP expuesto, la función a minimizar consistirá en una combinación lineal del coste de construcción y de la población cubierta por el tránsito de la red, para lo cual se resta del coste de construcción la población cubierta por la red, multiplicada por un parámetro α . Estos modelos combinan la formulación descrita para los PCSTPP: en primer lugar, se definen las siguientes variables binarias: $x_{ij}^l = 1$, si y sólo si la arista $\{i, j\}$ pertenece a la línea $l \in L$, $x_{ij} = 1^3$ si y solamente si la arista $\{i, j\} \in E$ pertenece a una línea, y $y_i^l = 1$ si y sólo si se construye una estación en el nodo i en la línea l . A su vez, sea z_k^l el número de aristas incidentes en un nodo $k \in N$ perteneciente a la línea $l \in L$, excluyendo las aristas de la línea l . Los costes de construcción c_{ij} tienen el mismo significado que los algoritmos anteriores, y c_i^l define el coste de construir una estación en el nodo i en la línea l . El modelo localización de líneas «recolección de premios» (PCLL) será, por tanto:

Minimizar:

$$\sum_{(i,j) \in E} c_{ij} x_{ij} + \sum_{l \in L} \sum_{i \in N} c_i^l y_i^l - \alpha \sum_{i \in N} p_i y_i \quad (2.12)$$

s.a. las restricciones:

$$x_{ij}^l \leq y_i^l \quad (i, j) \in E, l \in L \quad (2.13)$$

$$x_{ij}^l \leq y_j^l \quad (i, j) \in E, l \in L \quad (2.14)$$

$$x_{ij}^l \leq x_{ij} \quad (i, j) \in E, l \in L \quad (2.15)$$

$$\sum_{i < k} x_{ik}^l + \sum_{j > k} x_{kj}^l \leq 2 \quad k \in N, l \in L \quad (2.16)$$

$$\sum_{i,j \in S} x_{ij}^l \leq |S| - 1 \quad S \subset N, |S| \geq 2, l \in L \quad (2.17)$$

$$\sum_{\substack{i,j \in N \\ i < j}} x_{ij}^l \geq \sum_{i \in N} y_i^l - 1 \quad l \in L \quad (2.18)$$

³Haremos aquí un pequeño abuso de notación, con el fin de respetar el desarrollo matemático de Laporte et al. [2011b].

2.2. MODELADO ALGORÍTMICO PARA REDES DE TRÁNSITO RÁPIDO.

$$z_k^l \geq \sum_{h \in L/\{l\}} \left(\sum_{i < k} x_{ik}^h + \sum_{j > k} x_{kj}^h \right) - M(1 - y_k^l) \quad k \in N, l \in L \quad (2.19)$$

$$z_k^l \leq \sum_{h \in L/\{l\}} \left(\sum_{i < k} x_{ik}^h + \sum_{j > k} x_{kj}^h \right) + M(1 - y_k^l) \quad k \in N, l \in L \quad (2.20)$$

$$z_k^l \leq M y_k^l \quad k \in N, l \in L \quad (2.21)$$

$$\sum_{k \in N} z_k^l \geq 1 \quad l \in L \quad (2.22)$$

$$x_{ij}^l, x_{ij}, y_i^l = 0 \text{ ó } 1 \quad (i, j) \in E, i \in N, l \in L \quad (2.23)$$

$$z_k^l \geq 0 \quad k \in N, l \in L. \quad (2.24)$$

En la función objetivo (2.12), el parámetro α controla el porcentaje de población cubierta por la red. Un valor $\alpha = 0$ implicaría no construir ninguna red. Por el contrario, valores muy elevados de α resultarán óptimos para cubrir a gran parte de la población. El coste de construir una estación en el nodo i es calculado para cada línea, pues el coste de las estaciones de transferencia (es decir, donde los pasajeros cambian de una línea a otra) depende del número de líneas conectadas a través de ellas. Las restricciones (2.13) y (2.14) imponen que la arista (i, j) no pueda ser parte de la red si no se construye ninguna estación en i ni en j . Por su parte, la restricción (2.15) implica que la línea l puede usar la arista (i, j) sólo en el caso de que $x_{ij} = 1$ y, por lo tanto, se incurre en el coste de construcción correspondiente. La restricción (2.16) obliga a que el grado de cada nodo sea 0, 1 o 2, mientras que las restricciones (2.16) y (2.17) —nótese que hay un número combinatorio de estas restricciones— indican que todas las líneas constituirán un conjunto de caminos. Por otro lado, las restricciones (2.16)-(2.18) aseguran que cada línea se compone de una única ruta. La formación de líneas no conectadas a las demás se impide mediante las restricciones (2.19)-(2.22), siendo M un número suficientemente grande en (2.19) y (2.20). Las restricciones (2.19)-(2.21) son las responsables de definir z_k^l . La imposición de que cada línea deba cruzarse con al menos otra línea surge de (2.22). Finalmente, las restricciones (2.23) y (2.24) imponen condiciones a las variables. Este modelo evita la construcción de líneas circulares como la que se aprecia en la Figura 2.1, lo cual requeriría relajar algunas restricciones. Sin embargo, se permite que varias líneas compartan la misma arista, con la ventaja de que no sea necesario construir todas las líneas proyectadas.

El modelo descrito no optimiza la direccionalidad del tráfico entre los pares O/D , y garantiza únicamente que la red esté conectada. No obstante, ilustra la necesidad de modelar adecuadamente la función objetivo y las restricciones según los objetivos prioritarios de un proyecto de diseño y localización de líneas de tránsito. Modelos alternativos consistirían en minimizar el tiempo total de viaje (tiempo de viaje + tiempo de transferencia). El tamaño del modelo crece muy rápidamente con el número de pares O/D . Una evolución natural del problema descrito son los modelos propuestos para el diseño de redes de tránsito rápido, donde se minimiza el efecto de la interrupción en la cobertura total del viaje. El modelo propuesto en este trabajo se introducirá en el Capítulo 3, ilustrando algunos de los resultados en el Capítulo 5.

2.3. Modelos de amontonamiento en redes de tránsito rápido

En las áreas metropolitanas densamente pobladas, los estudios empíricos han encontrado que el amontonamiento dentro de los vehículos de tránsito ha empeorado cada vez más en los últimos años. El amontonamiento crónico dentro de los vehículos no sólo es causado por la falta de infraestructura física, sino también por la provisión inadecuada de servicios. Dada la prevalencia de los vehículos de transporte superpoblados, realizaremos en este trabajo un pequeño estudio tanto cualitativo como cuantitativo, centrándonos especialmente en la remodelación de las funciones de coste de amontonamiento en los vehículos para trenes ligeros. En la referencia (Qin [2014]), uno de los textos clave que motivaron este trabajo, Feifei Qin propone tres estudios de casos numéricos que muestran que la aplicación de distintas funciones de costes de aglomeración en los vehículos a diferentes modos de tránsito tienen implicaciones no sólo para la estructura de costes de los sistemas de tránsito y la magnitud de las prestaciones óptimas del servicio, sino también para la presencia de economías de escala en el consumo.

Los residentes de áreas urbanas densamente pobladas inevitablemente enfrentan dificultades al viajar. Durante las horas punta, el crecimiento sustancial de vehículos satura la capacidad de las carreteras, provocando una grave congestión del tráfico. Se trata de un fenómeno simultáneo a las dificultades de obtención de material rodante adicional. Ambos factores hacen que las instalaciones de transporte público a menudo no puedan hacer frente al creciente número de pasajeros. Por lo tanto, en las horas punta los pasajeros del transporte público a menudo se ven obligados a experimentar situaciones de incomodidad y estrés asociadas a tener que pararse dentro de vehículos extremadamente abarrotados.

En el campo de la economía del transporte, se han llevado a cabo un gran número de investigaciones teóricas y estudios prácticos sobre la congestión vial. Sin embargo, el efecto del amontonamiento en los vehículos sobre las operaciones de tránsito eficientes no se ha considerado detenidamente, a pesar de que *algunas* investigaciones exploratorias han tratado de abordar este problema desde la década de 1970. En la mencionada referencia, se destaca la falta de estudios que brinden enfoques efectivos para lidiar con el efecto del amontonamiento dentro de los vehículo y se destaca la brecha existente entre la investigación teórica y la implementación práctica del problema. Tres aspectos constituyen las principales razones por las que la investigación sobre el amontonamiento en vehículos aún está sin abordar. En primer lugar, dada la prevalencia de las aglomeraciones en el tránsito urbano, a día de hoy no existe una definición definitiva para describir los problemas de amontonamiento en vehículos. En segundo lugar, debido a la diversidad de modos de transporte, la investigación existente no ofrece formas legibles de medir y ponderar el nivel de aglomeración en los vehículos, generando a su vez más confusión en términos de evaluación y modelado. Finalmente, también existen muchas controversias sobre cómo cuantificar y monetizar los costes de amontonamiento e incorporarlos en la programación y fijación de precios del tránsito.

Así pues, en esta sección desarrollaremos un pequeño marco conceptual en esta materia para definir, medir y evaluar con precisión el amontonamientos en los vehículos. A su vez, se definirán tres funciones no lineales para el coste de aglomeración, cuyo objetivo es capturar la relación entre el valor promedio del tiempo de conducción y los niveles de aglomeración.

2.3.1. Contextualización.

Dada la insatisfacción generalizada con las aglomeraciones y el amontonamiento por parte de los pasajeros, se ha realizado una cantidad considerable de estudios para especificar los efectos sociales del amontonamiento. Desafortunadamente, en la mayoría de los estudios el efecto de amontonamiento en los vehículos es secundario al propósito central de la investigación. Tan solo unos pocos estudios científicos se han centrado en abordar este problema en los vehículos para el transporte público.

El amontonamiento objetivo se mide cuantitativamente por el número de personas por unidad de espacio, mientras que el amontonamiento subjetivo se refiere a un estado de ánimo personal percibido que puede ocurrir cuando existe una gran disparidad entre la distancia interpersonal esperada y la distancia real. Así, un aspecto central del amontonamiento subjetivo es la “falta sentida de libertad de comportamiento y privacidad” cuando el espacio físico se vuelve demasiado limitado. Trasladado este concepto al campo de los estudios de transporte, el “amontonamiento” equivale a las experiencias desagradables de demasiados pasajeros que se acomodan en un espacio confinado, lo que empeora su bienestar. En consecuencia, se han especificado cuatro efectos de amontonamiento clave en la literatura:

- Amontonamiento en los vehículos.
- Amontonamiento en los andenes.
- Incremento de tiempos de espera.
- Incremento de tiempos de parada.

En las simulaciones realizadas en este trabajo sólo consideraremos el efecto del amontonamiento en los vehículos. La función que definiremos para ello en la presente sección, y que retomaremos en el Capítulo 3, representará la sobreocupación en nuestro modelo. Se utilizará para definir la carga de los arcos. Una posible clasificación de los distintos grados de amontonamiento se presenta en el Cuadro 2.1.

Nivel	Espacio de pasajero de pie	Perspectiva del pasajero
I	$> 1,0m^2/\text{pasajero}$	Los pasajeros pueden dispersarse y estirarse.
II	$0,5 - 1,0m^2/\text{pasajero}$	Viaje cómodo de pie con espacio suficiente entre pasajeros.
III	$0,40 - 0,49m^2/\text{pasajero}$	Viaje de pie sin contacto corporal.
IV	$0,30 - 0,39m^2/\text{pasajero}$	Contacto corporal ocasional.
V	$0,20 - 0,29m^2/\text{pasajero}$	Cercanía incómoda con otros pasajeros.
VI	$< 0,2m^2/\text{pasajero}$	Presión o aplastamiento debido al resto de pasajeros.

Cuadro 2.1.: Niveles de amontonamiento. (Fuente: Qu et al. [2020])

Dado el diseño interior con diferentes cantidades de asientos y espacio para estar de pie, los pasajeros tienen diferentes expectativas de conseguir asientos o viajar de pie. Por ejemplo, debido a que los trenes interurbanos están diseñados simplemente para contener asientos, los pasajeros creen que comprar un billete les da derecho a obtener un asiento. Una vez que se ven obligados a ponerse de pie, sienten un gran estrés e incomodidad. En este caso, el amontonamiento ocurre cuando los pasajeros no pueden obtener asientos como inicialmente esperaban. Por el contrario, los

modos ferroviarios urbanos y algunos autobuses están diseñados para transportar una gran cantidad de personas de pie en lugar de proporcionar tantos asientos como sea posible. En consecuencia, los pasajeros no perciben el hecho de estar de pie como estar amontonados. Conforme suben al vehículo más pasajeros, éstos comienzan a percibir amontonamiento cuando la densidad de personas de pie supera cierto umbral (diferente para cada persona), lo que lleva a una “invasión de pasajeros”.

El amontonamiento en un vehículo es difícil de medir, y por ello las investigaciones empíricas hacen uso de descripciones cualitativas o medidas cuantitativas para medir su nivel. Esto incluye, por ejemplo, modelar la competencia que ocurre en una ruta ferroviaria describiendo los niveles de aglomeración en función de los asientos ocupados y la facilidad para encontrar éstos. A menudo, se trata de descripciones poco precisas y difícilmente cuantificables, lo que limita su uso en investigación. Por tanto, con el fin de especificar los niveles de aglomeración con la mayor precisión posible, se considerarán dos métricas cuantitativas: el factor de carga (denotado por LF , *Load Factor*, en algunos textos), y la densidad de pasajeros de pie (pasos/ m^2). El uso del factor de carga presenta ciertas limitaciones, como es el caso de los modos de tránsito de capacidad masiva, ya que una capacidad permanente alta puede dar lugar a factores de carga superiores al 100%. Es por ello que otros estudios optan por la segunda métrica (la densidad de pasajeros de pie) como una alternativa adecuada.

Ya en la década de 1970, algunos estudios señalaron que el amontonamiento afecta sustancialmente al valor monetario que los pasajeros asignan al ahorro de tiempo de viaje. A medida que aumenta su nivel, aumenta la disutilidad de viajar, por lo que los pasajeros tienen que asignar valores significativamente altos de tiempo de viaje para garantizar su bienestar. En este sentido, el amontonamiento en los vehículos puede evaluarse mediante índices de penalización, que pueden tener en cuenta factores como el valor monetario por unidad de tiempo y el valor monetario por viaje, o el incremento de tiempo respecto a un viaje sin sobreocupación. Debido a que el factor multiplicador de tiempo es inherentemente más transferible entre diferentes contextos en comparación con las valoraciones monetarias, se ha convertido en el enfoque predominante.

2.3.2. Paradigma matemático.

Los estudios empíricos actuales sugieren que las penalizaciones por amontonamiento crecen monótonamente, según aumente el nivel de éste, independientemente del tipo de indicador empleado. Sin embargo, la forma más apropiada para una función que describa la relación matemática entre las penalizaciones por aglomeración y sus niveles aún no se ha investigado en profundidad. Además, dado que la percepción del amontonamiento en los vehículos varía de un modo de tránsito a otro, no es posible usar una función de coste de amontonamiento común para todos los casos. En este contexto, resultan interesantes los estudios de Marvin Kraus (Kraus [1991]).

En su investigación en torno a las funciones de coste de amontonamiento existentes en el interior de los vehículos, el trabajo de Kraus dividió el valor del tiempo de conducción en dos partes: el valor del tiempo para pasajeros que aseguran asientos (P_{v0}) y el valor del tiempo en situaciones de amontonamiento donde los pasajeros tienen que ponerse de pie ($P_{v0} + \pi$). La forma genérica del modelo se puede formular como:

$$P_v = \begin{cases} P_{v0} & \text{Sin gente de pie.} \\ P_{v0} + \pi & \text{Algunas personas de pie.} \end{cases} \quad (2.25)$$

En la igualdad 2.25, π es la parte extra percibida por los viajeros. Dado que esta función permite expresar de forma sencilla el amontonamiento exterior, numerosos estudios posteriores la han empleado para sus investigaciones. Jara-Díaz y Gschwender (citados en Qin [2014]) actualizaron esta expresión, asumiendo que los valores del tiempo de estancia en los vehículos dejan de ser constantes y se convierten en una función continua decreciente:

$$P_v = P_{v0}(1 + \rho\phi), \quad (2.26)$$

donde, a modo de margen sobre el valor del tiempo de viaje (P_{v0}), la penalización por aglomeración (ρ) varía linealmente con la tasa de ocupación, (ϕ). En contraste con esta función lineal ampliamente utilizada, varios estudios han desarrollado funciones no lineales como piedra angular de su investigación, motivados por numerosas evidencias empíricas y por el hecho de que las funciones lineales asumen que el efecto del amontonamiento en el interior de los vehículos se modeliza de forma suave. En este trabajo estudiaremos tres de ellas, e implantaremos una de estas funciones en el modelo propuesto en el Capítulo 3 para reflejar los efectos de la sobreocupación.

Funciones no lineales de coste de amontonamiento para diferentes modos de tránsito.

Algunos autores no sólo han proporcionado una estimación detallada de las penalizaciones por aglomeración, sino que también han examinado posibles formas de la función, ya sea lineal, exponencial o de tipo Gompertz (sigmoidea). Las pruebas de bondad de ajuste para estas formas alternativas parecen avalar su elección respecto al caso lineal.

- **Función no lineal de costes de amontonamiento de un paso.**

Para algunos modos, como autobuses y trenes interurbanos, cuando el número de pasajeros dentro del vehículo es inferior a la capacidad de asientos la penalización por aglomeración no resulta relevante. Por tanto, el valor del tiempo de viaje permanece constante. Sin embargo, cuando viajan pasajeros de pie, la incomodidad y el estrés resultantes para éstos se ven reflejadas en un incremento de los tiempos de viaje. Para modelar esto, se puede derivar una función no lineal de un paso (al aproximarla, cada iteración dependerá únicamente del paso anterior) de la siguiente manera:

$$P_v = P_{v0} \frac{k}{N} + \left(1 - \frac{k}{N}\right) P_{v0} \left(1 + \beta e^{\alpha(\theta-1)}\right). \quad (2.27)$$

La ecuación (2.27) es resoluble analíticamente tras un proceso de linealización, y expresa el valor ponderado del tiempo de viaje de forma probabilística. Si denotamos por k la capacidad de asientos y por N el número de pasajeros a bordo, la relación de k/N muestra la probabilidad de conseguir un asiento. Para aclarar la distinción entre estar sentado y de pie, se asigna un valor constante de tiempo de conducción (P_{v0}) a los pasajeros sentados, y asumiendo que las penalizaciones por aglomeración para los pasajeros de pie aumentan exponencialmente con el factor de carga θ (o sea, N/k). En la

parte exponencial, se puede asignar arbitrariamente un cierto valor de α (> 5) como factor de ajuste. Por el contrario, el parámetro β debe calibrarse cuidadosamente a partir de la estimación empírica de las penalizaciones por aglomeración. Para facilitar la exploración, (2.27) se puede reescribir como:

$$P_v = P_{v0} + \left(1 - \frac{k}{N}\right) P_{v0} \beta e^{\alpha(\theta-1)}. \quad (2.28)$$

Si el número de pasajeros es menor que el número de asientos ($N < k$), el factor $(\theta - 1)$ tendrá signo negativo, al ser $\theta < 1$. Mediante el parámetro de escala α , la parte exponencial $\beta e^{\alpha(\theta-1)}$ se tenderá a cero, lo que hará que el valor medio del tiempo de viaje sea P_{v0} . Si el número de pasajeros es igual a la capacidad de asientos ($N = k$), el elemento $(1 - k/N)$ será cero, lo que indica que el valor medio resultante del tiempo de conducción es P_{v0} . A medida que aumenta el número de pasajeros que tienen que estar de pie en el vehículo, aparecerá un margen variable $(\beta e^{\alpha(\theta-1)})$, a modo de penalización por amontonamiento. Ante una sobreocupación inusualmente grande, el valor del tiempo de viaje tenderá al infinito.

• **Función no lineal de dos pasos para costes de amontonamiento.**

A diferencia de los modos interurbanos, la mayoría de los modos de tránsito en las ciudades basados en raíles (como el tranvía y los trenes ligeros) están diseñados para transportar un gran número de personas de pie. Para esos modos, el amontonamiento en los vehículos sólo tiene lugar cuando se supera cierto umbral. Aunque hasta ahora ninguna investigación específica se ha centrado en explorar este umbral, numerosas investigaciones asumen que el amontonamiento en los vehículos tiene lugar cuando el factor de carga alcanza excede el 140%, o la densidad de hacinamiento está por encima de los cuatro pasos/ m^2 . Distinguir entre pasajeros que viajan de pie o sentados no basta para describir correctamente el impacto del amontonamiento en el ahorro de tiempo de viaje de los pasajeros. Por tanto, es preciso modelar una nueva función que permita diferenciar entre la utilidad del viaje estando sentado, viajando de pie aún cuando haya pocos pasajeros, y de pie en vehículos llenos de gente:

$$P_v = P_{v0} + \frac{\rho P_{v0}}{1 + e^{\alpha(1-\theta)}} + \beta P_{v0} e^{\gamma(\theta-\delta)}. \quad (2.29)$$

De manera similar a la propuesta anterior, el primer término del lado derecho de (2.29), (P_{v0}) , corresponde al valor del tiempo de viaje en el caso base donde todos los pasajeros pueden encontrar asientos. Los parámetros α y γ pueden recibir valores arbitrarios, ya que no afectan fundamentalmente al valor del tiempo de conducción. Sin embargo, los parámetros β y ρ deben calibrarse en función de las penalizaciones por aglomeración bajo diferentes funciones lineales.

Para esta función, si el factor de carga es menor que uno ($\theta < 1$), el denominador del segundo término $(1 + e^{\alpha(1-\theta)})$ tenderá a infinito conforme α tienda a infinito, haciendo a su vez que el segundo término tienda a cero. Paralelamente, dado que el factor de carga es menor que el umbral ($\theta < \delta$), el tercer término $(\beta P_{v0} e^{\gamma(\theta-\delta)})$ tenderá a ser despreciable cuando el exponente tienda a cero. En consecuencia, el valor medio del tiempo de conducción mantendrá el valor constante $-P_{v0}$. A medida que suban más pasajeros, la capacidad limitada de asientos implicará que sólo una pequeña

proporción de ellos pueda encontrar asientos. El umbral (δ) permite que algunos pasajeros viajen de pie en condiciones de viaje con poca gente. Análogamente, el parámetro de escala α hace que el denominador del segundo término $(1 + e^{\alpha(1-\theta)})$ no se diferencie apreciablemente de 1. Al mismo tiempo, el parámetro γ reduce el tercer término $(\beta P_{v0} e^{\gamma(\theta-\delta)})$ a cero. Por lo tanto, en presencia de pocos pasajeros, el impacto de estar de pie en el valor promedio del tiempo puede reflejarse multiplicando una constante por el valor del tiempo en el caso base, que es $(1 + \rho)P_{v0}$. Si el factor de carga supera el umbral ($\theta > \delta$), el vehículo se abarrotará más y los pasajeros sufrirán la aglomeración. En este caso, el tercer término $(\beta P_{v0} e^{\gamma(\theta-\delta)})$ ya no se puede ignorar, y crece exponencialmente a medida que cada pasajero adicional suba al vehículo.

• **Función escalonada para costes de amontonamiento.**

A diferencia de los modos para autobuses, las pautas de diseño de capacidad física para trenes pesados generalmente adoptan la *densidad permanente* para especificar los niveles de aglomeración, en lugar de usar el factor de carga. Las condiciones de viaje se describen en términos del espacio de pie disponible y la capacidad de movimiento de los pasajeros. En este caso, el valor promedio del tiempo de viaje se puede modelar como una función escalonada. Nosotros sólo haremos distinción del valor del tiempo para seis niveles de densidad permanente (es decir, desde 0 pax/m² hasta 6 pax/m²). La función escalonada específica para un tren pesado se puede formular como:

$$P_v = P_{v0} \left(1 + \frac{\rho_1}{1 + e^{\gamma(\lambda_0 - N)}} + \frac{\rho_2}{1 + e^{\gamma(\lambda_1 - N)}} + \frac{\rho_3}{1 + e^{\gamma(\lambda_2 - N)}} + \frac{\rho_4}{1 + e^{\gamma(\lambda_3 - N)}} + \frac{\rho_5}{1 + e^{\gamma(\lambda_4 - N)}} + \frac{\rho_6}{1 + e^{\gamma(\lambda_5 - N)}} \right), \quad (2.30)$$

donde λ_0 especifica la capacidad de asientos. Una nueva serie de parámetros $\lambda_i (i = 1, 2, \dots, 5)$, siendo $\lambda_0 < \lambda_1 < \dots < \lambda_5$, denota el número de pasajeros para el i -ésimo nivel de densidad permanente. El valor de la penalización por aglomeración $\rho_i (i = 1, 2, \dots, 6)$ es el cambio incremental entre dos niveles consecutivos. A través del parámetro γ , el valor medio del tiempo se mantiene constante en cada nivel, saltando discretamente de uno a otro.

Minimización total del coste del sistema.

En la optimización de las operaciones de tránsito, uno de los enfoques adoptados es la minimización del coste total del sistema con respecto a los elementos operativos, como las tarifas, la frecuencia, el tamaño de los vehículos y las rutas. Dado que se asume que la capacidad de los vehículos y la densidad de rutas son factores externos, la única variable de control considerada es la frecuencia.

El coste del tiempo de espera constituye una parte apreciable de los costes totales del sistema. Dado que los viajeros pueden cambiar su comportamiento según el tipo de servicio, dividimos los servicios de tránsito en dos tipos: servicios basados en frecuencia y servicios basados en horarios. Para los primeros, los pasajeros llegan a las estaciones al azar. En el segundo caso, los pasajeros consultan los horarios antes de su salida. En esta sección asumiremos que cuando la frecuencia es mayor a 12 vehículos/hora, los servicios pueden considerarse basados en la frecuencia. Cuando la

CAPÍTULO 2. DISEÑO DE REDES DE TRÁNSITO RÁPIDO.

frecuencia es inferior a esta cantidad, el comportamiento de espera de los pasajeros comprende dos partes: espera pasiva en otros lugares y tiempo de espera activa en las estaciones. La fórmula general del coste promedio del tiempo de espera, (w) , para dos tipos de servicio es:

$$w = P_w \left(t_w + \frac{2\mu}{f} \right), \quad (2.31)$$

donde t_w es el tiempo fijo de seguridad que los pasajeros pasan esperando en las estaciones. μ es la relación entre el valor del tiempo de espera pasiva y la espera activa. P_w denota el valor del tiempo de espera y f es la frecuencia. Tomando como base las funciones anteriores, los costes totales del tiempo de espera, denotados por C_{u-w} , son:

$$C_{u-w} = P_w \left(t_0 + \frac{2\psi}{f} \right) Q, \quad (2.32)$$

donde:

$$t_0 = \begin{cases} 0, & \text{si } f \geq 12\text{veh/h,} \\ t_w, & \text{si } f < 12\text{veh/h,} \end{cases} \quad (2.33)$$

$$\psi = \begin{cases} 1, & \text{si } f \geq 12\text{veh/h,} \\ \mu, & \text{si } f < 12\text{veh/h.} \end{cases}$$

Otro componente clave de los costes en el tiempo de viaje se puede especificar como el producto del valor promedio del tiempo de conducción (P_v) y el tiempo de conducción promedio (T_v). Si l es la duración promedio del viaje y L presenta la duración de la ruta, el tiempo promedio de viaje se puede modelar como una fracción del tiempo del ciclo (t_c):

$$T_v = t_c \left(\frac{l}{2L} \right). \quad (2.34)$$

A medida que el valor del tiempo de conducción de los pasajeros crece con el grado de aglomeración, el impacto de la aglomeración en los costes del tiempo de conducción se puede abordar reemplazando P_v con las funciones de coste de aglomeración no lineales que ya hemos definido:

$$C_{u-v} = P_v t_c \left(\frac{l}{2L} \right) Q. \quad (2.35)$$

2.3. MODELOS DE AMONTONAMIENTO EN REDES DE TRÁNSITO RÁPIDO

El *factor de carga del vehículo*, (N), es el número medio de pasajeros a bordo, que se puede calcular como la relación entre el número total de pasajeros por hora y la frecuencia:

$$N = \left(\frac{Q}{f}\right) * \left(\frac{l}{2L}\right). \quad (2.36)$$

Por convenio, los costes operativos se dividen en costes fijos y variables incurridos en la ejecución de los servicios. Representado por C_{op} , el coste operativo es:

$$C_{op} = c_0 + c_1 B \eta + c_2 V B. \quad (2.37)$$

El primer elemento denota los costes fijos por hora, c_0 . La segunda parte (c_1) es el coste unitario por hora, que está determinado por la cantidad de vehículos requerida en las horas punta. El tercer término, (c_2), es el coste unitario por kilómetro de vehículo. En la práctica, para evitar averías inesperadas, algunos vehículos permanecen sin usar en las cocheras como respaldo. En consecuencia, se introduce una *tasa de reserva de flota*, η , para reflejar esto. Denotando por V la velocidad promedio de viaje, el conjunto de vehículos requeridos (B) se puede formular como:

$$B = f t_c = f \left(\frac{2L}{V}\right) \quad (2.38)$$

Usando (2.38) para eliminar B de (2.37), la función de coste operativo es:

$$C_{op} = c_0 + c_1 \left(\frac{2L}{V}\right) \eta f + c_2 2L f \quad (2.39)$$

En última instancia, el objetivo es minimizar el coste total del sistema con respecto a la frecuencia (f):

$$\underset{f}{Min} TSC = C_{u-w} + C_{u-v} + C_{op} \quad (2.40)$$

En la práctica, la frecuencia de servicio no debe ser ni demasiado baja ni demasiado alta por motivos de funcionamiento y seguridad. En la desigualdad (2.41), f_{\min} es el nivel de servicio mínimo deseado por el público y f_{\max} es la frecuencia máxima factible decidida por la capacidad de la estación y las consideraciones de seguridad:

$$f_{\min} \leq f \leq f_{\max} \quad (2.41)$$

Además de las limitaciones de frecuencia, la capacidad de la línea debe ser suficiente para satisfacer la demanda:

$$\omega \left(\frac{Q}{2f} \right) \leq K, \quad (2.42)$$

donde ω es la fracción de pasajeros que viajan a través de la sección más cargada y K denota el límite de capacidad física. La frecuencia óptima se busca estableciendo la primera derivada de la función de coste total del sistema igual a cero y resolviéndola sujeta a las dos restricciones anteriores.

2.3.3. Ideas principales sobre amontonamiento.

Los medios de transporte público a menudo circulan en condiciones aglomeración excesiva en las grandes ciudades. Si bien el amontonamiento se puede mitigar mediante la mejora de la infraestructura y la expansión de la red (lo cual resulta muy costoso), también se puede evitar mediante medidas de alivio del amontonamiento de bajo coste si éste se formula correctamente.

La ventaja de utilizar funciones de coste de aglomeración no lineales reside en una representación más realista y racional, siendo avaladas por estudios numéricos. Estos estudios muestran que, independientemente de la linealidad de la función de costes de aglomeración, la inclusión de efectos de aglomeración tiene impactos complejos en el desempeño de los servicios de transporte, el coste del sistema y el grado de economías de escala. La incorporación de funciones de coste de aglomeración en la optimización del sistema generalmente genera servicios de tránsito rápido, costes bajos para los usuarios y un uso más eficiente de los vehículos. En este contexto, Mohring describió el efecto que lleva su nombre (Mohring [1972]), y que establece que, si la frecuencia de un transporte público se incrementa con la demanda, dicho aumento en la demanda acorta el tiempo de espera de pasajeros en paradas y estaciones. Dado que el tiempo de espera forma parte de los costes del transporte, el efecto Mohring implica aumentar rendimientos a escala para servicios de transporte urbano planificado. Cuando se consideran los efectos de la sobreocupación, la formulación propuesta por Kraus resulta ser tan importante como el fenómeno descrito por Mohring.

3. Modelo para el diseño de una red de tránsito rápido.

El presente trabajo se basa en el modelo descrito en Laporte et al. [2011a] y en Laporte and Mesa [2019]. Los modelos de programación matemática como los PCLL descritos en la segunda sección del Capítulo 2 son generalmente intratables para la mayoría de instancias, excepto para instancias de tamaño realista desde el punto de vista computacional. Además, sólo pueden capturar algunos aspectos del problema, pues no todas las especificaciones se prestan a una caracterización a través de restricciones lineales.

Con el fin de discernir claramente la evolución y formulación matemática del modelo, la primera parte del estudio considerará el efecto sin sobreocupación, mientras que en la última sección se abordará el efecto de la sobreocupación, proponiendo variables y restricciones adicionales para la implementación numérica del modelo.

3.1. Modelo matemático sin sobreocupación.

En este trabajo se va a considerar el problema de la localización del trazado de una línea de tránsito rápido que maximice el número de viajes captados, teniendo en cuenta los distintos efectos de la sobreocupación.

3.1.1. Datos y notación.

1. En primer lugar, suponemos que para la línea de metro se va a escoger un camino en una red subyacente, $\mathcal{N} = (N, E)$, es decir, que contiene las estaciones o paradas potenciales N , y las conexiones potenciales E entre las estaciones. Para cada arista (no dirigida) $e = \{i, j\}$ se consideran dos arcos (dirigidos): $a = (i, j)$, $\hat{a} = (j, i)$. Podemos suponer que uno de los extremos de la línea está en el conjunto $O \subset N$, y el otro en $D \subset N$. Denotamos por $\delta(k)$ al conjunto de aristas incidentes en el nodo k , y δ^- y δ^+ al conjunto de arcos entrantes y salientes de k , respectivamente.
2. La estructura de coste contiene el coste de construir una estación en el nodo i : b_i , el de construir el tramo e : c_e , y el presupuesto disponible, $C_{\text{máx}}$.
3. La demanda es por pares y está representada por el conjunto $W \subset N \times N$. Cada par $w = (w^s, w^t) \in W$ tiene asociado una demanda g^w que representa el número de viajeros de w^s a w^t . Así, $G = (g^w)$ es la matriz origen-destino. Además, existe un modo de tránsito alternativo

CAPÍTULO 3. MODELO PARA EL DISEÑO DE UNA RED DE TRÁNSITO RÁPIDO.

cuya utilidad viene dada por la matriz $U = (u^w)$. Supondremos que el par w será captado por la nueva línea si su utilidad usando esta línea es mejor que u^w .

4. Denotamos por t_a el tiempo en el vehículo, al recorrer el arco a , por t_k el tiempo de parada en la estación k , y por t_{wait} el tiempo de espera en el andén.

El problema consiste en seleccionar estaciones y tramos que formen una línea, satisfaciendo las restricciones de presupuesto, y de tal forma que se capte al mayor número de pasajeros posible.

3.1.2. Variables.

Contemplamos tres tipos de variables en nuestro modelo:

1. **Variables de diseño:** $y_i = 1$, si se construye una estación en el nodo i , cero en caso contrario.
 $x_e = 1$, si se construye el tramo de la arista e y cero en caso contrario.
2. **Variables de flujo:** $f_a^w = 1$, si la demanda w atraviesa el arco a , y 0 en el caso contrario.
3. **Variables de captura:** $z^w = 1$, si la línea captura el par de demanda w , y 0 en caso contrario.

3.1.3. Función objetivo y restricciones.

El objetivo será maximizar la demanda captada, dado por la siguiente función objetivo, sujeta a las restricciones que se exponen a continuación:

Maximizar la demanda captada por la nueva línea:

$$\max \sum_{w \in W} g^w z^w \quad (3.1)$$

Restricciones:

1. **De presupuesto:**

$$\sum_{e \in E} c_e x_e + \sum_{i \in N} b_i y_i \leq C_{\text{máx}} \quad (3.2)$$

O, alternativamente:

$$\begin{aligned} \sum_{e \in E} x_e &\leq E_{\text{máx}} \\ \sum_{i \in N} y_i &\leq N_{\text{máx}} \end{aligned} \quad (3.3)$$

Las restricciones (3.3) limitan el presupuesto de construcción. La suma del coste de construir una estación, multiplicado por la variable que define si la estación existe o no, más la suma del coste de construir un tramo, multiplicado por la variable que define si el tramo se va a construir o no, tiene que ser menor o igual que el presupuesto disponible para construir la línea.

2. De diseño:

$$x_e \leq y_i, \quad e \in E, i \in e \quad (3.4)$$

$$\sum_{o \in O} y_o = 1, \sum_{d \in D} y_d = 1 \quad (3.5)$$

$$\sum_{e \in E} x_e + 1 = \sum_{i \in N} y_i \quad (3.6)$$

$$\sum_{e \in \delta(k)} x_e \leq 2y_k, k \in N \setminus (O \cup D) \quad (3.7)$$

$$\sum_{o \in O} \sum_{e \in \delta(o)} x_e = 1, \quad \sum_{d \in D} \sum_{e \in \delta(d)} x_e = 1 \quad (3.8)$$

La restricción (3.4) indica que si se construye una estación, se construyen los tramos que inciden en ella. Por su parte, la restricción (3.5) indica que de todas las posibles estaciones, tiene que haber una para cada origen, y una para cada destino. La restricción (3.6) expresa la condición de que exista un árbol en el grafo (número de nodos igual al número de aristas más uno). A su vez, (3.7) denota la condición de que la línea sea un camino. Por último, (3.8) expresa que de todas las estaciones de origen, sólo se construye una. Ídem con las estaciones de destino.

3. Conservación de flujo:

$$\sum_{a \in \delta_w^+(i)} f_a^w - \sum_{a \in \delta_w^-(i)} f_a^w = \begin{cases} z^w, & \text{si } i = w^s \\ -z^w, & \text{si } i = w^t \\ 0, & \text{en otro caso} \end{cases} \quad (3.9)$$

La restricción de flujo (3.9) presenta tres posibles casuísticas. Yendo de arriba a abajo, en el nodo en el que comienza la línea que capta el par de la demanda w , la diferencia del lado izquierdo de la igualdad es la variable de captación z de la demanda. El segundo caso expresa que en el nodo en el que termina la línea que capta el par de la demanda w , la diferencia entre los sumatorios es menos la variable de captación z de la demanda. El último caso indica que

el flujo en los nodos intermedios debe ser cero. Por tanto, la suma del flujo en los arcos que captan demanda entrante en los nodos debe ser igual a la suma del flujo de los arcos que captan demanda saliente.

4. Localización-asignación:

$$f_a^w + f_a^w \leq x_e, \quad e = \{i, j\} \in E, a = (i, j) \quad (3.10)$$

$$\sum_{a \in \delta^+(w^s)} f_a^w \leq y_{w^s}, \quad \sum_{a \in \delta^-(w^t)} f_a^w \leq y_{w^t} \quad (3.11)$$

$$z^w \leq y_k, k \in \{w^s, w^t\} \quad (3.12)$$

Las restricciones anteriores ligan las variables de flujo con las variables de diseño. En concreto, la restricción (3.10) indica que si hay flujo, la variable f_a^w vale 1. Se ha tenido que construir el tramo. (3.11) expresa que si se construye una estación en el origen de un par, sale flujo. Si no se construye, no puede salir flujo. Por último, (3.12) indica que no puede haber flujo de la misma demanda en ambos sentidos.

5. Elección de modo:

$$\sum_{a \in A} t_a f_a^w + \sum_{k \in N \setminus (O \cup D)} t_k \sum_{a \in \delta^+(k)} f_a^w + t_{wait} z^w \leq u^w \quad (3.13)$$

En la restricción (3.13), el tiempo en recorrer una arista se divide en dos: real y percibido. La suma de los tiempos por cada arista, por la variable flujo para ambos tipos de tiempo tiene que ser menor que el tiempo del modo alternativo de transporte contra el que se ha puesto a competir.

3.2. Efectos del amontonamiento de pasajeros.



(a) Amontonamiento en los vehículos.



(b) Amontonamiento en los andenes.



(c) Incremento de tiempos de espera.



(d) Incremento del tiempo de parada.

Figura 3.1.: Distintos efectos de la sobreocupación sobre una red de metro (Fuente: Mesa et al. [2022]).

Los siguientes efectos tienen lugar cuando el número de —potenciales— pasajeros desborda las capacidades de los andenes o de los vagones:

1. Amontonamiento en los vehículos: disminuye la utilidad percibida.
2. Amontonamiento en los andenes: disminuye la utilidad percibida.

3. Exceso de tiempo de espera: cuando hay que esperar al segundo o tercer tren.
4. Incremento de los tiempos de parada: debido al incremento de los pasajeros entrando o saliendo de los vagones.

3.2.1. Amontonamiento en los vehículos.

A continuación, detallamos cada uno de los efectos anteriormente citados. Puesto que la utilidad de los pasajeros ha sido medida mediante el tiempo de recorrido, la no-utilidad producida por la aglomeración incrementará el tiempo de espera (o, en su defecto, la percepción del mismo). Tomando como base la referencia Qin [2014], donde se realiza un análisis sobre el efecto del amontonamiento en el interior de trenes, haremos distinción entre *tranvías* y *metro ligero*, por una parte, y *metro pesado*, por otra.

En el primer caso se define y se utiliza el *factor de carga*, que no es más que la relación entre el número de pasajeros y la capacidad establecida. En contrapartida, en el segundo caso se utiliza la *densidad de pasajeros* que viajan de pie. Centrando nuestra atención en el primer caso, haremos uso de la siguiente función no lineal dependiente del factor de carga, θ , cuya formulación es similar a la que hemos visto en la igualdad (2.29):

$$t(\theta) = t_0 \left(1 + \frac{\rho}{1 + e^{\alpha(1-\theta)}} + \beta e^{\gamma(\theta-\delta)} \right), \quad (3.14)$$

donde α, β, γ y ρ son parámetros a calibrar. El parámetro δ determina el umbral por encima del cual se considera sobrecarga, y t_0 corresponde al tiempo real de recorrer el arco a . La notación empleada en esta fórmula es ligeramente distinta a la usada en el apartado 2.3.2, a fin de distinguir entre los distintos modelos teóricos de los capítulos anteriores y el problema propuesto en este trabajo.

En nuestro caso, podemos considerar el factor de carga por arco a , denotándolo como θ_a , mediante:

$$\theta_a = \frac{\sum_{w \in W} g^w f_a^w}{\text{Capacidad}}, \quad (3.15)$$

entendiendo *capacidad* como la propiedad de los vehículos de poder contener cierta cantidad de pasajeros hasta un límite determinado. La contribución del tiempo en el vehículo a la utilidad del par w en el vehículo a la utilidad del par w en el modo público, $\sum_{a \in A} t(\theta_a) f_a^w$, es ahora no lineal, pues aunque θ_a sea lineal en las variables f_a^w , la función t no lo es, lo cual haría necesario discretizar o utilizar una herramienta de optimización orientada a funciones no lineales (ver la tabla del **Anexo A.2**).

Respecto al segundo caso (considerando la densidad de pasajeros que viajan de pie), se consideran las funciones constantes a trozos del cociente de la densidad de pasajeros por metro cuadrado:

3.2. EFECTOS DEL AMONTONAMIENTO DE PASAJEROS.

$$p_a = \frac{\sum_{w \in W} g^w f_a^w}{\text{Superficie}}, \quad (3.16)$$

es decir:

$$t(p) = t_0 + \sum_{j=1}^i \rho_j, \text{ si } i < p \leq i + 1, i = \{0, \dots, 5\}, \quad (3.17)$$

cuando se supone que la densidad máxima es de 6 personas por metro cuadrado. No obstante, aquí la intuición nos dice que el modelo propuesto en Qin [2014] no parece del todo lógico, pues se propone una función escalonada constante a trozos. Un ejemplo visual del amontonamiento de pasajeros en un vehículo puede verse en la Figura 3.1-(a).

Una interesante perspectiva puede encontrarse en Haywood and Koning [2013] donde, usando datos obtenidos mediante encuestas llevadas a cabo en las líneas 1 y 4 del metro de París, se determinan las pendientes de las rectas correspondientes a cada segmento $([i, i + 1])$ de densidad de pasajeros por m^2) del factor de multiplicación del tiempo de viaje por aglomeración de pasajeros en los vehículos.

En la gráfica de la Figura 3.2 se representa la función lineal a trozos (3.18). Se incluyen también, a título comparativo, el interpolador de Hermite¹ (pchip) y el polinomio interpolador de grado 6 (polyfit)². Para todos los ajustes apreciamos un crecimiento cuasi-exponencial, semejante en conjunto al que obtendríamos con un polinomio de grado bajo. No obstante, en el primer tramo, $x, y \in [0, 1] \times [0.8, 1]$, observamos que los tiempos no varían, y sólo a partir de cierto valor crítico de la densidad de pasajeros comenzará a apreciarse un incremento del factor de multiplicación de tiempos, consecuencia de la saturación de las líneas y de los factores comentados más arriba.

$$\bar{t}(p) = \begin{cases} 1, & 0 \leq p \leq 1 \\ 1 + 0, 1(p - 1), & 1 \leq p \leq 2 \\ 1, 1 + 0, 2(p - 2), & 2 \leq p \leq 3 \\ 1, 3 + 0, 3(p - 3), & 3 \leq p \leq 4 \\ 1, 6 + 0, 4(p - 4), & 4 \leq p \leq 5 \\ 2, 0 + 0, 5(p - 5), & 5 \leq p \leq 6 \end{cases} \quad (3.18)$$

¹La interpolación de Hermite es un método para interpolar puntos de datos como una función polinomial. El polinomio de interpolación de Hermite generado está estrechamente relacionado con el polinomio de Newton, pudiendo derivarse ambos del cálculo de diferencias divididas. La interpolación de Hermite coincide con una función desconocida tanto en el valor observado, como en el valor observado de sus primeras m derivadas. Esto significa que $n(m + 1)$ valores deben conocerse. El polinomio resultante puede tener un grado como máximo de $n(m + 1) - 1$. En el caso general, no es necesario que m sea un valor fijo; es decir, algunos los puntos pueden tener más derivadas conocidas que otros. En este caso, el polinomio resultante puede tener un grado $N - 1$, siendo N el número de puntos de datos.

²La interpolación polinómica denotada como *polyfit* en la mayoría de lenguajes de programación de alto nivel devuelve los coeficientes de un polinomio $p(x)$ de grado n que mejor encajen (o que tengan los mínimos cuadrados) con los datos de y . Los coeficientes de p están en potencias descendentes y la longitud de p es $n + 1$, tal que: $p(x) = p_1 x^n + p_2 x^{n-1} + \dots + p_n x + p_{n+1}$.

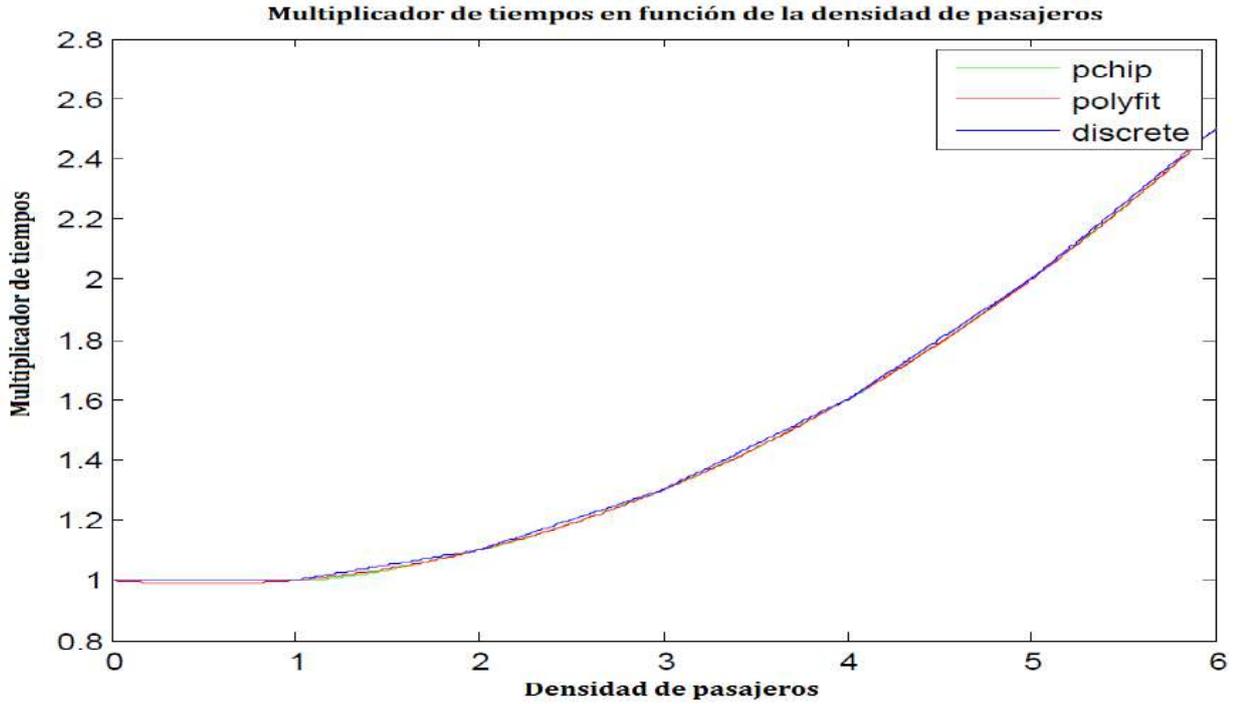


Figura 3.2.: Variación del factor de multiplicación del tiempo de viaje por aglomeración de pasajeros en los vehículos, en función de la densidad de pasajeros (Fuentes: Haywood and Koning [2013] y Mesa et al. [2022]).

La función (3.18) define la tasa de crecimiento lineal, a raíz de los resultados mostrados en Haywood and Koning [2013]. De un modo más genérico, cada una de las funciones componentes de la función anterior puede obtenerse mediante la siguiente expresión:

$$\bar{t}_i(p) = 1 + \sum_{j=0}^{i-1} 10^{-1}j + (10^{-1}i)(p - i), i \leq p \leq i + 1, \quad (3.19)$$

para $i = 0, 1, \dots, 5$, y teniendo en cuenta que $\sum_{j=0}^{-1} = 0$. Por otra parte, si $p_a = \sum_{w \in W} g^w f_a^w / S$ (siendo S la superficie), asumiendo el mismo rango de i y sustituyendo en la expresión definida arriba, tendremos:

$$\bar{t}_i(p_a) = 1 + \sum_{j=0}^{i-1} 10^{-1}j + (10^{-1}i) \left(\frac{\sum_{w \in W} g^w f_a^w}{S} - i \right), \quad (3.20)$$

para:

$$i \leq \frac{\sum_{w \in W} g^w f_a^w}{S} \leq i + 1. \quad (3.21)$$

En base a lo anterior, la ecuación de reparto modal quedará definida como:

$$\sum_{a \in A} t_a \bar{t}_i(p_a) f_a^w + \sum_{k \in N \setminus (O \cup D)} t_k \sum_{a \in \delta^+(k)} f_a^w + t_{wait} z^w \leq u^w \quad (3.22)$$

Resulta sensato percatarse de que en el producto $t_a \bar{t}_i(p_a) f_a^w$, en la restricción anterior, existen términos cuadráticos, $f_a^w f_a^w$, que se pueden linearizar de cara a posibles implementaciones computacionales.

3.2.2. Amontonamiento en los andenes.

Con respecto a la aglomeración en los andenes —véase Figura 3.1-(b)—, asumiendo que el tiempo de espera en un andén pueda resultar más estresante que la aglomeración dentro de determinados vehículos, deberíamos considerar el parámetro t_{wait} multiplicado por un cierto factor de forma.

El amontonamiento en los andenes, más allá de las horas punta, está íntimamente relacionado con el diseño de las propias estaciones: una estación pequeña en una zona periférica cuyos habitantes se dirijan en masa a un evento será susceptible de presentar amontonamiento de pasajeros en sus andenes. También, el número de comodidades que haya en las estaciones (como bancos o puestos de información, por ejemplo) repercutirá en la capacidad que tengan éstas para albergar un mayor número de usuarios. Esto aparentemente contrasta con los factores que aumentan el confort del usuario, pues una experiencia más apacible disminuirá el tiempo de espera percibido. No obstante, la relación entre este tiempo percibido y el amontonamiento en los andenes reside mayoritariamente en las actividades de los propios pasajeros durante su tiempo de espera.

Aquellos pasajeros que viajen en grupo (de dos o más personas) habitualmente serán menos sensibles a la sobreocupación en los andenes, pues serán más propensos a interactuar entre sí, desviando la atención de las incomodidades propias de una aglomeración. En Millonig et al. [2012], se analiza el comportamiento de los pasajeros durante su tiempo de espera en los andenes. Estos se comportan de manera distinta dependiendo de si el tiempo de espera es mayor o menor. La mayoría de las personas no realiza ninguna actividad durante la espera, aguardando hasta que llega el vehículo. Sin embargo, existe otro grupo de pasajeros que durante ese tiempo deambula de un lado a otro por el andén, lo cual puede aumentar el estrés de aquellos usuarios que en principio son indiferentes hacia la espera, dando la sensación de mayor aglomeración y disminuyendo la utilidad percibida frente a otros modos. Otros pasajeros recurren a móviles, tabletas o libros. En la Figura 3.3 se recopila la mayoría de estos grupos.

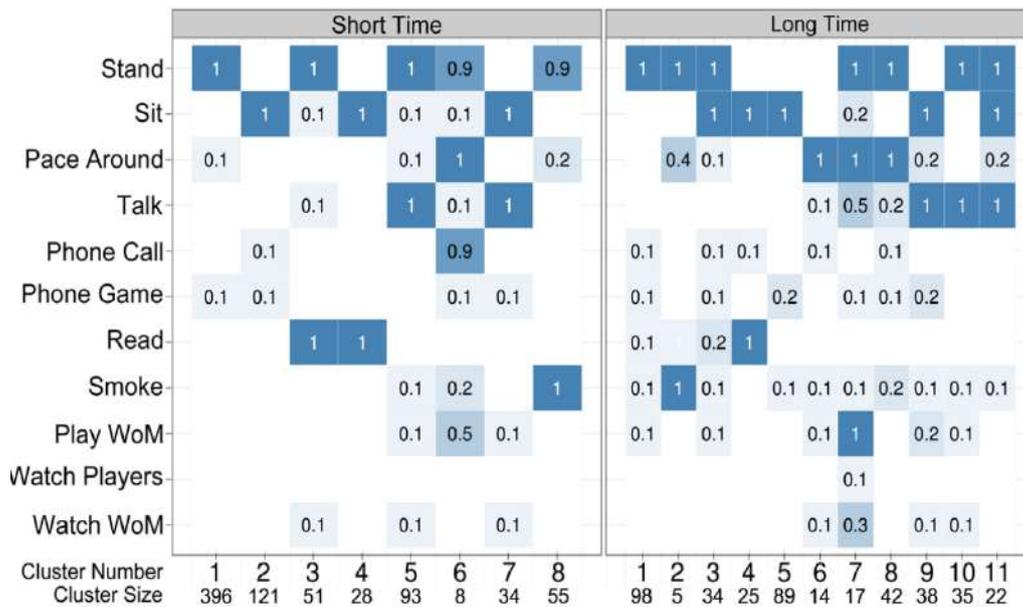


Figura 3.3.: Actividades de pasajeros durante esperas de corta y larga duración. (Fuente: Millonig et al. [2012])

El amontonamiento en los andenes causa mayor impacto en los pasajeros que viajan solos y en los que no disponen de una actividad que ocupe su tiempo durante la espera, frente a aquellos que se sumergen en actividades ociosas. Únicamente es percibido de forma negativa por ambos grupos cuando el nivel de aglomeración es tal, que resulta difícil realizar actividades durante la espera. También son destacables los momentos previos a la llegada del vehículo, pues se produce un movimiento de pasajeros aproximándose a las puertas de acceso, resultando en empujones e invasión del espacio personal, evidenciando la sobreocupación.

3.2.3. Incremento de tiempos de espera.

El tiempo de espera en los viajes en transporte público se percibe de forma negativa por los usuarios. Como parte de la estrategia para mitigar la aversión de los pasajeros hacia una posible espera y la consecuente elección de modos alternativos de transporte, cada vez son más habituales las paradas y estaciones asociadas a varias líneas simultáneamente. Algunos autores (Fan et al. [2016]) destacan que el tiempo de espera percibido por los usuarios en los andenes resulta menor cuando se dispone de un itinerario claro y accesible. Datos empíricos de los mismos autores sugieren que la mayoría de las esperas en las paradas sin servicios alternativos (bancos, techos, máquinas de *vending*, etc.) se perciben en torno a un 30% más largas de lo que realmente son. Estos servicios reducen significativamente los tiempos de espera percibidos.

A este respecto, existen estudios que destacan (Qu et al. [2020]) la variación del tiempo percibido sobre la utilidad de viaje conforme existan comodidades adicionales en las estaciones, desde zonas de reposo (siendo más importante su funcionalidad que su diseño), hasta lugares donde resguardarse de la lluvia, vigilantes y cámaras de seguridad e incluso la ubicación geográfica de una estación (en relación con su entorno y la tasa de delincuencia). Estas comodidades cobran mayor relevancia cuanto más larga sea la espera. Junto con lo anterior, llama la atención cómo diversos factores sociales influyen en el tiempo de espera percibido, imperando en algunos casos la funcionalidad frente al confort o viceversa; igualmente cuando existe sobreocupación. No obstante, de manera global el tiempo de

3.2. EFECTOS DEL AMONTONAMIENTO DE PASAJEROS.

espera percibido dependerá tanto de la hora del día (horas punta, horas de poco tránsito, últimos trenes) como del género del usuario. Respecto a esto último, resulta interesante es el hecho de que las personas de sexo femenino perciben un mayor tiempo de espera que las de sexo masculino, asociado normalmente a una sensación de inseguridad cuando no existe seguridad en la estación o su entorno. Esto incrementa el estrés sufrido por el pasajero, propiciando la elección de modos alternativos o el ingreso al vehículo a través de una estación ubicada en una zona más segura.

Otro factor interesante es la fiabilidad de los modos con raíles frente a autobuses y taxis. Aquellos modos que no interactúan con el resto del tráfico urbano y tienen un área de movimiento perfectamente delimitada tendrán mayor facilidad para ajustarse a un horario determinado.

Estudios más cuantitativos (Fan et al. [2016]) definen el tiempo fuera de los vehículos como suma del tiempo de camino hasta el punto de espera —incluyendo también el tiempo de ingreso y salida de los usuarios hacia o desde los vehículos—, y el tiempo asociado a la propia espera. Cuando se aborda el primero de estos tiempos, es posible determinar empíricamente la velocidad promedio de camino, dependiendo en gran medida del diseño de cada estación. El tiempo de espera es un factor crítico que afecta a la percepción del tiempo de viaje y está íntimamente relacionado con las condiciones de amontonamiento en las estaciones.

Restringiéndonos a las redes de metro, el tiempo de viaje de los pasajeros consistirá en el tiempo dentro y fuera del vehículo, ambos susceptibles de crecer ante el fenómeno de la sobreocupación y difíciles de estimar. Especialmente en las horas punta, a la frustración de los pasajeros por subir a un tren lleno se suma el concepto de “*fail to board*” (FtB), que se da cuando un pasajero no puede ingresar en un tren y debe esperar al siguiente. En general, el tiempo de espera se ve afectado por no poder acceder al primer tren que pase por el andén. Por tanto, sería sensato considerar en el modelo la capacidad que le queda al tren al parar en cada estación, lo cual repercutiría en que t_{wait} ya no sería constante, sino dependiente de la estación de acceso. En la Figura 3.1-(c) podemos apreciar una situación típica a la hora de propiciar un incremento de tiempos de espera. Este tipo de situaciones resultan habituales en grandes núcleos urbanos, especialmente fuera de las horas punta de uso de los vehículos. Asimismo, en la Figura 3.4 se presenta un esquema de la distribución espacio-temporal de los usuarios en un andén antes de ingresar al vehículo.

En los casos aquí considerados se obviarán los pasajeros que realizan transbordo entre un vehículo y otro, sin salir de la estación, siendo este caso fácilmente derivable a partir del caso general aquí expuesto, y su esquema una ampliación de las figuras ya presentadas. En Qu et al. [2020] se realiza una estimación basada en datos estadísticos de la probabilidad del itinerario seguido por los pasajeros considerando el fenómeno FtB, para la red de metro de la ciudad de Chengdu (China), considerando los efectos de la sobreocupación.

Supongamos que un pasajero entra en la estación i de una línea l en un tiempo t_i , y que caminará un tiempo t_{ai} hasta el andén. Si el pasajero pudiese ingresar inmediatamente en el primer tren m en llegar, se cumpliría la siguiente relación:

$$D_{m-1,i,l} \leq t_i + t_{ai} < D_{mil}, \quad (3.23)$$

siendo D el tiempo de llegada a una determinada estación, de una determinada línea, en un determinado

3.2. EFECTOS DEL AMONTONAMIENTO DE PASAJEROS.

Así pues, parte de los datos recogidos por Qu et al. [2020] se recopilan en la Figura 3.5. Destaca el hecho de que, aún con sobreocupación y pérdida de trenes, el mayor tiempo de viaje corresponde a aquel que los usuarios pasan dentro del vehículo. Esto justifica el enfoque numérico adoptado en este trabajo.



Figura 3.5.: Componentes del tiempo de viaje de pasajeros que viajan sin realizar transbordos. (Fuente: Qu et al. [2020])

3.2.4. Incremento del tiempo de parada.

Por último, considerando los tiempos de parada en caso de haber sobreocupación, éstos también dependerán de los flujos de entrada y salida en cada estación. Este último se obtiene sumando todos los elementos anteriores a la diagonal de la fila correspondiente a la estación, mientras que el flujo de entrada requeriría sumar los elementos posteriores a la diagonal. Situaciones que propician el aumento del tiempo de parada pueden ser la celebración de grandes eventos, incidentes en los vehículos y la vía, o cambio de conductores, entre otras. Un factor llamativo es el civismo de los pasajeros, reflejado en actitudes como dejar salir a los pasajeros del vehículo antes de ingresar. En caso de no ser así, se pueden formar aglomeraciones en determinados puntos de acceso, incrementando el tiempo de espera de pasajeros tanto en el andén como dentro del vehículo y, en casos extremos, retrasando el itinerario previsto.

Ante grandes aglomeraciones, no sólo la presencia de pasajeros puede aumentar los tiempos de parada. En ocasiones, puede haber pasajeros que viajen con equipaje pesado, bicicletas o patinetes eléctricos. La dificultad de maniobra de estos pasajeros para salir de un vehículo que se ha ido llenando progresivamente puede ocasionar que otros viajeros ubicados en zonas más alejadas de las puertas no consigan salir al exterior antes del cierre de puertas, incrementando el tiempo de viaje y disminuyendo el confort y la utilidad percibida de la experiencia frente a otros modos. No se han

encontrado en la literatura estudios que consideren el tiempo de viaje de aquellos pasajeros que, por causas ajenas a su voluntad, se vean obligados a permanecer dentro de un vehículo por no poder desembarcar a tiempo.

El tiempo en el vehículo se puede determinar según el horario del tren si se conoce el itinerario del pasajero, y en general las predicciones asociadas a este tiempo serán más fiables y variarán menos que las correspondientes a los tiempos de espera fuera de los vehículos. Denotando por t_{vij} este tiempo, éste se podría estimar como:

$$t_{vij} = A_{mjl} + D_{mil}, \quad (3.27)$$

siendo A_{mjl} y D_{mil} los tiempos de llegada y salida del tren m de la línea l , en las estaciones j e i , respectivamente. Encontramos una ilustración de este caso en la Figura 3.1-(d).

3.3. Modelo matemático con sobreocupación.

En este apartado se va a considerar el problema de la localización del trazado de tránsito rápido que maximice el número de viajes captados teniendo en cuenta como efectos de la sobreocupación el amontonamiento en los vehículos. Tal y como se ha descrito en el apartado anterior, dicho efecto se percibe como una disminución de la utilidad por parte del viajero. A continuación, vamos a exponer el modelo matemático que posteriormente será implementado numéricamente, y comparado con el modelo anterior.

3.3.1. Datos y notación.

1. $P_{\text{máx}}$ es la carga máxima de un arco para que no haya sobreocupación (lo cual realmente se podría hacer dependiente de cada arco. No obstante, el razonamiento es similar y la variación es despreciable).
2. b es el factor que hace aumentar la percepción del tiempo de viaje en un arco por cada pasajero extra cuando hay sobreocupación (la pendiente de la recta en la restricción (3.31)).
3. M es una constante suficientemente grande.

3.3.2. Variables.

1. P_a es la carga del arco a , de modo que:

3.3. MODELO MATEMÁTICO CON SOBRECUPACIÓN.

$$P_a = \sum_w f_a^w. \quad (3.28)$$

Se trata de un nuevo conjunto de variables que ayudarán a que el modelo propuesto sea más legible, de cara a su implementación numérica.

2. $O_a \in \{0, 1\}$ es una variable binaria que denota si hay sobreocupación en el arco a ($O_a = 1$) o no ($O_a = 0$).
3. s_a representa el tiempo en atravesar el arco a , teniendo en cuenta el posible efecto de la sobreocupación.
4. h_a^w es una variable positiva, producto de las variables $s_a \cdot f_a^w$. Se introduce con el fin de eliminar las no linealidades.

3.3.3. Función objetivo y restricciones.

La función objetivo seguirá siendo (3.1). Las restricciones son, junto con (3.2)-(3.13):

1. Restricciones para P_a :

$$P_a - P_{\text{máx}} - M \cdot (O_a) \leq 0, \quad \forall a \in A. \quad (3.29)$$

La restricción (3.29) limita la carga del arco. En caso de que $P_a > P_{\text{máx}}$, se tendrá $O_a = 1$.

2. Restricciones para s_a :

$$s_a \geq t_a, \quad \forall a \in A \quad (3.30)$$

$$s_a \geq t_a + b \cdot (P_a - P_{\text{máx}}) - M \cdot (1 - O_a), \quad \forall a \in A, \quad (3.31)$$

La restricción (3.30) implica que cuando $s_a = t_a$ no habrá sobreocupación, mientras que (3.31) adopta la forma $s_a = t_a + b \cdot (P_a - P_{\text{máx}})$ si hay sobreocupación (es decir, si $P_a > P_{\text{máx}}$). Notemos también que en la restricción de elección de modo habrá que sustituir el parámetro t_a por las variables s_a .

3. Restricciones para h_a^w :

$$h_a^w \leq M \cdot f_a^w \quad \forall w \in W, a \in A \quad (3.32)$$

$$h_a^w \geq s_a - M \cdot (1 - f_a^w) \quad \forall w \in W, a \in A \quad (3.33)$$

$$h_a^w \leq s_a \quad \forall a \in A. \quad (3.34)$$

El conjunto de restricciones (3.32)-(3.34) tiene como fin eliminar no linealidad del modelo. Destaca el hecho de que la primera de ellas implica que el tiempo en atravesar el arco a estará acotado superiormente por una constante positiva. Nótese que esto hace que $h_a^w = s_a \cdot f_a^w$, eliminando la no-linealidad que había en la restricción de elección de modo, y pondremos h_a^w donde había $s_a \cdot f_a^w$. Un análisis más escéptico cuestionaría la utilidad del tercer grupo de restricciones, pues el modelo hará que h_a^w no sea mayor que s_a si eso va a implicar pérdida de captura de demanda.

4. Entorno tecnológico.

Este capítulo describirá en líneas generales todas las herramientas computacionales que hemos usado para simular el modelo de diseño de redes de tránsito rápido teniendo en cuenta los distintos efectos de la sobreocupación. Primeramente, se justificará la elección del lenguaje de programación y la interfaz y editor usados. Por otro lado, una sección específica describirá el conjunto de librerías de optimización empleadas, describiendo la estructura del programa y, finalmente, mencionando de soslayo algunas alternativas igualmente válidas.

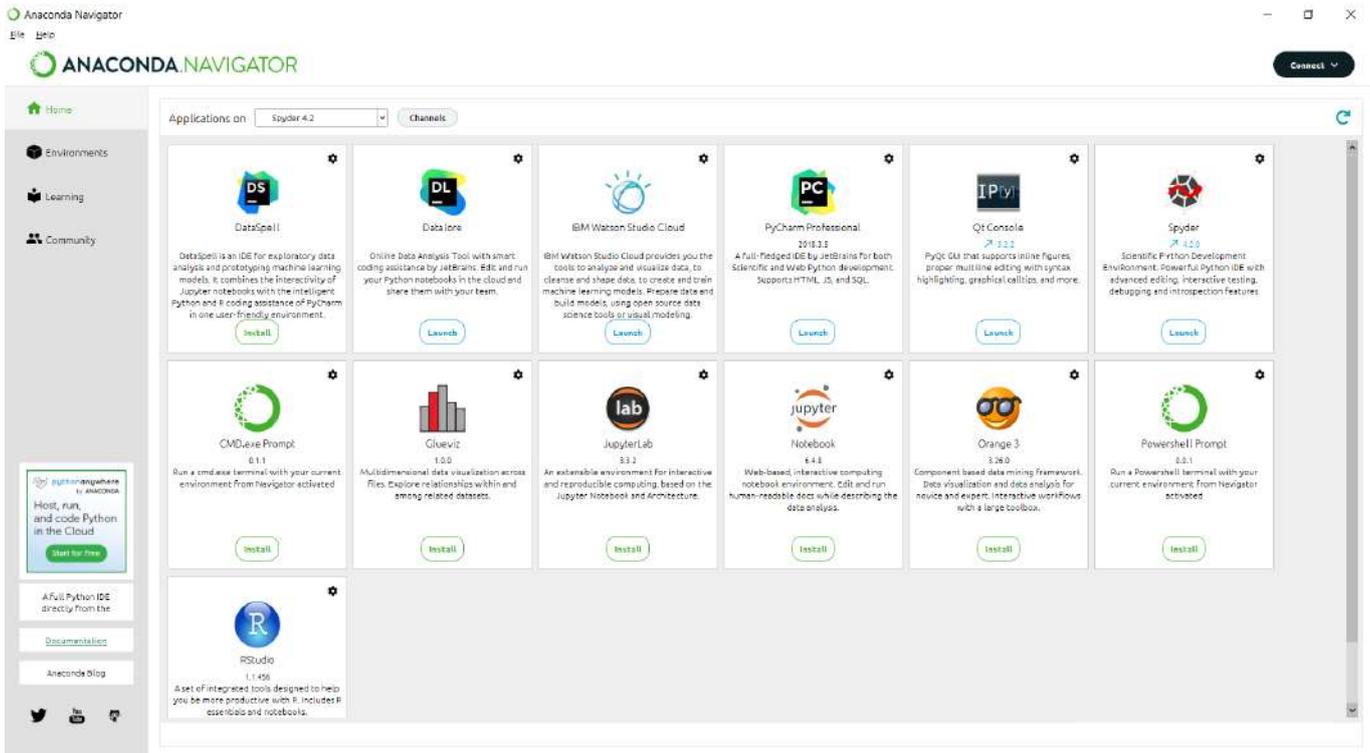
4.1. Lenguaje, interfaz y editor.

Para simular el modelo de optimización propuesto de este trabajo disponemos de una gran gama de herramientas numéricas, la mayoría de carácter gratuito en la actualidad. Más allá de los diferentes solvers contemplados en la tabla del Anexo A.2 —cabe señalar que la elección del solver supone uno de los últimos pasos en el proceso de simulación—, existen numerosos lenguajes de programación que facilitan la implementación de cualquier problema de optimización, estadística y en general cualquier problema dentro del ámbito de la investigación operativa. Si bien en el pasado herramientas como C o Fortran fueron muy populares, poco a poco estos lenguajes han ido dejando paso (lo cual no implica que estén en desuso) a nuevas propuestas como Python, R o Matlab. De este último grupo, nosotros haremos uso del primero.

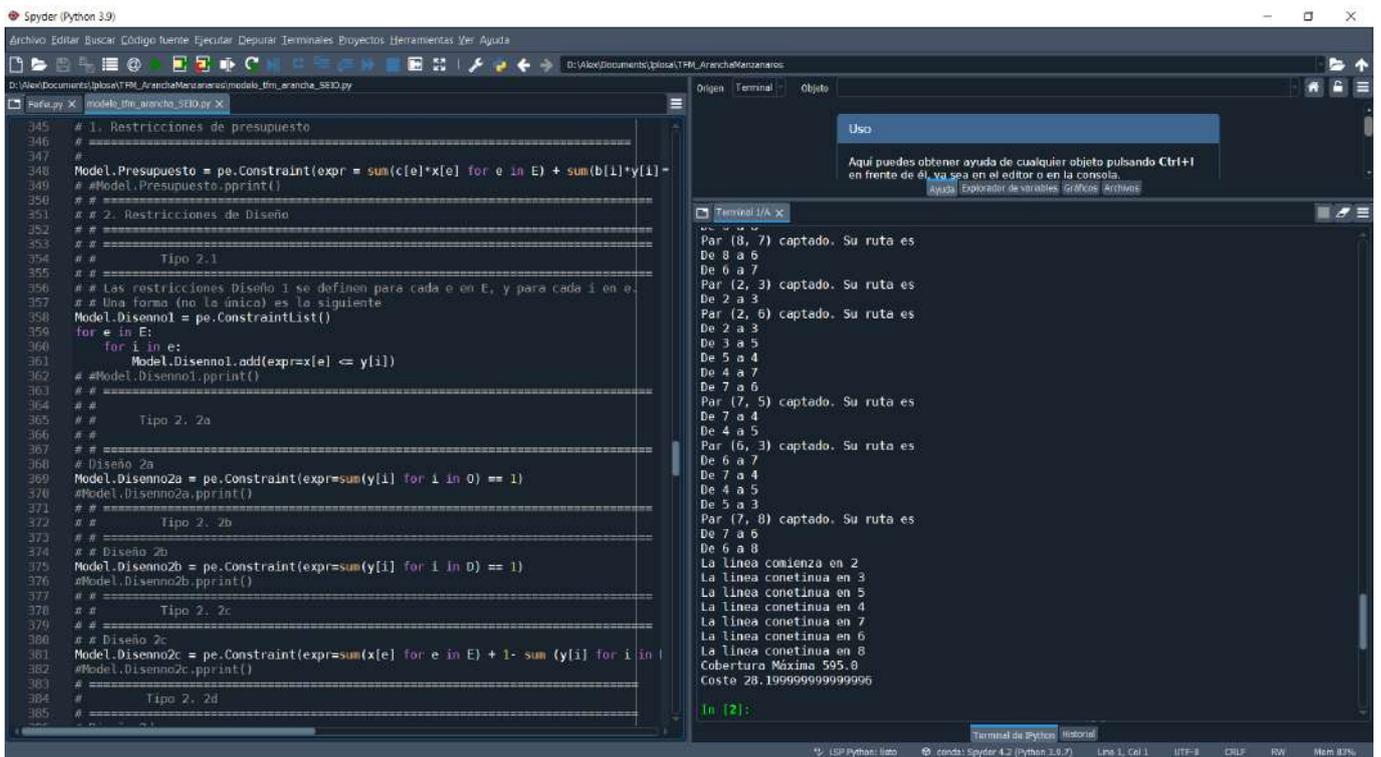
Python es un lenguaje de programación de alto nivel bien establecido y compatible con un énfasis en el desarrollo rápido, la claridad del código y la sintaxis, y un modelo de objeto simple. Entre sus innumerables ventajas, destaca su licencia permisiva de código abierto y las numerosas implementaciones y librerías disponibles sin costo para una amplia variedad de plataformas, tanto convencionales como exóticas. Entre esta amplia gama de herramientas asociadas a Python, encontramos numerosos solvers orientados a la optimización. Todos estos factores, sumados a su versatilidad y accesibilidad, nos llevan a elegir Python como el lenguaje de programación por defecto para las simulaciones numéricas de este trabajo.

Si bien es posible trabajar directamente en la consola de comandos del ordenador, es recomendable utilizar un editor específico para trabajar con Python. Para ello, haremos uso del navegador Anaconda, el cual está implementado para facilitar el acceso a editores de Python y R, tanto en la red como en el escritorio. No obstante, también presenta herramientas orientadas a proyectos de ciencia de datos y machine learning en general. El aspecto de la aplicación de escritorio de esta interfaz puede verse en la parte superior de la Figura 4.1. Por su parte, el editor escogido será Spyder. Spyder es un entorno de desarrollo integrado multiplataforma de código abierto para programación científica en el lenguaje Python, lo que lo convierte en la herramienta ideal para cubrir nuestras necesidades. Un ejemplo de código en Python en la consola de Spyder, así como la impresión de resultados en pantalla, puede verse en la parte inferior de la Figura 4.1.

CAPÍTULO 4. ENTORNO TECNOLÓGICO.



(a) Interfaz principal de Anaconda.



(b) Consola de Spyder durante la ejecución de un código.

Figura 4.1.: Interfaces de Anaconda y Spyder.

4.2. Software de optimización y estructura del programa.

En esta sección se va a aplicar el desarrollo teórico presentado a lo largo del presente trabajo sobre un ejemplo concreto. Se pretende describir de forma minuciosa el modelo matemático con su aplicación directa en todos los pasos del proceso, es decir, exponer de forma paralela la creación del marco teórico y su aplicación mediante el uso de las funciones pertinentes.

4.2.1. Pyomo

De entre las numerosas herramientas desarrolladas para abordar problemas de optimización, Pyomo ha sido nuestra elección final. Pyomo permite al usuario formular problemas de optimización en Python, de una manera similar a la notación usual en optimización matemática. Admite un estilo orientado a objetos para formular modelos de optimización, los cuales son definidos a través de gran variedad de componentes: conjuntos, parámetros, variables, objetivos, restricciones, etc. Las fuentes externas de datos pueden ser definidas usando hojas de cálculo. Pyomo admite modelos abstractos, los cuales son definidos sin datos, y modelos concretos, definidos con datos. En ambos casos, permite la separación del modelo y de los datos. Además, Pyomo es compatible con varios solvers comerciales y de código abierto, como Gurobi y GLPK. En la Figura 4.2 podemos ver el aspecto de la sintaxis de esta herramienta, tanto a la hora de definir la función objetivo y las restricciones, como especificando la elección del solver. Una extensa documentación adicional puede encontrarse en Sandia National Laboratories [2017] y en Hart [2009].

```

# NUEVE: CREAMOS LA FUNCION OBJETIVO
#
#
print('Fin declaración variables. Declaro restricciones y fo')
Model.obj = pe.Objective(expr = sum(g[w]*z[w] for w in W), sense=pe.maximize)
#
#
#DIEZ: RESTRICCIONES. Hoja de calculo restricciones_tfm_atx.xls
#
#
# 1. Restricciones de presupuesto
#
#
Model.Presupuesto = pe.Constraint(expr = sum(c[e]*x[e] for e in E)//
+ sum(b[i]*y[i] for i in N) <= Cmax)
#Model.Presupuesto.pprint()
#
# 2. Restricciones de Diseño
#
#
# Tipo 2.1
#
# Las restricciones Diseño 1 se definen para cada e en E, y para cada i en e.
# Una forma (no la única) es la siguiente
Model.Disenno1 = pe.ConstraintList()
for e in E:
    for i in e:
        Model.Disenno1.add(expr=x[e] <= y[i])
#Model.Disenno1.pprint()
#
#
# Tipo 2. 2a
#
#
# Diseño 2a
Model.Disenno2a = pe.Constraint(expr=sum(y[i] for i in 0) == 1)
#Model.Disenno2a.pprint()
#
# #Resolución
print('Fin declaracion restricciones. Resuelvo el modelo')
opt = pe.SolverFactory('glpk')
opt.solve(Model, tee=True)
results = opt.solve(Model,tee=True)
Model.pprint()
Model.display()
#print value of variables
#print objective value
print(value(Model.obj))
Model.obj.display()
Model.x.display()
Model.y.display()
print(value(Model.obj))

print('MOSTRAR RESULTADOS')
#Primero pongo el valor objetivo
print('Cobertura Máxima', pe.value(Model.obj))

```

(a) Función objetivo y restricciones.

(b) Elección del solver y resolución del modelo.

Figura 4.2.: Ejemplos de sintaxis con Pyomo.

CAPÍTULO 4. ENTORNO TECNOLÓGICO.

El solver escogido es GLPK. Su funcionamiento se basa en el uso del método simplex revisado y el método de punto interior primal-dual para problemas que no son enteros, junto al algoritmo de ramificación y poda combinado con el método de los planos de corte de Gomory (ver **Anexo A.1**) para abordar problemas enteros mixtos (Makhorin [2003]). En este punto, es importante destacar la trascendencia de la elección de la plataforma de desarrollo Pyomo que, como se ha descrito, es una parametrización específica en lenguaje de software libre de Python específico para representar las propiedades de los problemas de optimización. La fortaleza de su estructura reside en que simula en códigos la estructura del modelo matemático basada en conjuntos, es decir, respeta la integridad del marco teórico a la hora de transformarlo en programa. El programa identifica todos los conjuntos del modelo expuesto en el Capítulo 3 y establece bucles donde los índices recorren los conjuntos en el rango que se exponen en el ejemplo que se va a tratar. Se sigue la estructura del modelo, pudiendo aplicar ésta a otro tipo de problemas de transportes sin alterar la estructura del programa, simplemente variando la parametrización. De esta forma, el programa queda estructurado, con una sintaxis sencilla y fácilmente modificable sin tener que recurrir a bucles nuevos, modificando únicamente los conjuntos considerados, sus rangos, o bien alternando las restricciones siguiendo el esquema impuesto, sin más que añadir o eliminar variables y/o cambiando los recorridos de los índices. Si es preciso, se pueden crear conjuntos nuevos que, aunque no constituyan el “core” del problema principal, ayudarían de forma auxiliar incorporando restricciones nuevas.

5. Resultados experimentales.

En este capítulo se mostrarán las soluciones obtenidas a la hora de implementar el modelo propuesto en el Capítulo 3, haciendo uso de las herramientas computacionales descritas en el Capítulo 4. Los resultados expuestos a continuación se han presentado en el congreso recogido en Mesa et al. [2022].

Computacionalmente, abordaremos el problema de la localización del trazado de una línea de tránsito rápido que maximice el número de viajes captados teniendo en cuenta los distintos efectos de la sobreocupación. Tal y como se describe en el Capítulo 3, podemos suponer que para la línea de metro se va a escoger un camino de la red subyacente $N = (N, E)$, representada en la Figura 5.1, es decir, que contiene las paradas potenciales N y las conexiones E entre estaciones. Podemos suponer que uno de los extremos de la línea está en el conjunto O y otro en D , ambos incluidos en el conjunto de 9 nodos. En la parte inferior de la misma figura, vemos la red complementaria.

La estructura de coste de la red, ilustrada en la Figura 5.2, estará constituida de los 9 costes de construir una estación en el nodo i : b_i . A su vez, 13 costes correspondientes de construir el tramo e : c_e . Con el fin de acotar las iteraciones del programa, el coste máximo, $C_{\text{máx}}$, se fija en $C_{\text{máx}} = 1000$. Podemos escribir la matriz origen-destino, $G = (g^w)$, como:

$$G = \begin{pmatrix} 0 & 9 & 26 & 19 & 13 & 12 & 4 & 6 & 4 \\ 11 & 0 & 14 & 26 & 7 & 18 & 3 & 7 & 9 \\ 30 & 19 & 0 & 30 & 24 & 8 & 3 & 9 & 11 \\ 21 & 9 & 11 & 0 & 22 & 16 & 21 & 18 & 16 \\ 14 & 14 & 8 & 9 & 0 & 20 & 12 & 18 & 9 \\ 26 & 1 & 22 & 24 & 13 & 0 & 11 & 28 & 21 \\ 7 & 5 & 6 & 19 & 15 & 13 & 0 & 16 & 14 \\ 5 & 9 & 11 & 16 & 17 & 25 & 17 & 0 & 21 \\ 6 & 8 & 10 & 18 & 11 & 20 & 14 & 20 & 0 \end{pmatrix} \quad (5.1)$$

Además, existe un modo de tránsito alternativo cuya utilidad viene dada por la matriz $U = (u^w)$:

$$U^{\text{OM}} = \begin{pmatrix} 0 & 1,6 & 0,8 & 2 & 1,6 & 2,5 & 4 & 3,6 & 4,6 \\ 2 & 0 & 0,9 & 1,2 & 1,5 & 2,5 & 3,2 & 3,5 & 4,5 \\ 1,5 & 1,4 & 0 & 1,3 & 0,9 & 2 & 3,3 & 2,9 & 3,9 \\ 1,9 & 2 & 1,9 & 0 & 1,8 & 2 & 2 & 3,8 & 4,1 \\ 3 & 1,5 & 2 & 2 & 0 & 1,5 & 3 & 2 & 3 \\ 2,1 & 2,7 & 2,2 & 1 & 1,5 & 0 & 2,5 & 3 & 2,5 \\ 3,9 & 3,9 & 3,9 & 2 & 3 & 2,5 & 0 & 2,5 & 2,5 \\ 5 & 3,5 & 4 & 4 & 2 & 3 & 2,5 & 0 & 2,5 \\ 4,6 & 4,5 & 4 & 3,5 & 3 & 2,5 & 2,5 & 2,5 & 0 \end{pmatrix} \quad (5.2)$$

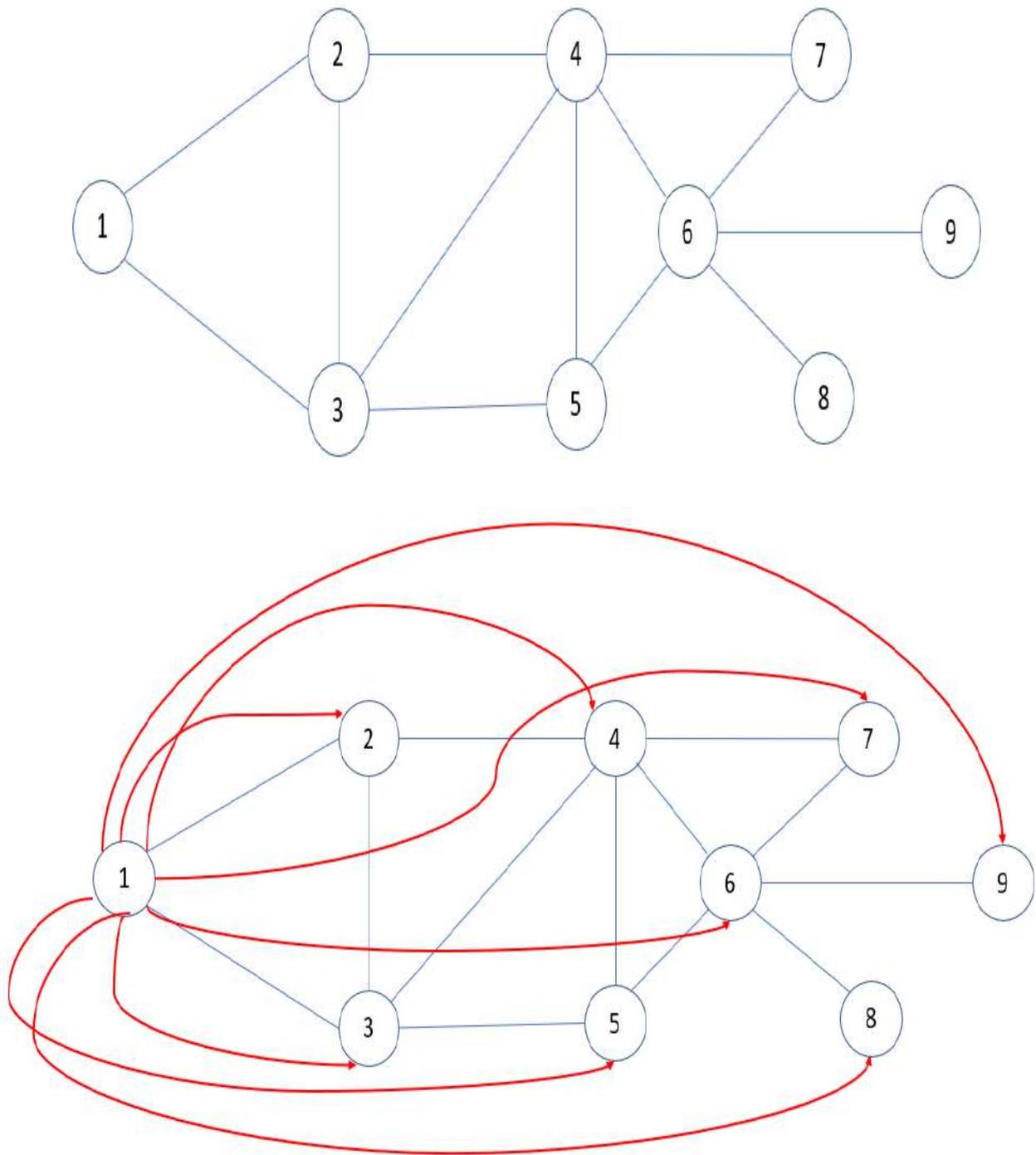


Figura 5.1.: Red subyacente (arriba) y red complementaria (abajo). La red subyacente considerada consta de 9 nodos, 13 aristas y 26 arcos, mientras que la red complementaria constará de 36 aristas y 72 arcos. En la figura, se ilustran las diferentes rutas, unión entre el nodo 1 con el resto de nodos de la red.

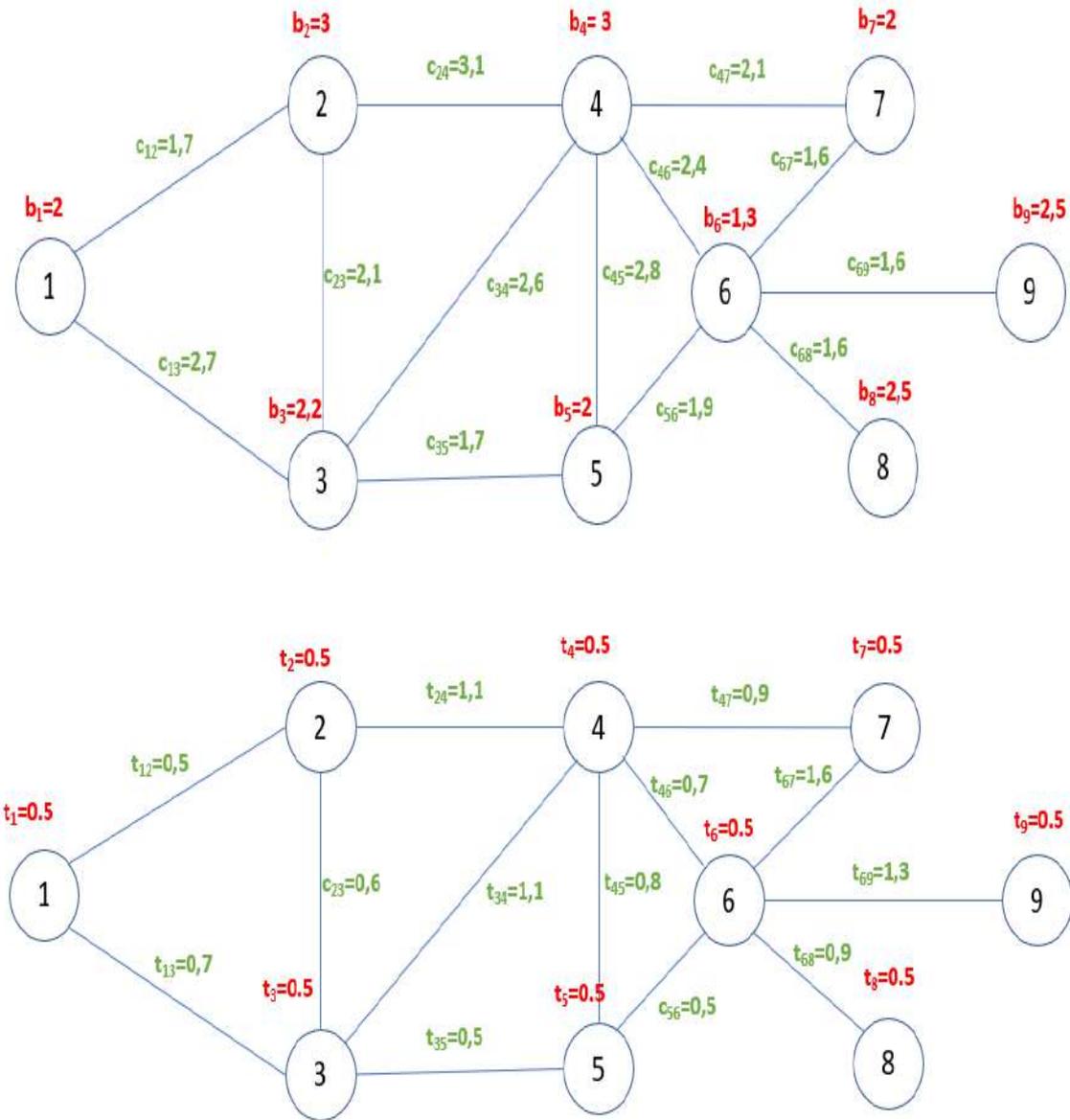


Figura 5.2.: Estructura de costes de la red. La red inferior muestra los costes cuando se considera el tiempo del vehículo en recorrer el arco a , t_a , el tiempo de parada en la estación k , t_k , y el tiempo de espera en el andén, t_{wait} .

5.1. Sin efecto de sobreocupación.

C_{\max}	Nº estaciones	Nº tramos	Cobertura máx. de pares	Línea
< 18	\emptyset	\emptyset	\emptyset	No hay solución.
18	4	3	196	(2, 4)(6, 8)(4, 6)
20	5	4	291	(6, 8)(2, 3)(5, 6)(3, 5)
50	7	6	596	(6, 8)(2, 3)(6, 7)(4, 5)(4, 7)(3, 5)
10^3	7	6	596	(6, 8)(2, 3)(6, 7)(4, 5)(4, 7)(3, 5)
10^4	7	6	596	(6, 8)(2, 3)(6, 7)(4, 5)(4, 7)(3, 5)
10^5	7	6	596	(6, 8)(2, 3)(6, 7)(4, 5)(4, 7)(3, 5)

Cuadro 5.1.: Solución sin sobroocupación. Sensibilidad al presupuesto.

Implementando numéricamente el modelo propuesto en el Capítulo 3, mediante el programa descrito en el Capítulo 4, obtenemos los resultados recogidos en la **Tabla 5.1**. Se han tenido en cuenta tres consideraciones según el tipo de variables:

- Para las variables de diseño:
 - $y_i = 1$ si se escoge el nodo i para hacer una estación, y 0 en caso contrario.
 - $x_e = 1$ si se elige la arista e para construir un tramo entre estaciones. En caso contrario, valdrá cero.
- En las variables de flujo: se impone que $f_a^w = 1$ si la demanda w atraviesa el arco a , y $f_a^w = 0$ en caso contrario.
- Respecto a las variables de captura: $z^w = 1$ si la línea captura el par de demanda w , valiendo 0 si no se cumple esta condición.

Volviendo a los resultados de la tabla, apreciamos que por debajo de cierto valor presupuestario no existe solución para el problema planteado. Para un valor de $C_{\max} = 18$, es posible maximizar el número de pares captados mediante cuatro estaciones, interconectadas por tres tramos y obteniendo una cobertura máxima de 196 u.a. (unidades arbitrarias). Para $C_{\max} = 20$, el número de estaciones y tramos aumenta en una unidad, obteniendo un valor de 291 u.a. para la cobertura máxima de pares. Para valores de $C_{\max} \geq 50$, la solución obtenida será siempre la misma, constando de 7 estaciones conectadas mediante 6 tramos, proporcionando una cobertura máxima de pares de 596 u.a., notablemente superior a los valores obtenidos anteriormente.

Las soluciones obtenidas sobre la red subyacente se encuentran representadas en la Figura 5.3.

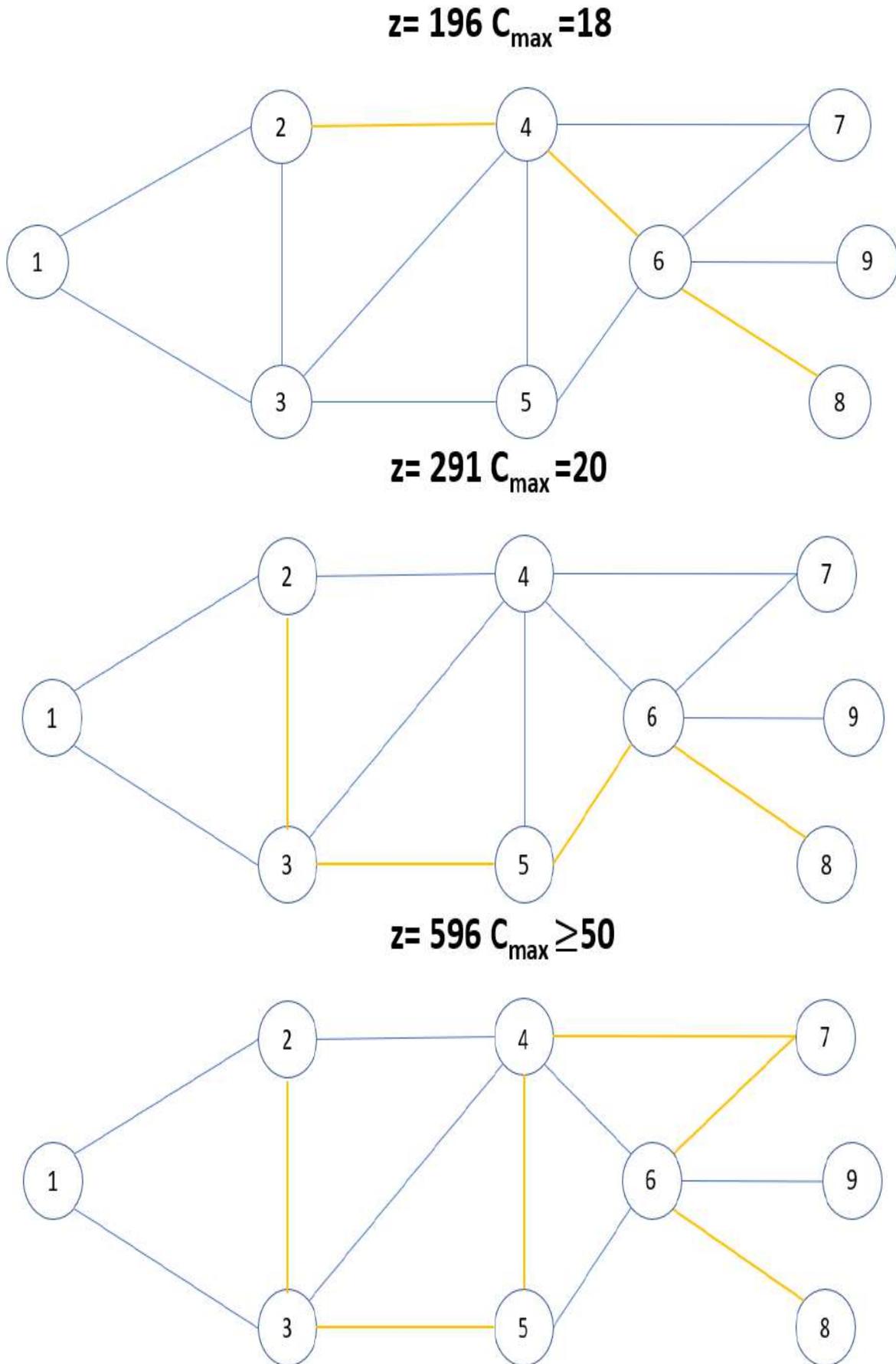


Figura 5.3.: Soluciones obtenidas sin tener en cuenta los efectos de sobreocupación, representadas sobre la red subyacente. Datos recogidos en la **Tabla 5.1.**

5.2. Considerando el efecto de la sobreocupación.

$p_{\text{máx}}$	b	Demanda captada	Línea
≤ 10	0,1	384	(1, 3)(3, 4)(4, 5)(5, 6)(6, 8)
10	0,01	552	(2, 3)(3, 5)(5, 4)(4, 6)(6, 8)
50	0,1	477	(2, 3)(3, 5)(5, 4)(4, 6)(6, 8)
50	0,01	563	(2, 3)(3, 5)(5, 4)(4, 6)(6, 8)
100	0,1	552	(2, 3)(3, 5)(5, 4)(4, 6)(6, 8)
100	0,01	578	(2, 3)(3, 5)(5, 4)(4, 6)(6, 8)

Cuadro 5.2.: Soluciones considerando sobreocupación, para $C_{\text{máx}} = 50$.

Tomando como base el modelo empleado para resolver el problema sin sobreocupación, en esta ocasión se consideran nuevos parámetros como la carga en el arco P_a , relacionada con el número de pasajeros, y la carga máxima en el arco para que no haya sobreocupación, $P_{\text{máx}}$. Las restricciones (3.29)-(3.34) permiten definir los parámetros O_a , s_a y h_a^w auxiliares para el modelo. En la Tabla 5.2 se presenta el parámetro b , que es la pendiente de la recta en la ecuación (3.31), es decir, el factor que hace aumentar el tiempo de viaje en un arco por cada pasajero extra cuando hay sobreocupación y que se puede asociar al coste de construcción de un nodo.

Los datos recogidos en la tabla indican que para un mismo valor del coste debido a la sobreocupación, se puede aumentar la demanda captada disminuyendo los costes por nodo. Para valores bajos de la funciones de coste por sobreocupación, se varían los valores b_i en un solo orden de magnitud (entre 0,1 y 0,01 en todos los casos considerados) para un mismo valor de $p_{\text{máx}}$. Los pares captados en la red subyacente serán los mismos a partir de cierto valor de la función de coste por sobreocupación (~ 50 unidades arbitrarias), pudiendo aún así maximizar la demanda captada para dichos pares reduciendo adecuadamente el coste por nodo.

Observamos que, teniendo en cuenta los efectos del amontonamiento en los vehículos, sólo obtenemos dos conjuntos solución para las líneas que maximizan la demanda captada. Las soluciones obtenidas sobre la red subyacente se encuentran representadas en las Figuras 5.4, 5.5 y 5.6.

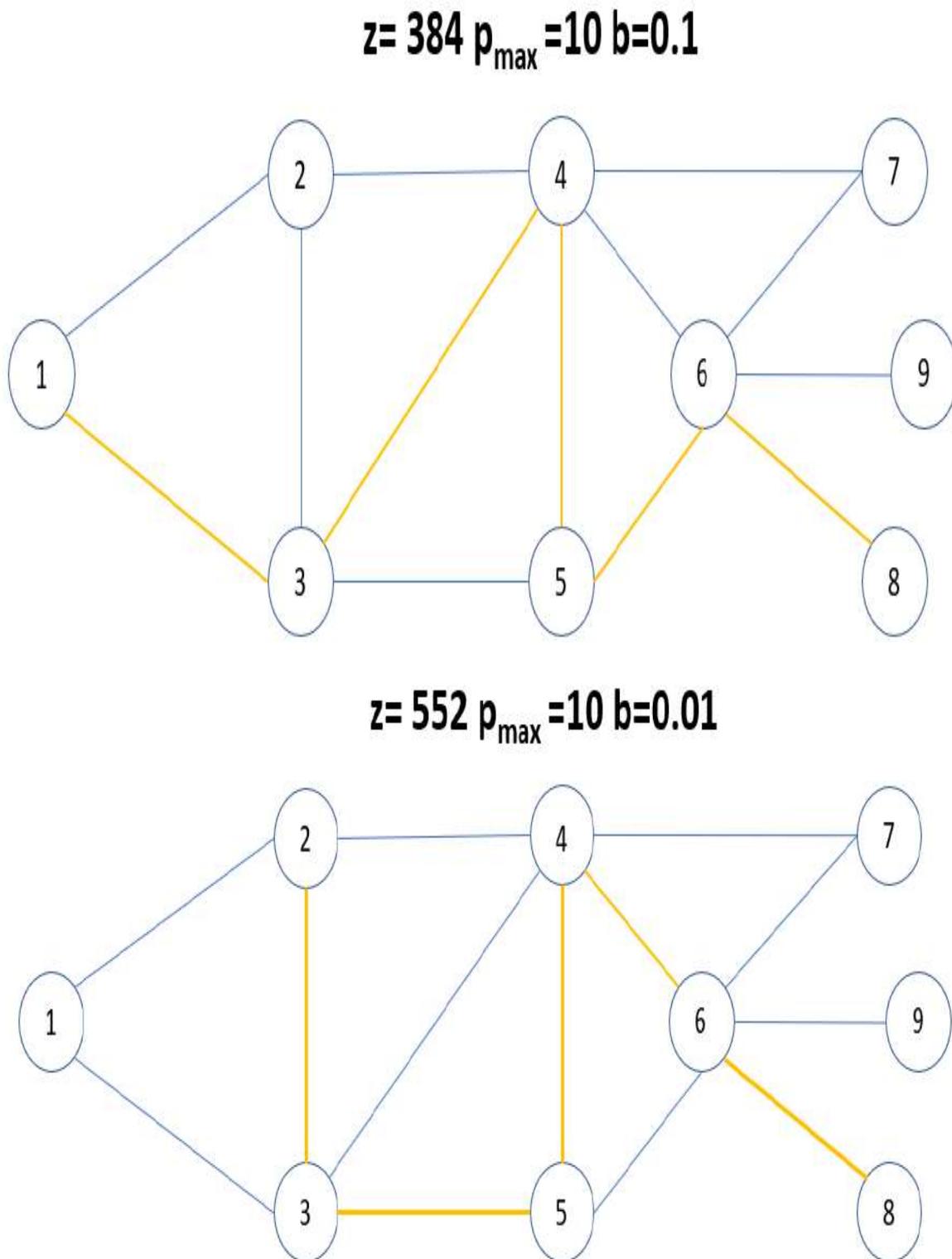


Figura 5.4.: Soluciones obtenidas teniendo en cuenta los efectos de sobreocupación, representada sobre la red subyacente. Datos recogidos en la **Tabla 5.2.**

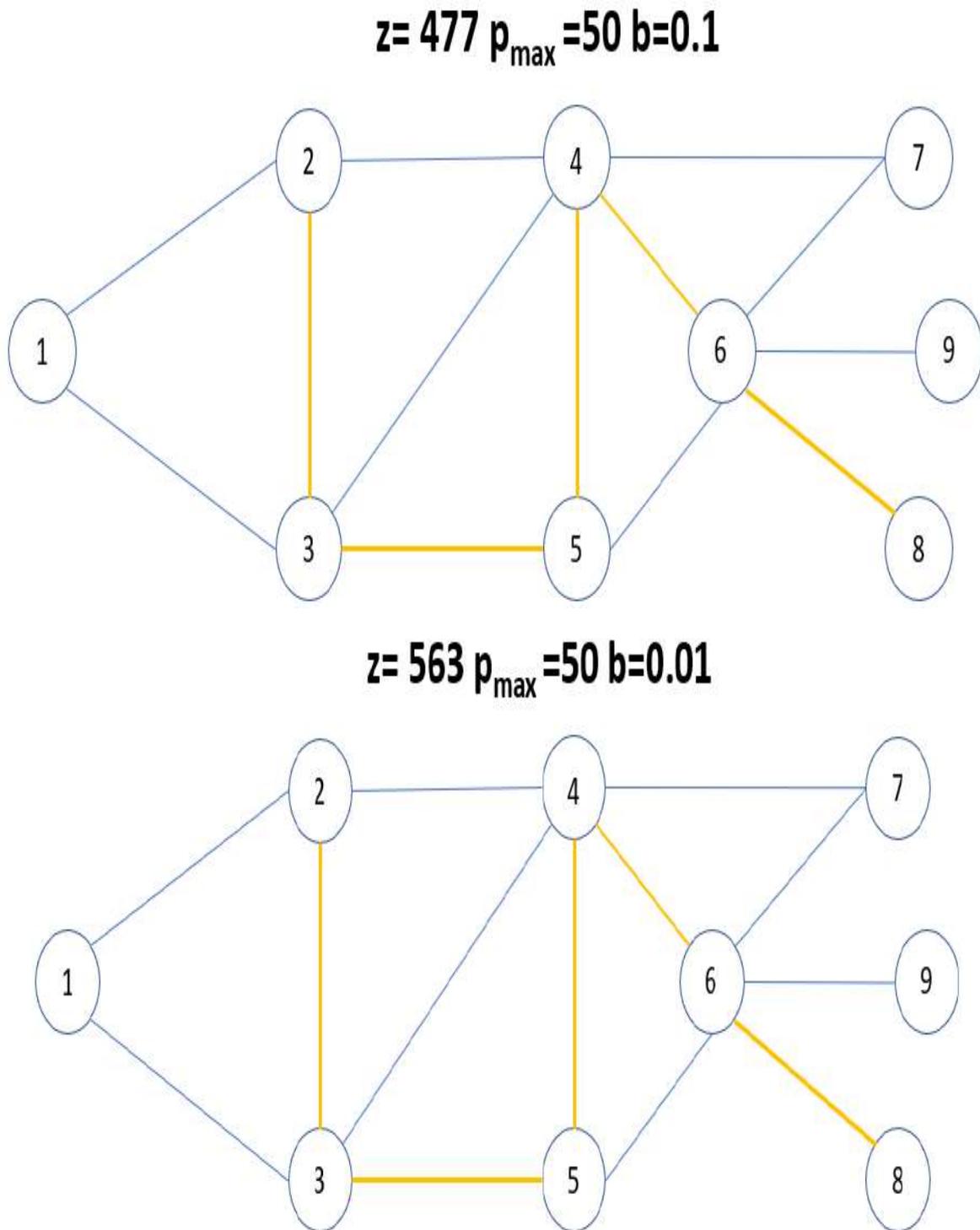


Figura 5.5.: Soluciones obtenidas teniendo en cuenta los efectos de sobreocupación, representada sobre la red subyacente. Datos recogidos en la **Tabla 5.2.**

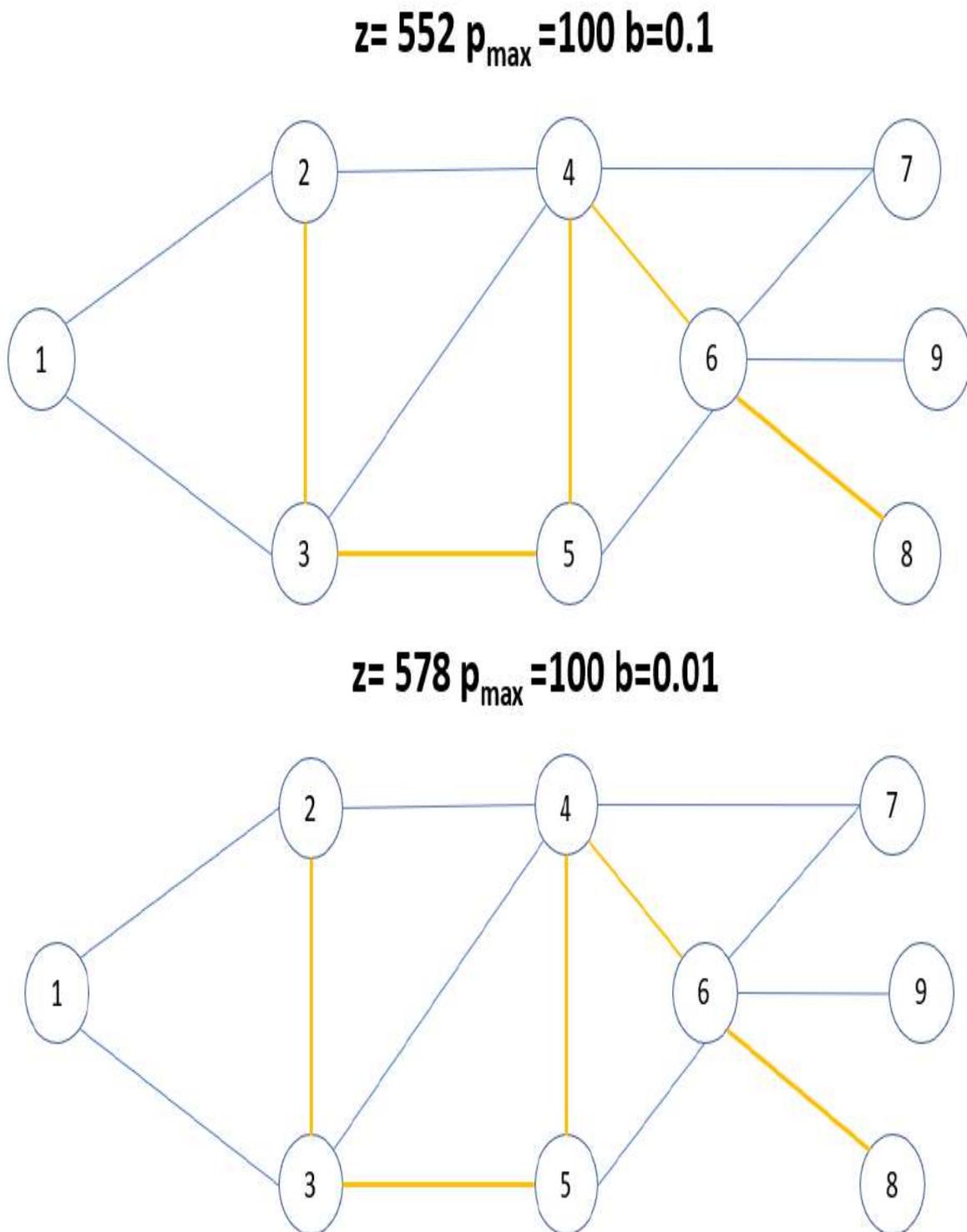


Figura 5.6.: Soluciones obtenidas teniendo en cuenta los efectos de sobreocupación, representada sobre la red subyacente. Datos recogidos en la **Tabla 5.2.**

5.3. Red ampliada.

En esta sección se considerarán redes más grandes, ampliando la red subyacente de la Figura 5.1. Así pues, el criterio ha sido ir añadiendo un nodo adicional a la red, hasta que los tiempos de resolución con glpk sean superiores a una hora. En la Tabla 5.3 se recogen las líneas obtenidas y la demanda captada para cada una de estas redes. La fila sombreada corresponde a un tiempo de simulación con $t_{glpk} > 1h$. Es preciso destacar que para los dos últimos casos se han modificado convenientemente los parámetro de sobreocupación: por un lado, el valor de $p_{máx}$ (carga máxima en el arco para que no haya sobreocupación) se aumenta a partir del cuarto nodo adicional. Por otra parte, el parámetro b (aumento del tiempo de viaje por cada pasajero extra cuando hay sobreocupación) se reduce ligeramente. Estas modificaciones están justificadas, puesto que consideramos la línea en conjunto a la hora de calcular resultados, y tiene sentido que la capacidad de ésta para tolerar y distribuir una gran afluencia de pasajeros aumente al añadir más tramos y estaciones. Llama la atención que una línea con un mayor número de nodos no implica necesariamente un aumento de la demanda captada, cuando no se modifican apropiadamente los parámetros de sobreocupación (como se aprecia en los casos de 10, 11 y 12 nodos). Por otro lado, cuando se es poco restrictivo con estos parámetros, se aprecia un posible aumento del número de pares captados, como se ocurre en los casos de 13 y 14 nodos. Las soluciones obtenidas se representan en las Figuras 5.7, 5.8 y 5.9.

Nº nodos	$p_{máx}$	b	Demanda captada	Línea
10	100	0,01	772	(2, 3)(3, 5)(5, 4)(4, 7)(7, 6)(6, 8)(8, 10)
11	100	0,01	774	(2, 3)(3, 5)(5, 4)(4, 7)(7, 6)(6, 8)(8, 10)
12	100	0,01	678	(1, 3)(3, 5)(5, 4)(4, 7)(7, 11)(11, 9)(9, 6)(6, 8)(8, 10)
13	500	0,005	563	(1, 3)(3, 5)(5, 4)(4, 7)(7, 6)(6, 8)(8, 10)(10, 9)(9, 11)(11, 12)
14	500	0,005	1762	(1, 3)(3, 5)(5, 4)(4, 7)(7, 6)(6, 8)(8, 10)(10, 12)(12, 9)(9, 11)(11, 14)

Cuadro 5.3.: Soluciones de la red ampliada, teniendo en cuenta el efecto de la sobreocupación, para un coste máximo fijo $C_{máx} = 50$.

A su vez, se usará un solver alternativo con el fin exponer las diferencias en tiempo de cómputo cuando se emplean distintas herramientas.

Nº nodos	10	11	12	13	14
t_{glpk} (s)	90,8	109,9	1947,7	1666,0	9481,0
t_{CPLEX} (s)	0,81	1,08	6,14	13,6	22,61

Cuadro 5.4.: Tiempos de resolución de las redes ampliadas, para distintos solvers.

La Tabla 5.4 contiene los tiempos de resolución para las redes ampliadas usando glpk y el solver comercial CPLEX, perteneciente a la empresa IBM. Es muy destacable la diferencia entre los tiempos empleados para resolver el modelo según el solver. Se observa que CPLEX proporciona soluciones en tiempos mucho más cortos que glpk. Esto es debido a una mezcla de factores, como la limitación de glpk a modelos de programación lineal (LP) y de enteros-mixtos (MIP), frente a la versatilidad de CPLEX para casi cualquier tipo de problema, tal y como se muestra en la tabla del Anexo A.2. La potencia de cómputo necesaria cuando se emplea esta nueva herramienta es mucho menor, en parte debido a su carácter comercial y el consecuente desarrollo constante, lo que la vuelve más idónea a la hora de abordar problemas más complejos que el que se ha considerado en este trabajo.

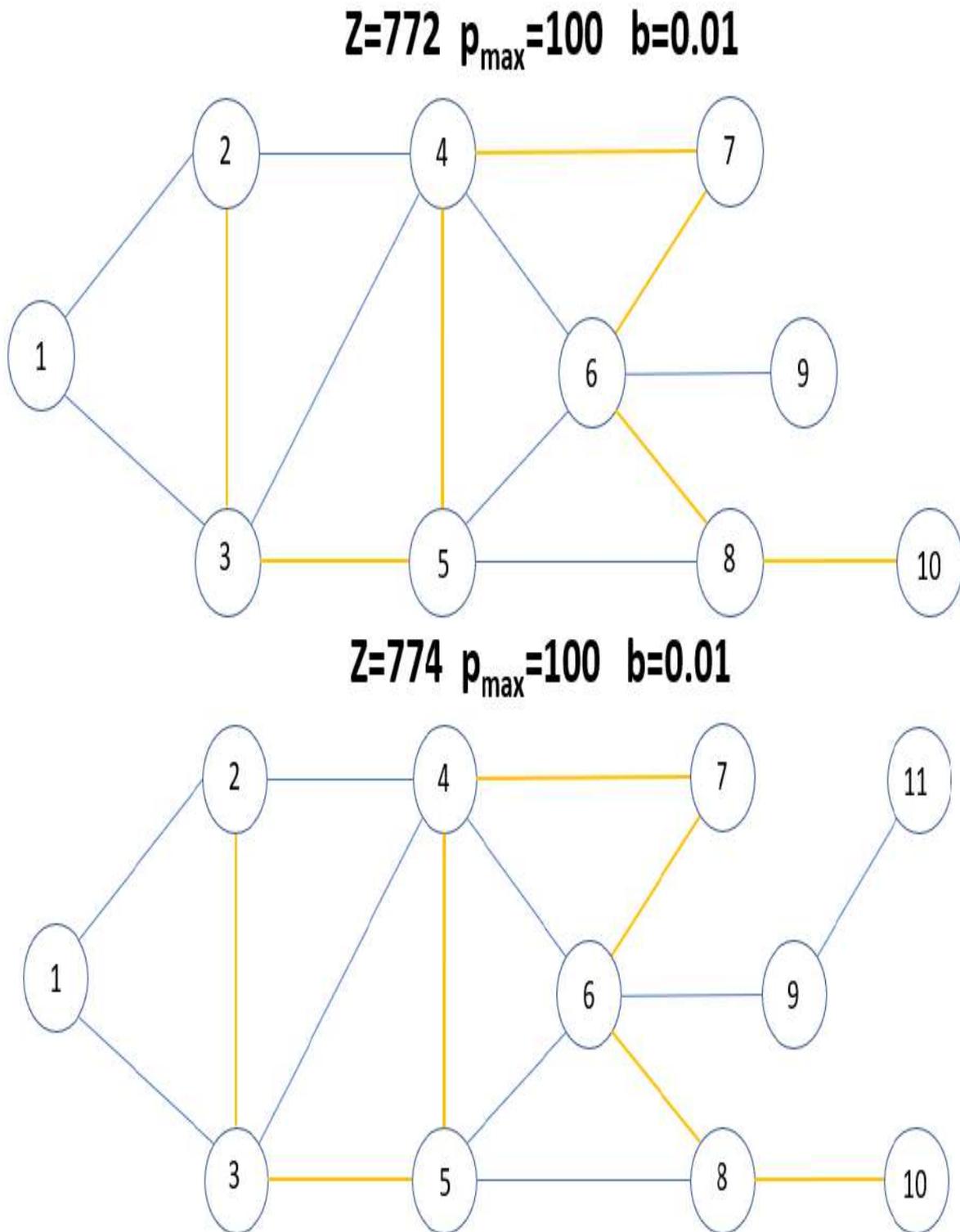


Figura 5.7.: Soluciones obtenidas teniendo en cuenta los efectos de sobreocupación, representada sobre la ampliación de la red subyacente, con 10 y 11 nodos. Datos recogidos en la **Tabla 5.3.**

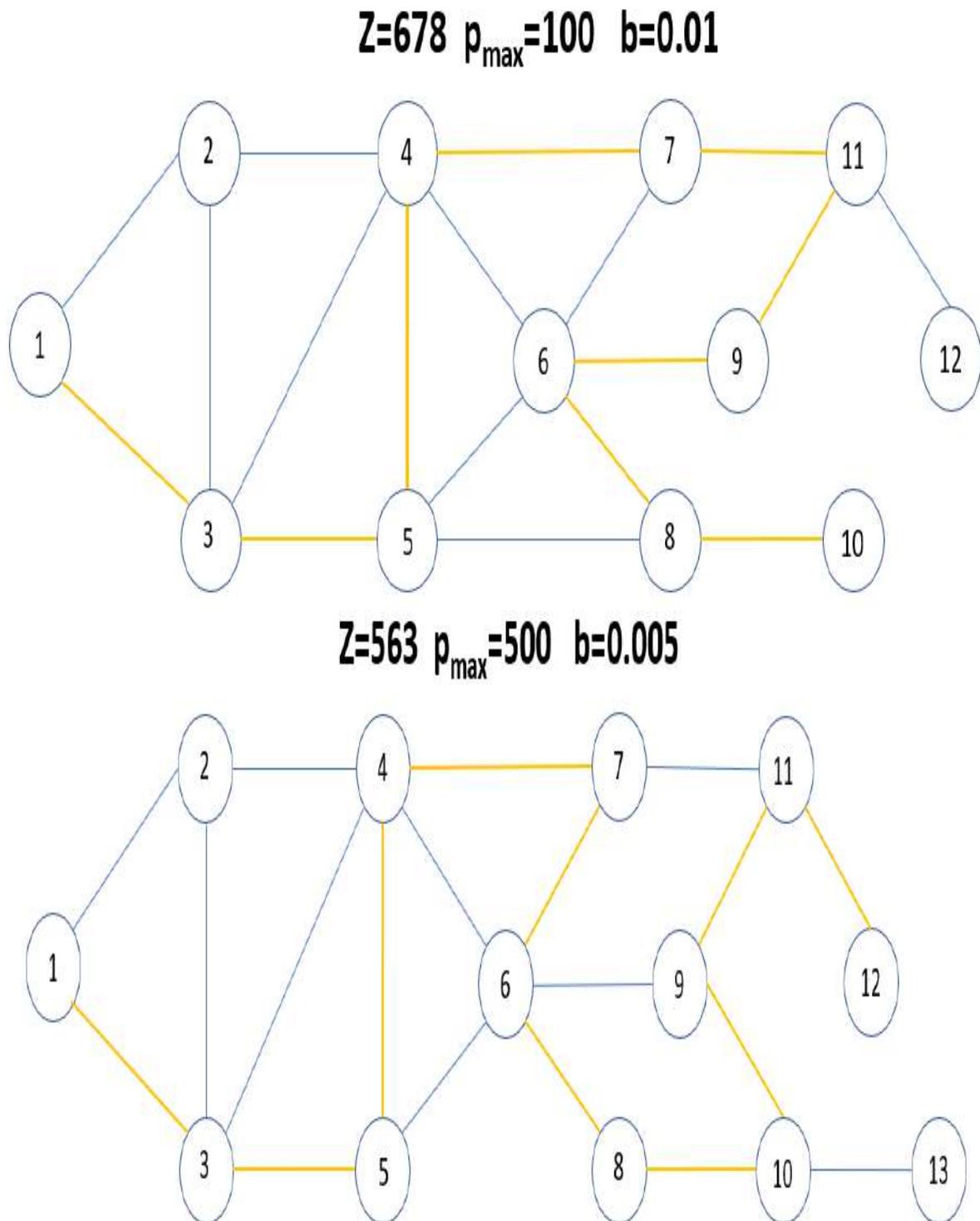


Figura 5.8.: Soluciones obtenidas teniendo en cuenta los efectos de sobreocupación, representada sobre la ampliación de la red subyacente, con 12 y 13 nodos. Datos recogidos en la **Tabla 5.3.**

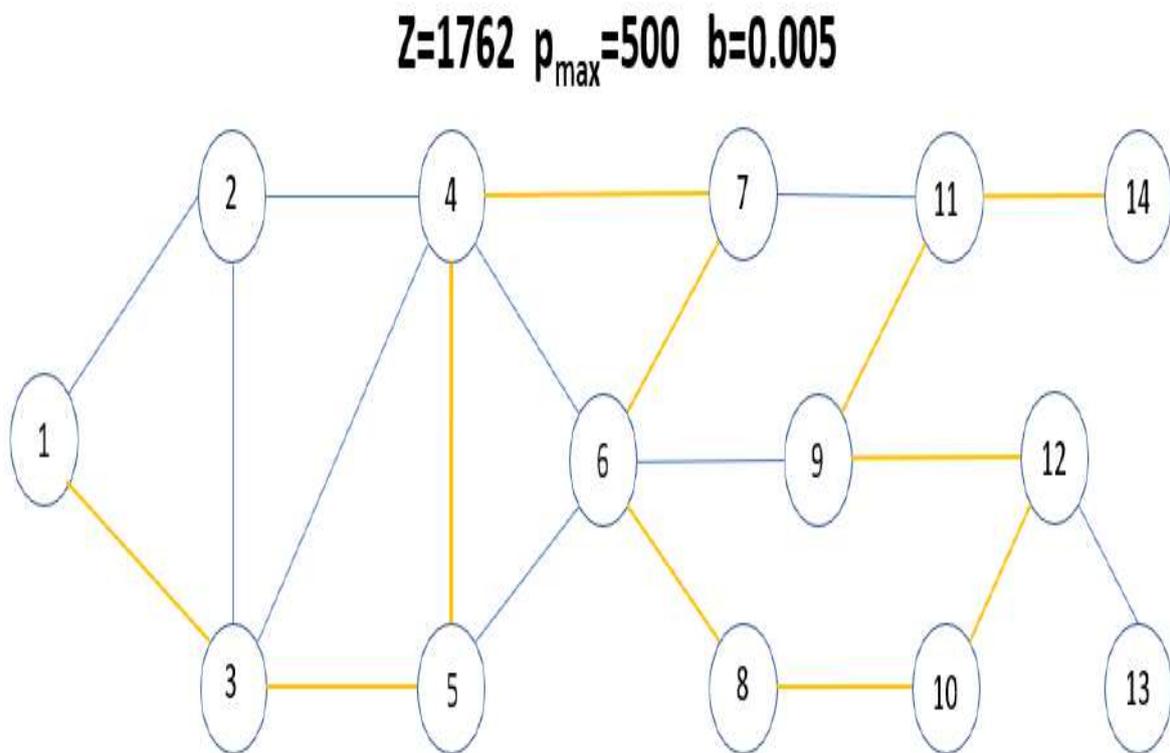


Figura 5.9.: Solución obtenida teniendo en cuenta los efectos de sobreocupación, representada sobre la ampliación de la red subyacente, con 14 nodos. Datos recogidos en la **Tabla 5.3.**

6. Ampliación del problema y posibles horizontes de estudio.

Es posible extender el trabajo realizado desde dos perspectivas claramente diferenciadas: la perspectiva algorítmica y la conceptual.

Existen enfoques más complejos para resolver modelos de programación matemática para problemas de diseño de redes de tránsito rápido, como el que se ha estudiado en este trabajo. Por ejemplo, aquellos algoritmos basados en la descomposición de Benders a menudo se combinan con otras técnicas algoritmos genéticos o modelos que hacen uso de algoritmos de recocido simulado (*simulated annealing*), entre otros. También, cuando el coste computacional no es un aspecto relevante en el problema a tratar y se da mayor peso al análisis estadístico, aparecen numerosas técnicas que abarcan desde la programación dinámica, hasta las heurísticas de ramificación y poda o los algoritmos de colonia de hormigas.

Desde el punto de vista conceptual, el paso inmediato para continuar el estudio realizado sería modificar el modelo, esta vez para estudiar los efectos de la sobreocupación debido al amontonamiento de pasajero en los andenes o, si damos mayor relevancia al tiempo, los incrementos de tiempos de espera en las estaciones y durante las paradas.

Un enfoque muy interesante también puede ser considerar las funciones no lineales de penalización de coste (2.27) y (2.30), en lugar de la presentada en la ecuación (3.14). Un estudio más ambicioso podría consistir en proponer nuevas funciones no lineales que se adaptasen mejor al problema estudiado, cuya formulación combinaría argumentos cualitativos y cuantitativos, mediante la recopilación previa de datos empíricos y su análisis pertinente.

7. Resumen y conclusiones principales.

El diseño de sistemas de tránsito rápido es un proceso complejo que involucra la participación de numerosos actores. Estos proyectos a menudo se caracterizan por sus altos costes e incertidumbre. La formulación de modelos matemáticos precisos y el diseño de algoritmos para tales problemas suponen un reto difícil, ya que los objetivos y las restricciones no están tan bien definidos como se requiere en los modelos de investigación operativa. Para abordar estos problemas, se pueden emplear numerosas técnicas analíticas que permiten ayudar en la toma de decisiones o resolver algunas variantes específicas. No obstante, el juicio y la intervención humana siguen siendo factores críticos en el proceso de planificación. Si además se tienen en cuenta los efectos de la sobreocupación, la complejidad del problema se incrementa notablemente, y es aquí donde la subjetividad condiciona el modelado matemático. Es por ello que en los últimos años se han realizado importantes avances metodológicos en el área de la planificación de la localización de las líneas de tránsito rápido, desarrollándose varios índices de calidad y proponiéndose modelos matemáticos de realismo creciente, algunos de los cuales pueden resolverse directamente usando solver estándar o mediante elaborados algoritmos.

El análisis realizado en este trabajo se apoya en la teoría de grafos y la optimización matemática. En el Capítulo 2 se hace hincapié en la extraordinaria laboriosidad a la hora de plantear un proyecto de diseño de redes de tránsito (rápido). Desde un punto de vista cuantitativo, los modelos de localización de redes y de líneas requieren algoritmos elaborados como los STPP y los PCSTPP. Estos algoritmos a menudo trascienden el ámbito de aplicación para el que fueron diseñados.

El concepto que da sentido a este trabajo, más allá de un análisis genérico del problema del diseño de redes, es el de **sobreocupación**. En urbes densamente pobladas, numerosos estudios han encontrado que el amontonamiento dentro de los vehículos de transporte público ha ido empeorando progresivamente en los últimos años. El hacinamiento crónico dentro de los vehículos no sólo es causado por la falta de infraestructura física, sino también por la provisión inadecuada de servicios. Cuando se produce el fenómeno de sobreocupación, la utilidad percibida por el usuario disminuye, puesto que cambia el confort, que se traduce en tiempo de viaje percibido. Se activa la competencia modal, incluida en la utilidad. Por ejemplo, un pasajero puede preferir ir en autobús o en taxi en lugar de soportar los efectos del hacinamiento. El usuario se puede negar a utilizar la línea de tránsito rápido porque no soporta la congestión.

En general, son pocas las situaciones reales donde el problema se puede solventar mediante programación lineal. En el caso del problema estudiado, formulamos un problema de programación lineal entera-mixta. Problemas aún más complejos derivan hacia el terreno no-lineal, ocasionando cambios apreciables tanto en las funciones objetivo como en las restricciones asociadas.

Todas las implementaciones numéricas en este trabajo se han realizado en lenguaje Python, debido a su versatilidad y sintaxis intuitiva. La librería escogida para implementar el modelo propuesto ha sido Pyomo, debido a su fácil uso y carácter gratuito. De entre los numerosos solvers existentes

CAPÍTULO 7. RESUMEN Y CONCLUSIONES PRINCIPALES.

recopilados en la tabla del **Anexo A.2**, se ha escogido **glpk** para realizar las simulaciones. En general, es posible acceder a una gran variedad de solvers para problemas lineales cuando el uso de herramientas comerciales no es una opción. Las conclusiones que acompañan a esta elección son cuanto menos interesantes: en primer lugar, los solvers de código abierto (gratuitos) pueden servir como una alternativa viable a los solvers comerciales como CPLEX, a expensas de presentar ciertas limitaciones como su uso en problemas no lineales o un tiempo de cómputo más lento.

Dada la prevalencia de los vehículos de transporte superpoblados, se ha realizado un pequeño estudio tanto cuantitativo como cualitativo, definiendo de manera pertinente una función de coste de sobreocupación en los vehículos para diferentes modos de transporte. En el Capítulo 3 se ha definido primeramente la función objetivo (3.1) junto con sus respectivas restricciones de presupuesto, diseño, conservación de flujo, localización-asignación y elección de modo, sin tener en cuenta los efectos de la sobreocupación, y acto seguido se ha procedido a explorar posibles causas de la sobreocupación, como pueden ser el amontonamiento en vehículos y andenes, o los incrementos de tiempos de espera y de parada. El caso analizado en este trabajo se centra en el amontonamiento en los vehículos. A fin de simular numéricamente los efectos del amontonamiento, se han definido nuevas variables que linealizan el modelo y permiten su implementación numérica, en detrimento del número de restricciones que pasará a ser mayor.

Los resultados obtenidos sin sobreocupación indican que debe haber un presupuesto mínimo para que el modelo planteado tenga solución, y que en un primer rango de valores la solución sobre la red subyacente considerada implicará un número reducido de estaciones, tramos y cobertura máxima de los pares captados, coincidiendo todos los resultados hacia 7 estaciones, 6 tramos y una cobertura máxima de 596 pares captados a partir de un presupuesto máximo $C_{\text{máx}} = 50$.

El modelo con sobreocupación permite alternar otra serie de parámetros, como la percepción del tiempo de viaje y la carga en un arco cuando hay sobreocupación (b y P_a , respectivamente). Por simplicidad, se ha fijado un mismo valor de $C_{\text{máx}}$ para todos los casos considerados con sobreocupación. A la luz de los resultados experimentales obtenidos para el modelo propuesto, todo parece indicar que para un mismo valor de la carga máxima que puede soportar un arco antes de considerar sobreocupación, es posible aumentar la demanda captada disminuyendo en el modelo la percepción del tiempo de viaje por parte de los usuarios. Esto resulta apreciable para valores bajos de $P_{\text{máx}}$, variando b en un factor únicamente un orden de magnitud (entre 0,1 y 0,01 en todos los casos considerados). No obstante, los pares captados en la red subyacente serán los mismos a partir de cierto valor de este parámetro de sobreocupación (~ 50 unidades), pudiendo aún así maximizar la demanda captada para dichos pares modificando adecuadamente los parámetros de sobreocupación del modelo. Este resultado cobra sentido si tenemos en cuenta la relevancia de una estación frente a otras en función de su ubicación estratégica dentro de un área metropolitana, sugiriendo que tal vez las inversiones en aquellas estaciones más céntricas deban ser mayores, debido a factores como una mayor afluencia e interconexiones con otros medios, frente a aquellas estaciones ubicadas en áreas periféricas dentro de una metrópolis y que presumiblemente requieren de una menor complejidad logística y económica. Si bien la sobreocupación se puede mitigar mediante la mejora de la infraestructura y la expansión de las redes, con el enorme coste económico que eso implica, también es posible evitarlo mediante medidas de alivio de sobreocupación de bajo coste: desde la perspectiva más cuantitativa, haciendo más apacible la experiencia del usuario tanto en trenes como en estaciones para que no perciba disutilidad en su experiencia debido a las aglomeraciones (y minimizando el riesgo de que se active la competencia modal); desde un punto de vista matemático, modelando la sobreocupación correctamente, considerando en cierto modo que la red es más “tolerante” hacia este fenómeno. Los

resultados obtenidos pueden ser útiles a la hora de ayudar a los tomadores de decisiones a examinar mejor los impactos de la aglomeración de vehículos en la planificación y diseño de redes de tránsito.

Con el fin de extender los resultados obtenidos, se han simulado varias redes ampliadas partiendo la red subyacente, aumentando el número de nodos y arcos, teniendo en cuenta la sobreocupación. Se ha observado que el número de pares captados aumenta siempre y cuándo el aumento del tamaño de la red se vea reflejado en una modificación apropiada de los parámetros de sobreocupación, siendo éstos más tolerantes al tratarse de redes más grandes. Paralelamente, tras la resolución con glpk, se ha empleado un solver comercial (CPLEX) para resolver todas estas redes, observando un tiempo de cómputo mucho menor, lo que destaca la importancia de escoger la herramienta adecuada cuando se trabaja con problemas más complejos que el considerado en este trabajo. Vemos pues, que se ha demostrado matemáticamente que el resultado de la línea elegida varía si se tienen en cuenta los efectos de la sobreocupación en el modelo, con lo cual en este trabajo se pone de manifiesto de manera objetiva lo descrito anteriormente. Además, la diferencia en las líneas obtenidas es sustancial, y depende de los parámetros de sobreocupación que se han incluido en el modelo.

Una discreta ampliación de este estudio podríamos encontrarla en la implementación sobre nuestro modelo de las funciones de coste (2.27) y (2.30). Modelos más ambiciosos incluirían las funciones no-lineales no sugeridas u otras, acompañadas de nuevas restricciones y una red subyacente mucho mayor a la considerada. Adicionalmente, la continuación natural de este trabajo reside en plantear funciones objetivo y restricciones similares para modelar el amontonamiento en los andenes y los incrementos de tiempos de espera y de parada en los trenes.

En el futuro, el diseño de redes de tránsito rápido debe tener en cuenta los distintos efectos de la sobreocupación, ya que como se ha demostrado, el modelo arroja resultados distintos (línea, estaciones, tramos, demanda captada) cuando se obvia el hacinamiento. Por tanto, el reto está en la formulación de los distintos efectos de la sobreocupación sobre el modelo expuesto en el presente trabajo, donde se ha explorado una pequeña fracción del gran y prometedor horizonte que representan los problemas de diseño de redes de tránsito rápido en las áreas de la optimización, la algoritmia y la investigación operativa, teniendo en cuenta las técnicas existentes en la actualidad.

Referencias

- R. Anand, D. Aggarwal, and V. Kumar. A comparative analysis of optimization solvers. *Journal of Statistics and Management Systems*, 20(4):623–635, 2017. doi: 10.1080/09720510.2017.1395182. URL <https://doi.org/10.1080/09720510.2017.1395182>.
- A. Costa, J.-F. Cordeau, and G. Laporte. Steiner tree problems with profits. *INFOR*, **44**, 07 2005. doi: 10.1080/03155986.2006.11732743.
- A. De los Santos. *Rapid transit network design and line planning*. PhD thesis, Universidad de Sevilla, Sevilla, 2014.
- Y. Fan, A. Guthrie, and D. Levinson. Waiting time perceptions at transit stops and stations: Effects of basic amenities, gender, and security. *Transportation Research Part A: Policy and Practice*, **88**: 251–264, 06 2016. doi: 10.1016/j.tra.2016.04.012.
- M. Gendreau, G. Laporte, and J. Mesa. Locating rapid transit lines. *Journal of Advanced Transportation*, **29**:145 – 162, 03 1995. doi: 10.1002/atr.5670290202.
- R. Gilmore and R. Gomory. A linear programming approach to the cutting stock problem i. *Oper Res*, **9**, 01 1961. doi: 10.1287/opre.9.6.849.
- F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers Operations Research*, **13**(5):533–549, 1986. ISSN 0305-0548. doi: [https://doi.org/10.1016/0305-0548\(86\)90048-1](https://doi.org/10.1016/0305-0548(86)90048-1). URL <https://www.sciencedirect.com/science/article/pii/0305054886900481>. Applications of Integer Programming.
- W. Hart. *Python Optimization Modeling Objects (Pyomo)*, volume 47, pages 3–19. 01 2009. ISBN 978-0-387-88842-2. doi: 10.1007/978-0-387-88843-9_1.
- L. Haywood and M. Koning. Estimating crowding costs in public transport. *SSRN Electronic Journal*, 04 2013. doi: 10.2139/ssrn.2256332.
- M. Jamaledin, H. Ibrahim, M. S. Ferwati, M. F. Khamidi, and M. Indraganti. Pre-transit oriented development assessment guidelines for assessing metro station dependent component area. *Designs*, **6**(3), 2022. ISSN 2411-9660. doi: 10.3390/designs6030056. URL <https://www.mdpi.com/2411-9660/6/3/56>.
- M. Kraus. Discomfort externalities and marginal cost transit fares. *Journal of urban Economics*, **29** (2):249–259, 1991.
- G. Laporte and J. Mesa. The design of rapid transit networks. *Location Science*, **3**:241–255, 01 2015. doi: 10.1007/978-3-319-13111-5_22.
- G. Laporte and J. Mesa. *Location Science*, chapter 22: The Design of Rapid Transit Networks, pages 687–703. Springer International Publishing, 01 2019. ISBN 978-3-030-32176-5. doi: 10.1007/978-3-030-32177-2_24.

Referencias

- G. Laporte, J. Mesa, and F. Ortega. Assessing the efficiency of rapid transit configurations. *TOP: An Official Journal of the Spanish Society of Statistics and Operations Research*, **5**:95–104, 02 1997. doi: 10.1007/BF02568532.
- G. Laporte, A. Marin, J. Mesa, and F. Perea. Designing robust rapid transit networks with alternative routes. *Journal of Advanced Transportation*, **45**:54 – 65, 01 2011a. doi: 10.1002/atr.132.
- G. Laporte, J. Mesa, F. Ortega, and F. Perea. Planning rapid transit networks. *Socio-Economic Planning Sciences*, 45:95–104, 09 2011b. doi: 10.1016/j.seps.2011.02.001.
- A. Lucena and M. Resende. Strong lower bounds for the prize collecting steiner problem in graphs. *Discrete Applied Mathematics*, 141:277–294, 05 2004. doi: 10.1016/S0166-218X(03)00380-9.
- A. Makhorin. GLPK Reference Manual. <http://www.gnu.org/software/glpk/glpk.html>, 2003.
- J. Mesa, A. Manzanares, and F. Perea. Rapid transit network design considering passenger crowding. Granada, Spain, 2022. SEIO, June.
- A. Millonig, M. Sleszynski, and M. Ulm. Sitting, waiting, wishing: Waiting time perception in public transport. 09 2012. doi: 10.1109/ITSC.2012.6338777.
- S. Mitchell, M. OSullivan, and I. Dunning. Pulp: a linear programming toolkit for python. *The University of Auckland, Auckland, New Zealand*, 65, 2011.
- H. Mohring. Optimization and scale economies in urban bus transportation. *The American Economic Review*, **62**(4):591–604, 1972.
- F. Qin. Investigating the in-vehicle crowding cost functions for public transit modes. *Mathematical Problems in Engineering*, **2014**, 2014.
- H. Qu, X. Xu, and S. I.-J. Chien. Estimating wait time and passenger load in a saturated metro network: A data-driven approach. *Journal of Advanced Transportation*, **2020**:1–17, 08 2020. doi: 10.1155/2020/4271871.
- Renaissance Planning Group. A framework for transit oriented development in florida. 03 2011.
- Robert Fourer, David Gay, Brian Kernighan. AMPL Reference Manual. <https://portal.ampl.com/docs/archive/first-website/BOOK/CHAPTERS/24-refman.pdf>, 2003.
- S. Roy, S. Mitchell, and F. Peschiera. Optimization with PuLP. <https://coin-or.github.io/pulp/index.html>, 2009.
- Sandia National Laboratories. Pyomo Documentation 6.4.1. <https://pyomo.readthedocs.io/en/stable/>, 2017.

A. Anexos.

A.1. Método de los planos de corte de Gomory.

El método de los planos de corte es un procedimiento habitual utilizado para encontrar soluciones enteras de un problema lineal. Fue propuesto por Raplh Gomory en la década de los 50' como un método para resolver problemas de programación en enteros y enteros-mixta (Gilmore and Gomory [1961]). No obstante, no fue hasta la década de los 90' cuando se popularizó, cuando los avances en el mundo de la algorítmica mostraron las ventajas de combinar este método con las técnicas de ramificación y poda, soslayando numerosas inestabilidades numéricas. Hoy en día, se trata de una técnica presente de manera directa o indirecta en casi cualquier herramienta de optimización orientada a resolver problemas de programación lineal de enteros-mixta (MILP). Los cortes de Gomory se generan partir de un cuadro simplex, mientras que una cantidad considerable de otros tipos de corte resultan costosos o incluso NP-difíciles de separar.

Supongamos un problema de programación lineal entera en su forma canónica:

Maximizar:

$$c^T x \tag{A.1}$$

s.a. las restricciones:

$$Ax \leq b \tag{A.2}$$

$$x \geq 0, \text{ siendo todos los } x_i \text{ enteros.} \tag{A.3}$$

El método surge al eliminar primeramente el requisito de que los x_i sean números enteros, y resolviendo después el problema de programación lineal asociado para obtener una solución factible básica. Una interpretación geométrica consiste en interpretar la solución como un vértice del politopo convexo formado por todos los puntos factibles. Si este vértice no es un punto entero, el método encuentra un hiperplano con el vértice en un lado y todos los puntos enteros factibles en el otro —en la misma línea, se concibe cada restricción como un hiperplano equivalente—. Esto luego se agrega como una restricción lineal adicional para excluir el vértice encontrado, creando un programa lineal modificado. Finalmente, se resuelve el nuevo programa y se repite el proceso hasta que se encuentra una solución entera.

El uso del método simplex para resolver problemas lineales originará un conjunto de ecuaciones de la forma:

APÉNDICE A. **ANEXOS.**

$$x_i + \sum \bar{a}_{i,j} x_j = \bar{b}_i, \quad (\text{A.4})$$

donde x_i es una variable básica y las x_j son las variables no-básicas. Reescribiendo la ecuación para agrupar términos enteros y fraccionarios a cada lado, tendremos:

$$x_i + \sum [\bar{a}_{i,j}] x_j - [\bar{b}_i] = \bar{b}_i - [\bar{b}_i] - \sum (\bar{a}_{i,j} - [\bar{a}_{i,j}]) x_j. \quad (\text{A.5})$$

Para cualquier punto entero en la región factible, el lado derecho de la ecuación anterior será menor a la unidad, mientras que el lado izquierdo será entero. Por tanto, el valor común deberá ser menor o igual que cero. Teniendo en cuenta esto, la desigualdad:

$$\bar{b}_i - [\bar{b}_i] - \sum (\bar{a}_{i,j} - [\bar{a}_{i,j}]) x_j \leq 0 \quad (\text{A.6})$$

deberá cumplirse para cualquier punto entero en la región factible. Por consiguiente, las variables no-básicas serán iguales a los ceros en cualquier solución básica si x_i es un no-entero para la solución básica x ,

$$\bar{b}_i - [\bar{b}_i] - \sum (\bar{a}_{i,j} - [\bar{a}_{i,j}]) x_j = \bar{b}_i - [\bar{b}_i] > 0. \quad (\text{A.7})$$

Entonces, la desigualdad anterior excluye la solución factible básica y, por lo tanto, se trata de un corte con las propiedades deseadas. Al introducir una nueva variable de holgura x_k para esta desigualdad, se agrega una nueva restricción al programa lineal, es decir:

$$x_k + \sum ([\bar{a}_{i,j}] - \bar{a}_{i,j}) x_j = [\bar{b}_i] - \bar{b}_i, x_k \geq 0, \text{ siendo } x_k \text{ un entero.} \quad (\text{A.8})$$

A.2. Dominio de aplicación de diferentes solvers.

Solver	Desarrollador	LP	CP	MIP	MILP	MIQP	MISOCP	NLP
CPLEX	IMB	✓	✓	✓	✓	✓	✓	✓
XPRESS	FICO	✓	✓	✓	✓	✓	✓	✓
Gurobi	GuRoBi	✓	×	✓	✓	✓	✓	×
XA	Sunset Software Technologies	✓	×	✓	×	×	×	×
KNITRO	Ziena Opt. Inc.	✓	×	×	×	×	×	✓
GLPK	GNU Project	✓	×	✓	×	×	×	×
lp_solve	Michel Berkelaar	✓	×	✓	✓	✓	×	×
CBC	COIN-OR Project	✓	×	×	✓	✓	×	×
Lindo	Lindo Systems Inc.	✓	×	✓	✓	✓	×	✓
OML	Ketron Management	✓	×	×	✓	✓	×	×
FortMP	OptiRisk Systems	✓	×	×	✓	✓	×	×
FrontLine	FrontLine Systems	×	×	✓	×	×	×	×
CONOPT	ARKI Consulting	✓	×	×	✓	✓	×	✓
MOSEK	Mosek Aps	✓	×	✓	✓	×	✓	×
MIDACO	ASTER Labs	×	×	✓	×	×	×	✓
CLP	COIN-OR Project	✓	×	✓	✓	✓	×	×
MINOS	Stanford University	✓	×	✓	✓	✓	×	✓

Cuadro A.1.: Comparación entre diferentes solvers (tanto comerciales como de código abierto) en términos del dominio de aplicación. De izquierda a derecha, el tipo de problema al que es referido cada columna es: problemas de programación lineal (LP), programación con restricciones (CP), programación con enteros mixtos (MIP), programación lineal de mixtos-enteros (MILP), programación cuadrática cuadrática de mixtos-enteros (MIQP), programación cónica de segundo orden de mixtos-enteros (MISOCP), programación no-lineal (NLP). La tabla ha sido obtenida y ampliada a partir del análisis realizado en Anand et al. [2017].

A.2.1. Herramientas de optimización alternativas.

PuLP

PuLP es una biblioteca ideada en lenguaje Python, cuya razón de ser radica en proporcionar una alternativa simple y eficaz a la hora de describir y resolver problemas matemáticos de optimización. PuLP funciona completamente dentro de la sintaxis y expresiones naturales de Python (no requiriendo ningún otro paquete de software), al proporcionar objetos que representan problemas de optimización y variables de decisión, permitiendo expresar las distintas restricciones de cualquier problema de una manera muy similar a la expresión matemática original, lo cual es sin duda uno de los grandes atractivos de esta herramienta. Con el fin de mantener la sintaxis lo más simple e intuitiva posible, PuLP está orientado especialmente a resolver modelos lineales y de enteros mixtos. Por otra parte, PuLP es compatible con una gran variedad de solvers, tanto comerciales como de código abierto, lo que lo convierte en una excelente herramienta de apoyo a la hora de precisar usar más de un solver (Roy et al. [2009]).

Existen numerosos solvers de programación lineal y de enteros mixtos, tanto comerciales como de código abierto. PuLP adopta un enfoque modular para los solvers al manejar la conversión de expresiones Python-PuLP en números “sin procesar” (i.e., representaciones vectoriales y de matriz dispersa del modelo) internamente, para luego exponer estos datos a una clase de interfaz de solver, aprovechando el hecho de que la interfaz de muchos solvers es similar. A lo anterior merece la pena añadir que PuLP dispone de muy pocas funciones especiales o “palabras clave” —respetando así la sintaxis de Python—, y en general se proporcionan dos objetos principales (para un problema y para una variable), y luego se utilizan las estructuras de control y los operadores aritméticos de Python.

A diferencia de Pyomo, PuLP no permite a los usuarios crear modelos puramente abstractos, bajo la asunción de que la construcción de modelos abstractos no es necesaria para un gran número de enfoques en lenguajes modernos dinámicos y flexibles como Python. Estos lenguajes no distinguen entre datos o funciones hasta que se ejecuta el código, lo cual permite a los usuarios construir modelos complejos de forma pseudo-abstracta.

Documentación adicional sobre PuLP, así como vario ejemplos, puede hallarse en Mitchell et al. [2011] y en las referencias contenidas.

AMPL.

La optimización se ha convertido en un paradigma de vital importancia para una gran variedad de problemas en campos que van desde la ingeniería, hasta la física, la biología, la economía y la gestión empresarial. Es en este ámbito donde, más allá de Pyomo o PuLP, surgen herramientas comerciales especializadas como AMPL. Se trata de un lenguaje de programación algebraica (AML) empleado para describir y solucionar problemas de gran complejidad de programación matemática a grandes escalas (Robert Fourer, David Gay, Brian Kernighan [2003]).

Desarrollado en la década de lo 80' en los Laboratorios Bell, su sintaxis permite escribir problemas de optimización en un lenguaje declarativo basado directamente en la terminología algebraica. Las

A.2. DOMINIO DE APLICACIÓN DE DIFERENTES SOLVERS.

rutinas internas de esta herramientas permiten procesar modelos con números muy elevados de variables, integrando los modelos en hojas de cálculo y fuentes relacionales, ya sea localmente o en la nube.

Una de las ventajas que ofrece la interfaz estándar de AMPL es la de construir el modelo con independencia del solver escogido, permitiendo escoger a posteriori el que mejor se adapte a las necesidades del problema. Esto incluye desde programación lineal, hasta problemas lineales-cuadráticos, no lineales y globales (con un número masivo de parámetros). Todos recogidos en la tabla del **Anexo A.2** son compatibles con AMPL. Como valor añadido, destaca también su versatilidad al ser compatible con las consolas de lenguajes como Python, R, C++, C, MATLAB y Java, al mismo tiempo que mantiene la integridad y legibilidad del modelo.

El hecho de ser una herramienta de pago implica su constante desarrollo, mediante reporte y detección de errores y nuevas versiones, actualizándose con mayor rapidez que herramientas como Pyomo o PuLP. Por esta razón, así como la posibilidad que ofrece para definir restricciones sin necesidad de conocer las dimensiones del modelo y la familiaridad que presenta el uso de lenguaje algebraico, numerosas empresas de ingeniería y consultoría hacen uso de AMPL.

B. CÓDIGOS EMPLEADOS EN LENGUAJE PYTHON

- Estructura del programa.

La estructura del programa es la siguiente: en primer lugar, se importan las librerías que se van a usar a lo largo de toda la simulación. Primeramente, la librería *pandas* permite importar y modificar datos de un fichero, Por comodidad, importaremos la mayoría de nuestras variables desde un archivo Excel (“Datos_ejemplo_9_nodos.xlsx”). Por otro lado, la librería *numpy* permite utilizar una extensa gama de funciones y operaciones matemáticas, siendo quizás la librería con mayor presencia en cualquier código de Python. De cara resolver el problema de optimización, importamos *pyomo* o, directamente, algunos de sus paquetes como *SolverFactory*. Por último, *matplotlib* nos permite representar gráficamente cualquier resultado obtenido.

La disposición de datos de entrada para probar el modelo se basa en un grafo y matrices. El grafo base que se va a utilizar es el que se presenta habitualmente en artículos de esta línea de investigación (ver, por ejemplo, Laporte et al. [2011a]), orientado al estudio de carga de redes de transporte urbano: multimodal, público, privado, bus, metro, bici, etc. El Excel mencionado consta de varias hojas, donde se define la estructura del grafo: número de nodos, coste de la estación si un nodo es asignado a dicha estación, el tiempo de parada de los trenes en cada nodo, el tiempo de espera en los andenes (t_{wait}), designación del nodo si pertenece al conjunto origen, O , o bien al conjunto destino, D . El factor t es referido al valor del tiempo que se tarda en recorrer el arco $\{ij\}$ si éste existe, y un factor c para denotar el coste de construir un arco. Así pues, haciendo uso de *pandas*, se definen las variables df , dt , dc , dg y du para la lectura de nodos, tiempos de paradas, costes de construcción de tramos, valores O-D entre pares y valores de los tiempos en recorrer los arcos en modos alternativos, respectivamente.

A continuación, se procede a crear una lista con el conjunto de N nodos que conformarán el grafo de nuestro modelo. Para ello, se hace uso de la funciones *index* y *map*, con el fin de extraer datos de los *dataframes* existentes asignando índices, y ordenando dichos índices de cara a posteriores llamadas. Una vez definido N , se procede a definir los parámetros del modelo:

1. **Parámetros de nodos.** Para el conjunto de nodos, declaramos sus parámetros b , t y t_{wait} . En general, la definición de caso todas las variables la realizaremos mediante diccionarios.
2. **Parámetros de arcos.** Para el conjunto de arcos, creamos los parámetros t_{ij} (tiempo en un arco) y c (costes de construir tramos). El parámetro t_{ij} será simétrico. Se fija un valor C_{\max} para regular el espectro de simulación.
3. **Parámetros relacionados con la matriz origen-destino.** Se crean con este fin las matrices g y u . La demanda es a pares, y está representada por el conjunto $W \rightarrow N \mapsto N$. Cada par $w = (w^s, w^t)$ tendrá asociada una demanda g^w . Así, $G = g^w$ es la matriz origen-destino. Además, existe un modo de tránsito alternativo cuya utilidad viene dada por la matriz $U = u^w$, tal y como se describe en el Capítulo 3.

4. **Parámetros de sobreocupación.** Al ampliar el problema y considerar estos efectos, se introduce un valor fijo para la función de coste por sobreocupación, $P_{\text{máx}}$, la pendiente b (denotada como *slope* en el código, que tiene que ver con el coste de construcción de los nodos, y la el parámetro de ajuste M , denotado como *BigM* en el programa. Estos parámetros son necesarios para implementar las restricciones propuestas en el capítulo anterior.

Llegados a este punto, se procede a implementar el modelo en Pyomo. La sintaxis de esta herramienta requiere definir la creación del modelo explícitamente, que denominaremos *Model* y para lo cual usaremos la función *ConcreteModel*. A continuación, se definen los conjuntos de arcos A , el conjunto de aristas E y el conjunto de pares de demanda O-D W . La sintaxis de Pyomo permite definir estos conjuntos como funciones de Python, con la salvedad de que la salidas de estas funciones debe estar correctamente definida a lo largo de los N índices, evitando incoherencias como valores numéricos negativos para los arcos y definiendo correctamente la direccionalidad de las variables. La función *Set* de Pyomo permite inicializar los conjuntos definidos, pudiendo llamar a estos vinculándolos con el modelo definido (*Model*) y visualizando los resultados mediante la función *pprint*. Del mismo modo, se crean los conjuntos O y D para los orígenes y destinos, respectivamente, así como el conjunto L para aquellos nodos que no son origen ni destino, y el conjunto K para denotar al conjunto de aristas de E que inciden sobre el nodo k .

El siguiente paso consiste en crear las variables del modelo. Para ello, se declaran éstas detallando los índices sobre los que se definen, y su dominio. Recordemos que el problema propuesto considera cuatro tipos de variables:

1. **Variables de diseño:** y_i y x_i . Si se construye una estación en la posición i , $y_i = 1$. Si se construye un tramo en e , $x_e = 1$. x_i será una lista de variables binarias de dos índices, mientras que y_i es una lista de variables binarias de un índice.
2. **Variables de flujo:** f_a^w . Si el arco a captura la demanda w , $f[w, a] = f_a^w = 1$.
3. **Variables de captura:** z^w . Si la línea captura el valor de la demanda w , $z^w = 1$. z será una lista de variables binarias de dos índices.
4. **Variables de sobreocupación:** Pa , Oa , Sa y Haw , tal y como recoge el modelo propuesto.

Una vez definido todo lo anterior, estaremos en condiciones de crear la función objetivo. Para ello, se utiliza el comando *Objective* de Pyomo. Aquí podemos apreciar de primera mano la facilidad de uso que ofrece Pyomo, pudiendo expresar casi explícitamente la función objetivo (3.1) como: $expr = \text{sum}(g[w]*z[w] \text{ for } w \text{ in } W)$. Dentro del mismo comando, la consigna *maximize* permite indicar que buscamos maximizar esta función. Con la función objetivo escrita, todo está dispuesto para formular las restricciones. Cada restricción recibe su propia nomenclatura y consta de una sintaxis explícita similar a la de la función objetivo. Así pues, se definen las restricciones de presupuesto, restricciones de diseño (siete en total), restricciones de conservación de flujo, localización-asignación (tres tipos) y de elección de modos. Esta es sin duda la parte que presenta mayor complicación a la hora de escribir el programa, debido al elevado número de restricciones combinado con los bucles que presentan muchas de las funciones definidas. Las restricciones asociadas a la sobreocupación también se asocian posteriormente al modelo creado, de forma similar.

Finalmente, una vez construido el modelo se hace uso de *SolverFactory* y la función *solve* para resolver el problema. Es en este paso donde se especifica el solver utilizado, que en nuestro caso será

glpk. Las funciones *pprint* y *display* permiten visualizar en la consola de comandos los resultados obtenidos. Varias líneas adicionales acompañan al programa a lo largo de toda su elaboración, ordenando y especificando los resultados obtenidos a medida que se obtienen, de cara a facilitar su posterior análisis. Los grafos de la red obtenida se pueden representar mediante *matplotlib* o *NetworkX*.

```
# -*- coding: utf-8 -*-
"""
Created on Tue Nov 16 17:30:55 2021
@author: fepero
"""

# =====
# PRIMERO: MODELO MATEMATICO
# =====

# EN ESTE TRABAJO:
# 1. SE VA A CONSIDERAR EL PROBLEMA
# 2. DE LA LOCALIZACION
# 3. DEL TRAZADO
# 4. DE UNA LINEA
# 5. DE TRANSITO RAPIDO
# 6. QUE MAXIMIZE
# 7. EL NUMERO DE VIAJES CAPTADOS
# 8 TENIENDO EN CUENTA
# 9. LOS DISTINTOS EFECTOS
#10. DE LA SOBRECUPACION
# =====
# DATOS
# 1. Suponemos que para la linea de metro se va a escoger un camino en una red
# subyacente  $N=N(N,E)$ , es decir, que contiene las estaciones o paradas potenciales  $N$ 
# y las conexiones potenciales  $E$  entre estaciones.
# Para cada arista  $e=\{i,j\}$  se consideraran dos arcos:  $a=(i,j),=(j,i)$ .
# Podemos suponer que uno de los extremos de la linea esta el el conjunto  $O$  incluido en  $N$ 
# y que el otro en  $D$  incluido en  $N$ .
# Denotamos por  $\delta(k)$  al conjunto de aristas incidentes en el nodo  $k$ ,y  $\delta^-(k)$ 
# y  $\delta^+(k)$ el conjunto arcos entrantes y salientes de  $k$ , respectivamente.
# 2. La estructura de coste contiene el coste de construir el tramo  $c$ :  $c_e$  y
# el coste maximo  $C_{max}$ .
# Una alternativa consiste en acotar el numero de estaciones  $N_{max}$  y
# de tramos interestacion  $E_{max}$ .
# 3. La demanda es a pares y esta representada por el conjunto  $W$  incluido en  $N \times N$ .
# Cada par  $w=(w_s,w_t)$  incluido en  $W$  tiene asociada una demanda  $g_w$ .
# Asi  $G=(g_w)$  es la matriz orige-destino.
# Ademias existe un modelo de transito alternartivo
# cuya utilidad viene dada por la matriz  $U=(u_w)$ .
# 4. Denotamos por  $t_a$  el tiempo en el vehiculo, al recorrer el arco  $a$ ,
# por  $t_k$  el tiempo de parada en la estacion  $k$ ,
# y por  $t_{wait}$  el tiempo de espera en el anden.
```

APÉNDICE B. CÓDIGOS EMPLEADOS EN LENGUAJE PYTHON

```
# =====
# SEGUNDO: ENTORNO TENOLOGICO
# =====
# a. CODIGO EN PYOMO EN EL EDITOR SPYDER 4.2.3 BAJO LA PLATAFORMA
# ANACONDA NAVIGATOR 2.1.0
# HAY QUE INSTALAR TODAS LAS LIBRERIAS en el terminal dentro de anaconda
# en el enviroment de spyder 4.2.3f
# b. Los ficheros deben estar en el mismo sitio que se ha hecho a instalacion
# para ahorrar los paths en la terminal
# c. LIBRERIAS UTILIZADAS#
#import numpy as np
import pandas
import pyomo.environ as pe
# from pyomo.opt import SolverFactory
# from pyomo.environ import *
#import matplotlib.pyplot as plt

#=====
# TERCERO: ENTRADA DE DATOS: estructura de informacion que recibe el modelo en pyomo
#=====
# En EXCEL. En este caso el fichero de datos es 'Datos_ejemplo_9_nodos.xlsx'.
# Dentro del mismo hay cinco hojas:
# PPRIMERA HOJA: numero de nodo, el coste de la estacion si el nodo es asignado estacion,
# el tiempo de parada del tren en el nodo si es estacion elegida, twait tiempo de espera
# en el anden, 0 si es posible nodo del conjunto Origen
# y D si es posible nodo del conjunto destino.
# SEGUNDA HOJA: valor del tiempo que se tarda en recorrer el arco i j si existe,
# -1 si no existe.
# TERCERA HOJA: coste de construir la arista si existe, -1 si no existe.
# CUARTA HOJA: matriz g entre pares de centroides (asociados a todos los posibles nodos)
# pertenecientes al conjunto origen y al conjunto destino.
# QUINTA HOJA: para cada par origen destino tiempo que se tarda en ir
# por un modo de transporte alternativo.
#
# =====
# # Origen de los datos numericos
# =====

# El valor de los datos numericos que se han utilizado se han obtenido del articulo
# 'Design robust transit networks with alternative routes', de G. Laporte, A. Marin,
# J.A Mesa y F. Perea.

#=====#
# Introducimos la los datos de la hoja de calculo en pyomo con la libreria
# Pandas de python.
# =====
# Un DataFrame es una estructura de datos con dos dimensiones en la cual se
# puede guardar datos de distintos tipos
```

```

# (como caracteres, enteros, valores de punto flotante, factores y ms) en columnas.
# Es similar a una hoja de clculo o una tabla de SQL o el data.frame de R.
# Un DataFrame siempre tiene un ndice (con inicio en 0).
# El indice se refiere a la posicion de un elemento en la estructura de datos.
print('Leyendo datos')
#los nodos en df
df = pandas.read_excel('Datos_ejemplo_9_nodos.xlsx', sheet_name='nodos',
header = 0, index_col=[0])
#print(df)
# Tiempos de parada en dt:
dt = pandas.read_excel('Datos_ejemplo_9_nodos.xlsx', sheet_name='t',
header = 0, index_col=[0])
#print(dt)
# Costes de construir los tramos en dc:
dc = pandas.read_excel('Datos_ejemplo_9_nodos.xlsx', sheet_name='c',
header = 0, index_col=[0])
#print(dc)
# Valores ORIGEN-DESTINO entre pares en dg:
dg = pandas.read_excel('Datos_ejemplo_9_nodos.xlsx', sheet_name='g',
header = 0, index_col=[0])
#print(dg)
# Valores de los tiempos en recorrer los arcos en modos alternativos en du:
du = pandas.read_excel('Datos_ejemplo_9_nodos.xlsx', sheet_name='u',
header = 0, index_col=[0])
#print(du)
# =====
# CUATRO generamos el conjunto de nodos N del grafo.
#
# =====
# Para ello utilizamos las funciones siguientes
# La index function se utiliza para extraer del data frame datos mediante indices.
# La funcion map() se usa para mapear los valores de series de acuerdo
# con el imput correspondiente.
#Index.map(mapper, na_action=None)[source]
#Ahora generamos el conjunto de nodos.
# leemos la primera columna de df:
N = list(df.index.map(int))
#print(N)
#
#=====
# QUINTO: PARAMETROS
# =====
#
# Pyomo permite al usuario formular problemas de optimizacion en Python,
# en una manera similar a la notacion usual en optimizacion matematica.
# Admite un estilo orientado a objetos para formular modelos de optimizacion,
# los cuales son definidos a travs de gran variedad de componentes: conjuntos,
# parametros, variables, objetivos, restricciones. las fuentes externas de
# datos pueden ser definidas usando hojas de clculo. Pyomo admite a los
# modelos abstractos, los cuales son definidos sin datos, y los concretos,
# definidos con datos. En ambos casos, permite la separacion del modelo y los datos

```

APÉNDICE B. CÓDIGOS EMPLEADOS EN LENGUAJE PYTHON

```
# Pyomo solvers comerciales y de código abierto, incluyendo muchos Gurobi y GLPK.
# De ahora en adelante, usaremos el término parámetros para referirnos a cualquier
# componente que tome un valor definido y fijo y variables a aquellos componentes
# cuyo valor depender de las decisiones que tomemos.
# =====
# 1. Parámetros de nodos
# Para el conjunto de nodos declaramos sus parámetros b, t y t_wait.
# =====
# O, D accediendo al data frame selecciona para cada nodo i la columna b
# b es el nombre signado a un diccionario que va entre llaves para
# cada i en N, el propio valor de i, seguido dos puntos :, y el elemento
# de la fila i y columna b en la estructura de datos df. b es un diccionario
# que es una lista, porque asigna mediante dos puntos una traducción para cada
# i t de la columna b. Y lo hace con "AT"
b = {(i): df.at[i,'b'] for i in N}
#print(b)
# Otra tabla diccionario:
t_i = {(i): df.at[i,'t_i'] for i in N}
#print(t_i)
#t_wait = {(i): df.at[i,'t_wait'] for i in N}
t_wait = 1
#print(t_wait)
O_1 = {(i):df.at[i,'O_1'] for i in N}
#print(O_1)
D_1 = {(i):df.at[i,'D_1'] for i in N}
#print(D_1)
# Ya están metidos todos los datos del excel de los nodos
# =====
#2. Parámetros de arcos
#para el conjunto de arcos creamos los parámetros tij y c
# =====
#Ahora declararemos los parámetros relacionados con los arcos.
#Primero declaramos el parámetro t, solo cuando el valor correspondiente no sea -1.
# En este caso el parámetro t es simétrico.
# Así como lo de los arcos eran diccionarios listas, ahora son matrices.
t_ij = {(i,j):max(dt.at[i,j],dt.at[j,i]) for i in N for j in N
if max(dt.at[i,j],dt.at[j,i])>0}
#print(t_ij)
# Costes de construir tramos:
c = {(i,j):max(dc.at[i,j],dc.at[j,i]) for i in N for j in N
if max(dc.at[i,j],dc.at[j,i])>0}
#print(c)
Cmax=50
#=====
# 3. parámetros relacionados con la matriz OD.
# Para ello creamos g las matrices g y u.
# La demanda es a pares y esta representada por el conjunto W N N. Cada par
# w = (ws, wt) 2 W tiene asociada una demanda gw. Así G = (gw) es la matriz
# origen-destino. Además, existe un modo de tránsito alternativo cuya utilidad
# viene dada por la matriz U = (uw).
# g son los términos de la matriz origen destino, que se han cargado con pandas
```

```

# en la estructura dg. Cuando dice dg.at[i,j], dice que el dato esta "AT" en ingles
# de la posicon i,j de la tabla, y lo asigna a una pareja (i,j). Los almacena en
# Crearemos ahora los parametros relacionados con los pares OD.
# Primero, declaramos el parametro g, solo cuando el correspondiente valor no sea 0.
g = {(i,j):dg.at[i,j] for i in N for j in N if dg.at[i,j] > 0}
#print (g)
# Similar para du, con los tiempos en modos alternativos
u= {(i,j):du.at[i,j] for i in N for j in N if du.at[i,j] > 0}
#print (u)

#=====
# #Parmetros para el efecto de la sobreocupacion.
# P es la carga de un arco para que exista sobreocupacion.
# En las nuevas restricciones, slope se corresponde con el parametro b,
# que es el factor hace aumentar
# el tiempo de viaje en un arco por cada pasajero extra cuando se produzca sobreocupacion
# (la pendiente de la recta).
# BigM es M, un parametro muy grande.
Pmax = 200
slope = 0.1
BigM = 10000
#=====
# #SEXTO CREAMOS EL MODELO EN PYOMO
#=====
# =====
#
#
# =====
print('Fin leyendo declarando datos. Comienzo declaracin conjuntos')
Model = pe.ConcreteModel()

# =====
# SEPTIMO CREAMOS LOS CONJUNTOS DE ARCOS A, TRAMOS ENTRE ESTACIONES E Y DEMANDA W.
# =====
# El conjunto N ya esta creado.
# AHORA DEFINIMOS LOS CONJUNTOS QUE TIENE NUESTRO MODELO
#CON UNA REGLA (RULE).
# Vamos a definir cada conjunto con su indice.
# Para los arcos la regla para crear el conjunto de arcos A
# como [12,13,23,24,etc] consiste en del dataframe dt, leer los elementos de la matriz
# aij y aji mayor que cero. Distinto de -1 es que no existe.
# CUIDADO CON LA INDEXACION DE LAS LNEAS, en el def, return, model tenemos que dejar
# unos espacios antes de comenzar la instruccion
# Creamos el conjunto de arcos:
def A_rule(Model):
    return set([(i,j) for i in N for j in N if max(dt.at[i,j],dt.at[j,i]) > 0])
Model.A = pe.Set(initialize= A_rule, dimen=2)
A = Model.A
#A.pprint()
# Creamos el conjunto de aristas.
def E_rule(Model):

```

APÉNDICE B. CÓDIGOS EMPLEADOS EN LENGUAJE PYTHON

```

    return set([(i,j) for i in N for j in N if max(dt.at[i,j],dt.at[j,i]) > 0 and i < j])
Model.E = pe.Set(initialize= E_rule, dimen=2)
E = Model.E
#E.pprint()
# W son pares de demanda OD:
def W_rule(Model):
    return set([(i,j) for i in N for j in N if dg.at[i,j] > 0])
Model.W = pe.Set(initialize = W_rule, dimen=2)
W = Model.W
W.pprint()
# Creamos el conjunto O:
def O_rule(Model):
    return set([(i) for i in N if O_1[i] == 1])
Model.O = pe.Set(initialize = O_rule, dimen=1)
O = Model.O
#O.pprint()
# Creamos el conjunto destino, D:
def D_rule(Model):
    return set([(i) for i in N if D_1[i] == 1])
Model.D = pe.Set(initialize = D_rule, dimen=1)
D = Model.D
#D.pprint()
# Creamos el conjunto L de nodos que no son origen ni destino:
def L_rule(Model):
    return set([(i) for i in N if (O_1[i] == 0) and (D_1[i] == 0)])
Model.L = pe.Set(initialize = L_rule, dimen=1)
L = Model.L
#L.pprint()
# K(k)es el conjunto de aristas de E que inciden en el nodo k.
# def K_rule (Model):
#     for i in N:
#         O_1 = {(i):(i,j) for j in N}
#         return seti:(i,j) e if i==4
# Inicializamos el conjunto;
# Model.K = Set(initialize=K_rule, dimen=2)
# K = Model.K
# K.pprint()
# =====
# OCTAVO CREAMOS LAS VARIABLES DEL MODELO
# Comenzamos con la declaracin de variables del modelo
# Hay que detallar los ndices sobre os que se definen, y su dominio.
# =====
# 1.VARIABLES DE DISEO yi xe. Si se construye estacion en i, yi vale 1.
# Si se construye tramo en e, xe es 1.
# 2.VARIABLES DE FLUJO fa^w si el arco a captura la demanda w faw vale 1.
# 3.VARIABLES DE CATPURA z^w si la lineas captura el valor de la demanda w z^w vale uno
# las variables se representan con su tantos conjuntos haya como indices y el tipo
# y es una lista de variables binarias de un indice
print('Fin declaracin conjuntos. Declaro variables')
Model.y = pe.Var(N, within=pe.Binary)
# Para que sea ms comodo y no tener que ir arrastrando "Model.y" lo renombramos como y

```

```

y = Model.y
# x es una lista de variables binarias de dos indices:
Model.x = pe.Var(E, within=pe.Binary)
x = Model.x
#f[w,a] = 1, si la demanda w atraviesa el arco a, 0 en caso contrario.
# 4 Indices:
Model.f = pe.Var(A, W, within=pe.Binary)
f = Model.f
#z es una lista de variables binarias
Model.z = pe.Var(W, within=pe.Binary)
z = Model.z
# 5. Las nuevas VARIABLES para el efecto de SOBRECUPACION:
Model.Pa = pe.Var(A,within=pe.PositiveReals)
Pa = Model.Pa
Model.Oa = pe.Var(A,within=pe.Binary)
Oa = Model.Oa
Model.Sa = pe.Var(A,within=pe.PositiveReals)
Sa = Model.Sa
Model.Haw = pe.Var(A,W,within=pe.PositiveReals)
Haw = Model.Haw

# Fin nuevas variables para sobreocupacin.

# =====
# # #
# # #El presupuesto maximo es un parmetro que hay que declarar.

#
# =====
# NUEVE: CREAMOS LA FUNCION OBJETIVO.
# =====
#
print('Fin declaracin variables. Declaro restricciones y fo')
Model.obj = pe.Objective(expr = sum(g[w]*z[w] for w in W), sense=pe.maximize)
#
# =====
# #DIEZ: RESTRICCIONES. Hoja de calculo. restricciones_tfm_atx.xls
# =====
# =====
# 1. Restricciones de presupuesto.
# =====
#
Model.Presupuesto = pe.Constraint(expr = sum(c[e]*x[e] for e in E) +
sum(b[i]*y[i] for i in N) <= Cmax)
# #Model.Presupuesto.pprint()
# # =====
# # 2. Restricciones de Diseo.
# # =====
# # =====
# #
# # Tipo 2.1
# # =====

```

APÉNDICE B. CÓDIGOS EMPLEADOS EN LENGUAJE PYTHON

```

# # Las restricciones Diseo 1 se definen para cada e en E, y para cada i en e.
# # Una forma (no la nica) es la siguiente:
Model.Disenno1 = pe.ConstraintList()
for e in E:
    for i in e:
        Model.Disenno1.add(expr=x[e] <= y[i])
# #Model.Disenno1.pprint()
# # =====
# #
# #     Tipo 2. 2a
# #
# # =====
# Diseo 2a
Model.Disenno2a = pe.Constraint(expr=sum(y[i] for i in O) == 1)
#Model.Disenno2a.pprint()
# # =====
# #     Tipo 2. 2b
# # =====
# # Diseo 2b
Model.Disenno2b = pe.Constraint(expr=sum(y[i] for i in D) == 1)
#Model.Disenno2b.pprint()
# # =====
# #     Tipo 2. 2c
# # =====
# # Diseo 2c
Model.Disenno2c = pe.Constraint(expr=sum(x[e] for e in E) + 1- sum (y[i] for i in N) == 0)
#Model.Disenno2c.pprint()
# =====
#     Tipo 2. 2d
# =====
# Diseo 2d
Model.Disenno2d = pe.ConstraintList()
#para las aristas delta(kincidentes en el nodo de L
for k in L:
    Model.Disenno2d.add(sum(x[e] for e in E if k in e) <= 2*y[k])
# Model.Disenno2d.pprint()
# # =====
# #     Tipo 2 2e
# # =====
# Diseo 2e
#Model.Disenno2e = pe.ConstraintList()
# for k in O:
#     Model.Disenno2e.add(sum(x[e] for e in E if k in e) == 1)
Model.Disenno2e = pe.Constraint(expr= sum(sum(x[e] for e in E if k in e)
for k in O) == 1)
#Model.Disenno2e.pprint()
# #=====
# #     Tipo 2 2f
# # =====
# ##Diseo 2f
# Model.Disenno2f = pe.ConstraintList()

```

```

# for k in D:
#     Model.Disenno2f.add(sum(x[e] for e in E if k in e) == 1)
Model.Disenno2f = pe.Constraint(expr= sum(sum(x[e] for e in E if k in e)
for k in D) == 1)
#Model.Disenno2f.pprint()
# =====
#     Restriccion 3 de Conservacion de flujo
# =====
# Intervienen el conjunto W de pares origen destino ws wt,
# el conjunto de arcos de A que pasan por el nodo i entrando o saliendo
# y forman parte del camino minimo entre s y t, deltaw(k)
# entrante si es postivo o saliente si es negativo
# Variables de flujo: faw = 1, si la demanda w atraviesa el arco a, 0 en caso contrario.
# Variables de captura: zw si la linea captura el par de demanda w,
# 0 en caso contrario.
# Vamos a hacerlo en tres pasos:
# Flujo de salida desde cada origen igual a la demanda del par origen-destino.
# =====
# QUE ES LA LINEA? UN CAMINO DE LA RED (N, E) CON UN EXTREMO EN O Y OTRO EN D
# POR EJEMPLO EL CAMINO ENTRE 2 Y 8 (s, t)
# Por ese camino pasara la demanda de distintos pares OD (que son ws, wt y
# valen gst, o sea el valor de la matriz od)?
# Los pares od deberian estar entre centroides o y d. Pero en este modelo los
# centroides se CONECTAN directamente a los nodos mas cercanos.
# Por supongamos la linea es entre el centroide plaza de cuba y el
# centroide Mairena. La pregunta es que arcos (i,j) de la red que
# unen los nodos que coinciden con los centroides plaza de cuba (s)
# y mairena(t) forman parte del camino st y por lo tanto tienen
# flujo fij pero asociado a st. Por eso f tiene cuatro indices:
# porque viene de una demanda PAR concreta que CAPTA el arco que forma parte
# del camino y vale 1, o vale cero si no lo capta..
#Y estas restricciones lo que hacen es un balance en un nodo de la red con
# todos los arcos que le entran y le salen.
# Y cargados con la f que vale 0 o 1 segun lo anterior .
# =====
# EL MODELO MATEMATICO DICE QUE:
# Para cada Nodo en N hay que hacer la conservacin de flujo. Variamos i de 1 a N
# Despus recorremos todos los pares wst que son trminos de la matriz de origen destino
# distintos de 0, o sea el conjunto W. La variable f de 4 indices del nodo i, del nodo k
# que forma con el nodo i un arco existente ik, o sea del conjunto A,
# para s y t del wst de la matriz)
# f( k,i,ws,wt) para k que forme arco existente ki (el opuesto de la arista).
# =====
Model.Flujo = pe.ConstraintList()
for i in N:
    for (ws,wt) in W:
        if i == ws:
            Model.Flujo.add(sum(f[i,k,ws,wt] for k in N if (i,k) in A)-sum(f[k,i,ws,wt]
for k in N if (k,i) in A) == z[ws,wt])
        elif i == wt:
            Model.Flujo.add(sum(f[i,k,ws,wt] for k in N if (i,k) in A)-sum(f[k,i,ws,wt]

```

APÉNDICE B. CÓDIGOS EMPLEADOS EN LENGUAJE PYTHON

```

        for k in N if (k,i) in A == -z[ws,wt])
    else:
        Model.Flujo.add(sum(f[i,k,ws,wt] for k in N if (i,k) in A)-sum(f[k,i,ws,wt]
            for k in N if (k,i) in A) == 0)
#Model.Flujo.pprint()
# =====
# Restricciones 4 Localizacion-asignacion.
# =====
# Dado un par w, el flujo en un arco que forma parte del camino entre s y t mas
# el de su complementario tiene que ser 1 si pertenece a la arista
# correspondiente que ha salido elegida como parte del camino entre ws y wt.
# Como en el modelo hemos dicho la demanda que capta un tramo pasa en un
# sentido o pasa en otro entre estaciones ganadoras o hay una de i a j o de
# j a i.
# =====
Model.Locasa = pe.ConstraintList()
for w in W:
    for (i,j) in E:
        Model.Locasa.add(f[i,j,w] + f[j,i,w] <= x[i,j])
#Model.Locasa.pprint()
#=====
#Tipo 4b
# Cada nodo ws que coincida con el centroide origen enchufado a el por ser el mas cercano
# tiene flujo del arco saliente.
# =====
Model.Locasb = pe.ConstraintList()
# for i in N:
#     for (ws,wt) in W:
#         if i == ws:
#             Model.Locasb.add(sum(f[i,k,ws,wt] for k in N if (i,k) in A) == y[i])
# Versin Fede:
for (ws,wt) in W:
    Model.Locasb.add(sum(f[ws,k,ws,wt] for k in N if (ws,k) in A) <= y[ws])
#Model.Locasb.pprint()
#=====
#Tipo 4c
# Cada nodo que coincida con el centroide destino enchufado a el por ser el mas cercano
# tiene flujo que sale por sus arco saliente correspondiente al par s t de donde dicho arco ha
# sido elegido como parte de la ruta minima.
#=====
Model.Locasc = pe.ConstraintList()
# for j in N:
#     for (ws,wt) in W:
#         if j == wt:
#             Model.Locasc.add(sum(f[k,j,ws,wt] for k in N if (k,j) in A) == y[j])
# Versin Fede
for (ws,wt) in W:
    Model.Locasc.add(sum(f[j,wt,ws,wt] for j in N if (j,wt) in A) <= y[wt])
#Model.Locasc.pprint()
#=====
# Tipo 4d.

```

```

# Si hay flujo en un nodo k que es origen o destino de demanda la linea o camino elegido entre
# el nodo k y su extremo capta la demanda de dicho par
# Definimos el conjunto ws,wt.
#=====
Model.Locasd = pe.ConstraintList()
# for k in N:
#     for (ws,wt) in W:
#         if k == ws or k == wt:
#             Model.Locasd.add(expr = z[ws,wt] <= y[k])
# Versin Fede:
for (ws,wt) in W:
    Model.Locasd.add(z[ws,wt] <= y[ws])
    Model.Locasd.add(z[ws,wt] <= y[wt])
#Model.Locasd.pprint()
# =====
#     Restricciones 5 Eleccion del modo
# =====
# Intervienen para cada par y con el modo u tantas restricciones como combinaciones.
# La variable flujo sobre el conjunto A y W, el parametro ta sobre el conjunto L
# El parametro t a en el conjunto A, el parametro t en la estacion sobre
# conjunto de nodos intermedios L, f sobre L y sobre arcos incidentes a nodos de
# L, el tiempo de espera por par en cada linea.
Model.Modo = pe.ConstraintList()
# for j in N:
#     for w in W:
#         Model.Modo.add(sum(t_ij[k,j]*f[k,j,w] for k in N if (k,j) in A)
+ sum(t_i[k]*sum(f[k,j,w] for k in L if (k,j) in A) +
t_wait[k]*z[w] for k in L if (k,j) in A) <= u[w])
# Versin Fede.
for w in W:
#     for k in N:
# Considerando efecto de sobreocupacin:
#         Model.Modo.add(sum(t_ij[a]*f[a,w] for a in A) +
sum(t_i[k]*(sum(f[k,j,w] for j in N if (k,j) in A)) for k in L) + t_wait*z[w]<=u[w])
# Considerando efecto de sobreocupacin estas:
#         Model.Modo.add(sum(Haw[a,w] for a in A) +
sum(t_i[k]*(sum(f[k,j,w] for j in N if (k,j) in A)) for k in L) + t_wait*z[w]<=u[w])

# #Model.Modo.pprint()

# Aado las nuevas restricciones para la sobreocupacion.
Model.DefPa = pe.ConstraintList()
Model.Def01 = pe.ConstraintList()
Model.Def02 = pe.ConstraintList()
Model.DefSa1 = pe.ConstraintList()
Model.DefSa2 = pe.ConstraintList()
for a in A:
    Model.DefPa.add(Pa[a] == sum(g[w]*f[a,w] for w in W))
    Model.Def01.add(Pa[a] - Pmax - BigM*0a[a] <= 0)
    Model.Def02.add(Pa[a] - Pmax + BigM*(1-0a[a]) >= 0)

```

APÉNDICE B. CÓDIGOS EMPLEADOS EN LENGUAJE PYTHON

```
Model.DefSa1.add(Sa[a] >= t_ij[a])
Model.DefSa2.add(Sa[a] >= t_ij[a] + slope*(Pa[a] - Pmax) - BigM*(1-Oa[a]))

Model.DefH1 = pe.ConstraintList()
Model.DefH2 = pe.ConstraintList()
Model.DefH3 = pe.ConstraintList()
for a in A:
    for w in W:
        Model.DefH1.add(Haw[a,w] <= BigM*f[a,w])
        Model.DefH2.add(Haw[a,w] >= Sa[a]- BigM*(1-f[a,w]))
        Model.DefH3.add(Haw[a,w] <= Sa[a])

# Fin nuevas restricciones al considerar efecto sobreocupacin.

# RESOLUCION DEL MODELO.
print('Fin declaracion restricciones. Resuelvo el modelo')
opt = pe.SolverFactory('glpk')
opt.solve(Model, tee=True)
#results = opt.solve(Model,tee=True)
#Model.pprint()
#Model.display()
# print value of variables
# print objective value
#print(value(Model.obj))
#Model.obj.display()
#Model.x.display()
#Model.y.display()
#print(value(Model.obj))

print('MOSTRAR RESULTADOS')
# Primero pongo el valor objetivo:
print('Cobertura Mxima', pe.value(Model.obj))
# Ahora saco por pantalla las estaciones a construir:
for i in N:
    if pe.value(y[i])==1:
        print('Estacin en nodo',i)
# Ahora saco por pantalla las aristas a construir :
for e in E:
    if pe.value(x[e]) == 1:
        print('Enlace en arista',e, 'con carga', pe.value(Pa[e[0],e[1]]), 'y',
              pe.value(Pa[e[1],e[0]]))
        if pe.value(Oa[e[0],e[1]])==1:
            print('Arco saturado')
        else:
            print('Arco no saturado')
        if pe.value(Oa[e[1],e[0]])==1:
            print('Arco inverso saturado')
        else:
            print('Arco inverso no saturado')
```

```

# Saco por pantalla los pares cubiertos.
# Para cada uno de ellos la ruta que seguirian:
for w in W:
    if pe.value(z[w])==1:
        print('Par',w,'captado. Su ruta es')
        parar = 0
        i = w[0]
        while parar == 0:
            for j in N:
                if (i,j) in A and pe.value(f[i,j,w]) ==1:
                    print('De',i,'a',j)
                    i = j
                    if j == w[1]:
                        parar = 1
                        break
Visitados = {(i):0 for i in N}
for i in 0:
    if pe.value(y[i]==1):
        print('La linea comienza en',i)
        comienzo = i
        Visitados[i]=1
parar = 0
while parar == 0:
    parar = 1
    for i in N:
        if Visitados[i] == 0 and (comienzo,i) in E:
            if pe.value(x[comienzo,i])==1:
                print('La linea continua en',i)
                Visitados[i]=1
                comienzo = i
                parar = 0
        if Visitados[i] == 0 and (i,comienzo) in E:
            if pe.value(x[i,comienzo])==1:
                print('La linea continua en',i)
                Visitados[i]=1
                comienzo = i
                parar = 0
print('Cobertura Mxima', pe.value(Model.obj))
print('Coste', sum(c[e]*pe.value(x[e]) for e in E) + sum(b[i]*pe.value(y[i]) for i in N))

```
