

Trabajo de Fin de Grado

Ingeniería en Tecnologías Industriales

Diseño e implementación de un modelo de optimización para el control de aglomeraciones en las estaciones de una línea de metro

Autor: Pedro Morán Cerdón

Tutor: José David Canca Ortiz

**Dpto.
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla**

Sevilla, 2021



Trabajo de Fin de Grado
Ingeniería en Tecnologías Industriales

Diseño e implementación de un modelo de optimización para el control de aglomeraciones en las estaciones de una línea de metro

Autor:

Pedro Morán Cordon

Tutor:

José David Canca Ortiz

Catedrático de Universidad

Dpto. de Organización Industrial y Gestión de Empresas I

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2021

Proyecto Fin de Carrera: Diseño e implementación de un modelo de optimización para el control de aglomeraciones en las estaciones de una línea de metro

Autor: Pedro Morán Cordón

Tutor: José David Canca Ortiz

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2021

El secretario del Tribunal

A mi familia

Agradecimientos

Durante toda mi etapa universitaria he tenido la suerte de compartir clases, horas de estudio, exámenes con compañeros increíbles. Dar las gracias a todos ellos por su ayuda de forma desinteresada, sus ánimos cuando las cosas no salían todo lo bien que nos gustaría y los momentos vividos.

Quiero dar las gracias a todos los profesores que me han ayudado durante todos estos años. Además de aprender conocimientos que me ayudarán, y me están ayudando, en mi carrera profesional también me llevo muchas experiencias que me han hecho crecer personalmente. En especial, me gustaría dar las gracias a mi tutor para este trabajo. Agradezco enormemente su ayuda, implicación y disponibilidad durante todo el proyecto.

Por último y más importante, a mis padres. La razón última por la que me quedaba una hora más en la biblioteca, me repasaba una última vez un ejercicio... eran ellos. Todo lo que he conseguido y conseguiré se lo debo a ellos.

Pedro Morán Córdón

Grado de Ingeniería en Tecnologías Industriales.

Sevilla, 2022

Resumen

La optimización de recursos a través de métodos matemáticos ha cobrado una gran importancia en los últimos años. Estas técnicas de optimización tienen numerosas y muy distintas aplicaciones. Entre otras, grandes empresas como Inditex basan toda su logística y organización en alguno de estos métodos. (ABC, 2018)

En nuestro caso usaremos estas técnicas de optimización para tratar otro servicio cuya importancia debe aumentar en los próximos años: el transporte urbano y/o metropolitano por ferrocarril. En las próximas páginas se detallará un modelo matemático que aspira a dar solución, al menos parcialmente, al problema de la congestión en las estaciones de metro de grandes ciudades.

En primer lugar, se realizará una introducción en la que se explica por qué el tren es un modo de transporte que no debería parar de crecer. Más adelante, se explicará el enfoque seguido y el método que se usará para paliar la congestión en las estaciones: la reserva de vagones. También se mostrarán los datos que resultan necesarios para nuestro caso de estudio, así como la explicación detrás de ellos.

Para desarrollar el modelo propuesto, se han utilizado EXCEL, Python y GUROBI. Excel se utiliza tanto para recoger los datos como para plasmar los resultados. Python es el lenguaje en el que se han desarrollado las restricciones y la función objetivo del modelo. Por último, GUROBI es un programa de optimización lineal que nos ha permitido resolver el modelo, llevar a cabo los experimentos y estudiar distintas situaciones de una forma rápida, a pesar de manejar una gran cantidad de datos.

Una vez seleccionados los datos de los diferentes experimentos, se resolverán varios escenarios para comprobar la efectividad del método. En concreto, comprobaremos cómo afecta al flujo de pasajeros la aplicación de políticas de reserva de plazas. Descifrar el comportamiento del modelo será clave para discernir su efectividad y concluir si realmente la reserva de vagones puede ser una solución adecuada para mejorar el flujo de las estaciones de líneas ferroviarias de alto volumen de pasajeros.

Índice

Agradecimientos	9
Resumen	11
Índice	13
1 Índice de Tablas	15
2 Índice de Gráficas	17
3 Índice de Figuras	19
4 Objetivos	1
5 Introducción	3
6 Revisión de la Literatura	5
7 Descripción del modelo	11
<i>Hipótesis</i>	<i>14</i>
7.1.1 Hipótesis 1	14
7.1.2 Hipótesis 2	14
7.1.3 Hipótesis 3.	15
<i>Datos requeridos</i>	<i>15</i>
7.1.4 Restricciones	18
7.1.5 Función Objetivo	24
7.1.6 No linealidades	25
<i>Formulación del modelo usando notación simbólica</i>	<i>28</i>
8 Implementación en Python y Gurobi	33
9 Caso de estudio	43
<i>Descripción y explicación de datos</i>	<i>44</i>
10 Experimentación	49
<i>Demanda</i>	<i>49</i>
<i>Definición de experimentos</i>	<i>51</i>
10.1.1 Primera experimentación	51
10.1.2 Segunda experimentación	54
10.1.3 Tercera experimentación	57
10.1.4 Cuarta experimentación	59

11 Conclusiones**63****12 Bibliografía****65**

1 ÍNDICE DE TABLAS

Tabla 1: np, dp, op, tp representados en Excel.....	16
Tabla 2: Demanda en los primeros 5 minutos.....	16
Tabla 3: Horario del servicio.....	34
Tabla 4:Ejemplo de horarios.....	43
Tabla 5: Ejemplo de demanda.....	44
Tabla 6:Grupos de pasajeros p.....	45
Tabla 7:Horario.....	45
Tabla 8:Datos prefijados.....	46
Tabla 9: Datos iniciales para la 1ª experimentación.....	51
Tabla 10:Riesgo de aglomeración en 1ª experimentación.....	52
Tabla 11:Flujo de pasajeros en 1ª experimentación.....	53
Tabla 12:Datos iniciales en 2ª experimentación.....	54
Tabla 13: Vagones disponibles en 2ª experimentación.....	55
Tabla 14:Comparación de flujo de pasajeros con factor de escala 0.6.....	56
Tabla 15:Flujo de pasajeros y capacidad restante para factor de escala 0.6.....	56
Tabla 16:Datos iniciales para 3ª experimentación.....	57
Tabla 17:Flujo de pasajeros en 3ª experimentación.....	58
Tabla 18:Riesgo de aglomeración en 3ª experimentación.....	59
Tabla 19:Datos iniciales para 4ª experimentación.....	59
Tabla 20:Vagones disponibles en 4ª experimentación.....	60
Tabla 21:Flujo de pasajeros en 4ª experimentación.....	60
Tabla 22:Flujo de pasajeros en estación nª3 con factor de escala 0.5.....	60

2 ÍNDICE DE GRÁFICAS

Ilustración 1: 2ª suposición	14
Gráfico 2:Ejemplo de riesgo de aglomeración	23
Gráfico 3: Riesgo de aglomeración en función de pasajeros.....	27
Gráfico 4:Riesgo de aglomeración en 1ª experimentación.....	52
Gráfico 5 :Flujo de pasajeros en 1ª experimentación	53
Gráfico 6:Máximo número de pasajeros por estación para factor de escala 0.6	55
Gráfico 7:Demanda en estación 4 para factor de escala 0.6.....	57
Gráfico 8:Flujo de pasajeros en 3ª experimentación	58

3 ÍNDICE DE FIGURAS

Ilustración 1. Aumento de población en áreas urbanas (Fuente: UN World Urbanization Prospects).....	3
Ilustración 2: Zona de espera de pasajeros.....	12
Ilustración 3: Pasajeros esperando el primer tren.....	12
Ilustración 4: Pasajeros a bordo del tren sin reserva de vagones.....	13
Ilustración 5: Ejemplo con reserva de vagones.....	13
Ilustración 6: Librerías usadas en Python.....	33
Ilustración 7: IOFunctionsExcel en Python.....	33
Ilustración 8: Uso de IOFunctionsExcel.....	34
Ilustración 9: Variables declaradas en Python.....	35
Ilustración 10: Restricciones 1 en Python.....	35
Ilustración 11: Restricciones 2 en Python.....	35
Ilustración 12: Restricciones 3 en Python.....	36
Ilustración 13: Restricciones 5 en Python.....	36
Ilustración 14: Relación entre i, k, s y i, k en Python.....	37
Ilustración 15: Restricciones 6 en Python.....	37
Ilustración 16: Restricciones 7 en Python.....	38
Ilustración 17: Restricciones 8 en Python.....	38
Ilustración 18: Restricciones 9 en Python.....	39
Ilustración 19: Restricciones 10 en Python.....	39
Ilustración 20: Restricciones 11 en Python.....	39
Ilustración 21: Restricciones 12 en Python.....	40
Ilustración 22: Restricciones 13 en Python.....	40
Ilustración 23: Variables previas en Restricciones 14 en Python.....	41
Ilustración 24: Restricciones 14 en Python.....	41
Ilustración 25: Restricciones 15 y 16 en Python.....	42
Ilustración 26: Función objetivo en Python.....	42
Ilustración 27: Estación de metro en Madrid.....	46
Ilustración 28: Demanda tras pasar servicio 8.....	49

4 OBJETIVOS

El objetivo principal de este proyecto es concreto: Proponer un mecanismo de solución a la aglomeración en las distintas estaciones de una línea urbana o metropolitana de ferrocarril (cercanías o metro) que resulte fácil de aplicar y de implementar.

En primer lugar, evitar estas aglomeraciones permitirá que las estaciones no superen su aforo máximo y previsiblemente que tampoco se alcance un valor cercano al máximo. Acumulaciones cercanas a estos valores máximos presentan muchos inconvenientes a nivel de seguridad (piénsese en lo que ocurre en estaciones de metro en líneas altamente congestionadas en megalópolis como Tokyo) y por tanto mantener niveles de congestión lejanos a estos valores máximos ser una prioridad en el modelo propuesto. El mecanismo propuesto consiste en controlar el acceso de pasajeros a los andenes en función de la capacidad disponible y variable de los trenes a lo largo de la línea. La capacidad variable se consigue mediante la reserva de vagones en las primeras estaciones de la línea. Los vagones reservados van siendo liberados a medida que el tren se acerca a estaciones con mayor demanda, reduciendo de esta forma los tiempos medios de espera en las estaciones con mayor número de usuarios y evitando, al disponer de mayor capacidad, que se produzcan enormes congestiones en estas estaciones como consecuencia de que los vagones se han llenado de usuarios en estaciones precedentes.

En segundo lugar, una vez se hayan conseguido unos niveles de congestión razonables, podemos proponernos un objetivo más ambicioso: Disminuir la cantidad de pasajeros que esperan en cada estación, lo que no solo aumentará la eficiencia de la línea de tren, sino que también aumentará la satisfacción de los usuarios.

5 INTRODUCCIÓN

El número de habitantes en este mundo está creciendo rápidamente. Se estima que en 2050 la población mundial habrá llegado a los 10.000 millones de habitantes (Roser, 2013) Además, desde el comienzo de la industrialización, la población que está trasladándose a vivir en grandes ciudades no para de crecer. A día de hoy, alrededor del 50% de la población mundial vive en ciudades de más de 300.000 habitantes y se estima que ese número alcanzará el 70% en el año 2050. (Montañés, 2018)

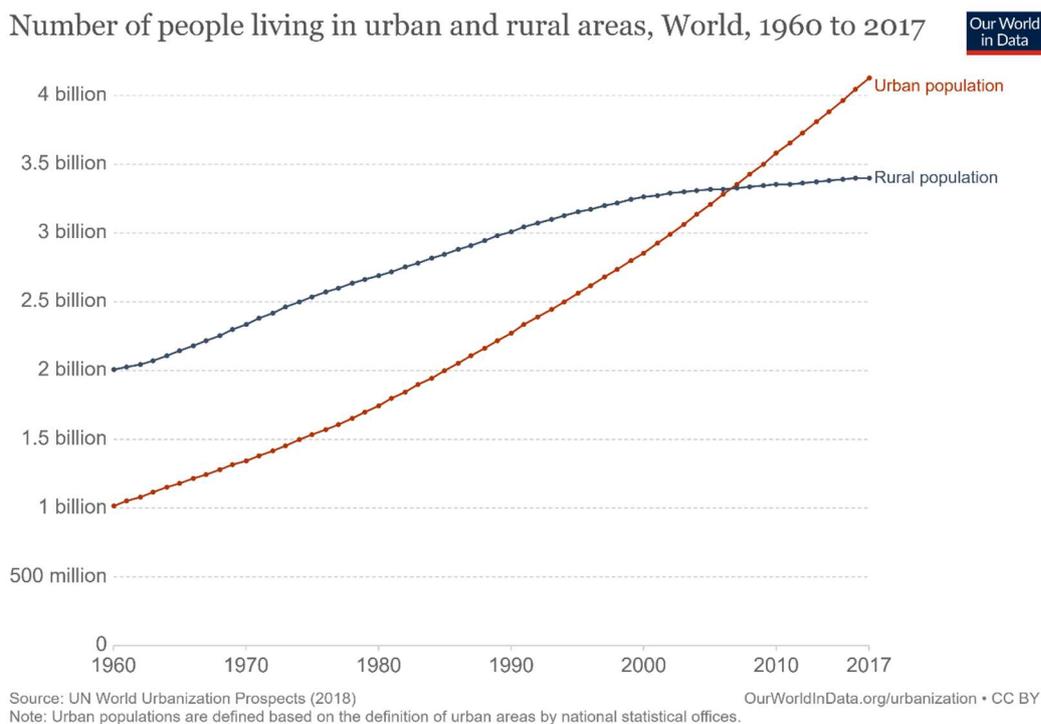


Ilustración 1. Aumento de población en áreas urbanas (Fuente: UN World Urbanization Prospects)

Este último año se ha puesto de manifiesto la importancia de evitar aglomeraciones. La pandemia global seguramente traerá como consecuencia que la distancia social sea una prioridad en el futuro. En estas grandes ciudades, son típicas las aglomeraciones en el transporte público y en las estaciones durante las horas punta. Según varios estudios como (Shi, et al., 2020) la relación entre el número de casos por COVID-19 y el número de pasajeros en trenes está estrechamente relacionado. Algo que, según este estudio, no ocurre con el transporte en avión y coche.

Otro aspecto a tener en cuenta es el cambio climático. El cambio climático es una realidad y existen diversos campos en los que se esperan grandes cambios en los próximos años. Entre ellos está la ganadería, la moda o el transporte.

El transporte es responsable del 26% de las emisiones de gases de efecto invernadero a nivel mundial. En concreto, los automóviles particulares contribuyen con un 18% del total (Chapman, 2007). El problema con los automóviles particulares no sólo acaba ahí. Fabricar un coche particular tiene un proceso muy costoso tanto a nivel de materia prima como de impacto medioambiental.

Como hemos dicho anteriormente, la población mundial crece y en concreto en las grandes ciudades. Que todos los habitantes tengan coche propio puede ser insostenible en un futuro cercano.

Para que un ciudadano no sienta la necesidad de comprarse un automóvil se antoja clave mejorar el transporte público, tanto para largas distancias como a nivel urbano e interurbano. En muchas ciudades como Madrid con la Normativa de Zona de Bajas Emisiones (El País, 2021) se están promoviendo leyes que incentivan a los ciudadanos a usar el transporte público.

En los últimos meses ha aparecido un nuevo factor que podría frenar el aumento de ciudadanos que usan el transporte público cada día: el teletrabajo. Sin embargo, las estadísticas nos siguen mostrando que la cantidad de personas que viajan en metro y tren es inmensa.

A nivel infraestructural la industria ferroviaria; interurbana, nacional e internacional, debe mejorar para que la transición al transporte urbano sea lo más llevadera posible para los ciudadanos. Además, el uso de estrategias que ayuden a mejorar el flujo de pasajeros, la seguridad de estos y optimicen el servicio público pueden ser igualmente importantes.

Este trabajo analiza una de estas estrategias: El modelo propuesto intentará abordar los problemas ocasionados por la aglomeración de pasajeros en las estaciones de líneas de ferrocarril de tránsito rápido en grandes redes del mundo a través de una idea, a priori, fácil de implementar.

6 REVISIÓN DE LA LITERATURA

Por todos los motivos que hemos expuesto hasta ahora, en los últimos años se han publicado numerosos estudios cuyo objetivo era atajar la congestión en estaciones de tren y metro de grandes ciudades.

Cada uno de estos estudios ha usado un enfoque diferente. Sin embargo, se pueden agrupar en 3 grandes grupos:

1. Optimización de los horarios de servicios para que se adapten a la demanda.

Se ha demostrado que la optimización de horarios es una forma muy efectiva de controlar la demanda y evitar aglomeraciones. Para dar con el horario correcto se han usado varios métodos.

Por ejemplo, en el estudio de (Canca, et al., 2014) se construye un horario de trenes adaptado a una demanda de pasajeros dinámica. En él se estudia la línea C5 de la estación de metro de la ciudad de Madrid. La demanda se presenta como una variable continua en función del tiempo con ciertas horas punta a lo largo del día. El horario de cada tren se basa en las aproximaciones de la demanda mediante una serie de variables y restricciones específicas.

Otro estudio que merece la pena mencionar sería el estudio de (Barrena, et al., 2014). Como en el anterior estudio mencionado, se tiene en cuenta una demanda de pasajeros dinámica. Después de analizar todas las variables y propiedades del problema, se implementa un método de algoritmo genético. Se consigue reducir el tiempo de espera de pasajeros en un 26% si se compara con el uso de métodos como el algoritmo Branch-and-Cut.

Otro estudio enfocado en el diseño de horarios es el de (Niu & Zhou, 2013). Basándose en un registro dependiente del tiempo de viajes con origen y destino conocido, Niu desarrolla un modelo de optimización no lineal para obtener horarios óptimos. La última hora de llegada de los pasajeros a bordo es usada para calcular periodos de tiempo de espera que puedan ser efectivos. El modelo se resuelve con un algoritmo genético al igual que en (Barrena, et al., 2014).

He de destacar estudios cuyo enfoque está más dirigido a congestiones ocurridas por una emergencia. Aparte de las aglomeraciones que pueden producirse durante una hora punta en una jornada laboral común, también pueden ocurrir situaciones como accidentes o retrasos de trenes que provoquen una gran acumulación de pasajeros.

Estudios como (Cadarsó, et al., 2013) afrontan esta problemática. El estudio propone un modelo de optimización de horarios y material móvil junto con el estudio del comportamiento de los pasajeros. Una de las mayores dificultades que presenta el problema es que ambos aspectos están ampliamente relacionados. El comportamiento de los pasajeros dependerá en gran medida de la respuesta de los operadores.

Sin embargo, para estaciones sobresaturadas con una alta congestión, como ocurre en estaciones de Shanghai o Tokyo, la optimización de los horarios no siempre es efectiva. Esto nos lleva al segundo enfoque.

2. Ofertas y descuentos para incentivar cambios de horario en pasajeros.

En general, la mayoría de las aglomeraciones se producen en horas punta, es decir, en horas en las que la mayoría de la población debe desplazarse para trabajar. Habrá diversas situaciones esporádicas en las que una estación puede sobrecongestionarse (acontecimiento deportivo, accidente...) pero es mucho más relevante enfocar el problema en estas horas punta. Si un ciudadano pierde mucho tiempo en largas esperas tanto al entrar como al salir a trabajar, estará perdiendo a su vez calidad de vida.

Diversos estudios plantean incentivar a los pasajeros a retrasar o adelantar la hora a la que suben al tren. Un ejemplo de ello sería el estudio llevado a cabo por Graham Currie (Currie, 2010), estudio que usaba la bajada de precios como disuasión. El estudio de Currie consistía en permitir los viajes en tren antes de las 7 de la mañana totalmente gratis. Esto provocó una bajada de flujo de pasajeros de entre el 1,2% y el 1,5% en los picos de congestión más alta.

Otro estudio, (Yang & Tang, 2018), usaba una estrategia de recompensa. Por cada cierto número de viajes en hora punta, se regalaba un viaje a una hora menos concurrida. La intención era que muchos pasajeros adelantaran o retrasaran su viaje a una hora poco concurrida con el fin de aprovechar el viaje gratis.

El problema de este tipo de estudios es la dificultad de implementación. En primer lugar, los horarios de los pasajeros están específicamente definidos por sus horarios de trabajo, lo cual es muy difícil de cambiar. Incluso, implementar en la realidad un estudio como el de Currie podría provocar estratificación de la sociedad. Los ciudadanos con más posibilidades de querer ahorrarse el precio de un viaje en tren serán aquellos con menor capacidad económica.

3. Implementar estrategias de control del flujo de pasajeros

En este grupo podría incluirse el modelo que se propone en este trabajo fin de grado, limitar el aforo de entrada de pasajeros a los andenes de algunas de las estaciones mediante la reserva de vagones (actuando sobre la capacidad disponible del tren) para intentar optimizar algunos de los problemas que se generan en estas aglomeraciones: situaciones de riesgo por acumulación de pasajeros cerca de las vías, problemas de

seguridad en los andenes y tiempos de espera. Nótese que la idea consiste en permitir el acceso a los andenes en función de la capacidad del tren, que resultará variable a lo largo de la línea. Se reservan vagones en las primeras estaciones y se liberan posteriormente en otras estaciones más concurridas, aumentando en estas la capacidad de los vehículos. De esta forma se mejoran los tiempos de espera. Para implementar estas estrategias será necesario describir las relaciones entre las variables de las que depende el flujo de pasajeros en un sistema de metro.

Por último, existen diversos métodos basados en controlar el flujo de pasajeros de los sistemas de metro. Es decir, controlar el flujo de pasajeros en trenes, zonas de espera y demás zonas comunes de una estación ferroviaria con el fin de evitar la congestión y aglomeración en lugares determinados.

Existen numerosos estudios que han propuesto modelos de optimización: En el modelo propuesto por (Jiang, et al., 2018) la medida implementada consiste en controlar el volumen de entrada de los pasajeros en una estación concreta. Estos estudios dan lugar a formulaciones matemáticas que coordinan y relacionan el número de pasajeros, el horario y la capacidad de los trenes en todo el sistema de metro.

Para optimizar el modelo pueden ser utilizados diversos algoritmos o programas. Por ejemplo, en el estudio realizado por (Xu, et al., 2016) para la resolución del método se usó un algoritmo genético. El objetivo del estudio era minimizar el tiempo medio que permanecía un pasajero en el sistema de metro combinando estrategias como ralentizar el tiempo en las entradas y en los pasillos.

El modelo propuesto se basa en el artículo preliminar de (Shi, 2020). Dependiendo de la capacidad de los trenes y considerando una demanda de pasajeros variable con respecto al tiempo, Shi propone un modelo que reserva vagones en cada servicio a lo largo de las distintas estaciones de una línea. El modelo intenta minimizar tanto el número de pasajeros esperando en cada estación como el riesgo a que una estación se congestione de una forma alarmante. Para resolver este tipo de modelos se recurre en general a algoritmos metaheurísticos. EN nuestro caso, el modelo propuesto será resuelto de forma exacta utilizando el Solver GUROBI:

7 DESCRIPCIÓN DEL MODELO

Para el desarrollo de un modelo correcto será necesario utilizar un número muy elevado de datos históricos como el número de pasajeros que esperan cada día o la estación a la que se dirigen, para lo que harán falta tecnologías fiables para poder recopilar esta información en tiempo real. Es algo que consideramos altamente factible, pues las tecnologías de recopilación de información han mejorado mucho en los últimos años. Pueden usarse tecnologías como las que se usan para predecir el tráfico en la carretera en aplicaciones como Google Maps o Waze (ABC 2017). En ellas, los usuarios envían información de manera anónima sobre su localización, algo que serviría para recopilar el tránsito en las estaciones de metro.

Los mayores problemas de congestión en las estaciones de metro se producen en horas punta están relacionadas con la jornada laboral. Es decir, de lunes a viernes y durante determinadas horas del día la aglomeración de pasajeros podrá suponer un problema en varias estaciones de las líneas férreas de tránsito rápido de una gran ciudad.

Si se contabilizaran los pasajeros que suben a bordo durante estas horas a lo largo de varias semanas, con total seguridad los valores no variarían mucho de un día a otro, o de una semana a otra. Por lo tanto, otra opción mucho más fácil y económica para obtener datos de entrada para el modelo propuesto sería mediante su recopilación y posterior realización de una previsión de la demanda.

Se considerará una línea de metro con $|S|$ estaciones y un horario compuesto por $|I|$ servicios. Estudiaremos la línea de trenes en una sola dirección, ya que hacerlo en 2 direcciones requeriría el doble de variables y datos, y no aportaría ningún valor adicional. Tanto para una dirección como para otra se aplicaría el mismo método de reserva de vagones, por lo que ambos problemas están desacoplados. De esta forma se simplifica el problema.

Además, supondremos conocido el número de pasajeros que llegan a cada estación en cada instante durante el horizonte de planificación, además de la estación a la que se dirigen. Dispondremos por tanto de una descripción de la demanda a lo largo del tiempo, formada por una serie de grupos de pasajeros P , de los que sabremos tanto la hora a la que llegan a la estación origen del viaje como su estación de destino.

Un aspecto vital en el modelo propuesto se refiere a la capacidad de las zonas de espera en cada estación. Dado que el modelo deberá determinar la cantidad de personas que se acumulan en estos espacios será fundamental disponer de sus aforos:

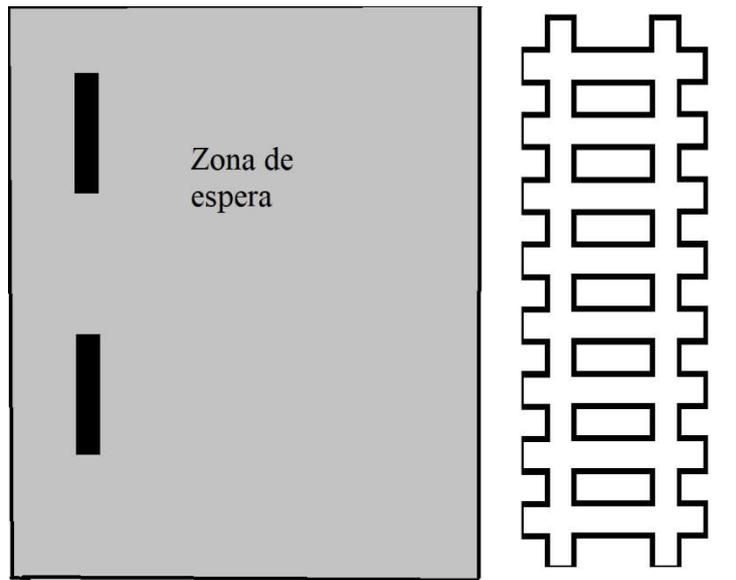


Ilustración 2: Zona de espera de pasajeros

Para evitar problemas de congestión en las plataformas se usará un procedimiento de reserva de vagones. A continuación, mostraremos la casuística del modelo a partir de un ejemplo sencillo:

Se trata de una línea con más de 2 estaciones y dos servicios realizados por dos trenes de 2 vagones con capacidad para un pasajero cada uno de ellos. Para comprobar el efecto que tiene la reserva de vagones nos fijaremos en las dos primeras estaciones de la línea. En cuanto a los pasajeros que esperan en cada estación la situación sería la siguiente:

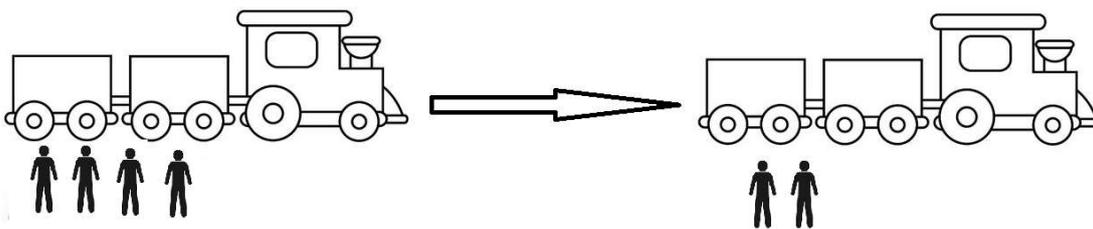


Ilustración 3: Pasajeros esperando el primer tren

En este ejemplo habrá 4 pasajeros esperando en la estación 1 y 2 en la estación 2. Si no hay reserva de vagones tanto el servicio 1 como el servicio 2 se usarán para satisfacer la demanda de la estación 1 mientras que los 2 pasajeros que esperan en la estación 2 tendrán que esperar a un tercer tren.

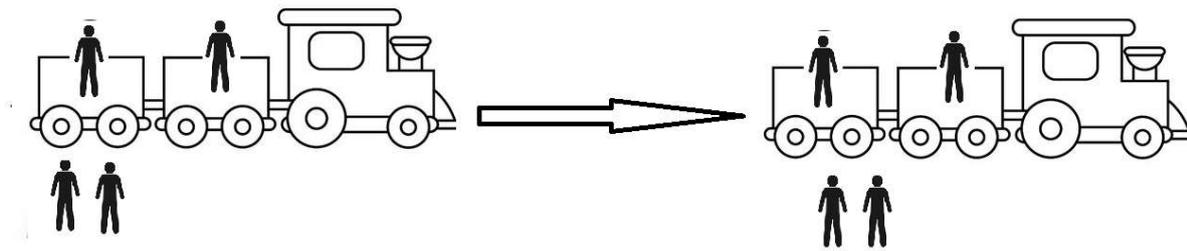


Ilustración 4: Pasajeros a bordo del tren sin reserva de vagones

Por tanto, la estación 2 será desatendida durante un largo periodo de tiempo y si el flujo de pasajeros en la estación 2 fuera aun mayor las posibilidades de aglomeración y congestión en ella serían muy altas.

En cambio, si se reservara un vagón en cada uno de los trenes que realizan los dos primeros servicios, se impide que los vagones se completen de inmediato en la estación 1. Esto hará que la demanda en la estación 2 pueda ser poco a poco satisfecha, disminuyendo la posibilidad de congestión.

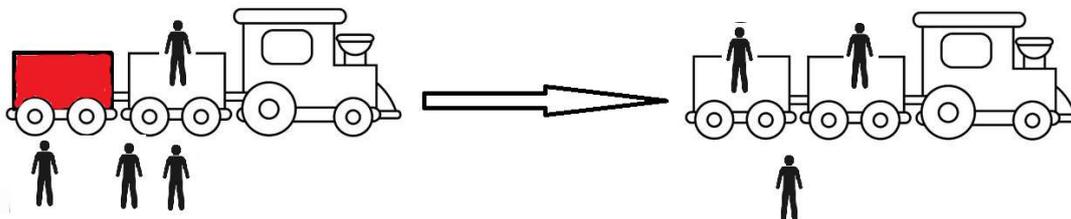


Ilustración 5: Ejemplo con reserva de vagones

Aunque en el ejemplo mostrado la cantidad de pasajeros esperando no sería excesiva, si extrapolamos la misma situación a la realidad podemos encontrarnos con ciertos problemas y dificultades.

1. Piénsese ahora en una situación real como por ejemplo la estación London Bridge (más de 100.000 pasajeros al día), la saturación de los trenes que realizan los servicios en hora punta podría generar problemas de congestión en las plataformas de las sucesivas estaciones, principalmente las mas cercanas: Tiempos de espera: Los pasajeros en la estación siguiente tendrían que esperar durante un largo periodo de tiempo para poder subirse a bordo de un tren.

2. Congestión: Los pasajeros se acumularán en la zona de espera de la estación siguiente. Esto, además de resultar incómodo, pueden provocar situaciones de peligro e inseguridad si el número de pasajeros es muy alto. Además, como hemos comentado en la introducción hará difícil mantener la distancia de seguridad.

Hipótesis

7.1.1 Hipótesis 1

Con el objetivo de poder simplificar cálculos hemos incluido una serie de hipótesis que no deberían influir en el desarrollo del problema:

Las estaciones de origen y destino de cada pasajero serán conocidas, así como su tiempo de llegada. No contemplaremos la posibilidad de que un pasajero cambie de estación de destino durante el trayecto.

7.1.2 Hipótesis 2

Los grupos de pasajeros se suponen perfectamente mezclados. Esto supone que la proporción de pasajeros esperando con destino a la estación 'k' será exactamente igual a la proporción de pasajeros que se suben a bordo con destino a la estación 'k', para un tren y una estación dada.

En el gráfico a continuación podemos comprender fácilmente esta suposición. Se trata de un tren con capacidad para 4 personas, al que están a punto de montarse 8 personas. Estas 8 personas están divididas en 4 grupos, cada grupo con destino a una estación diferente.

Debido a que sólo caben 4 personas, habrá 4 que suban al tren y 4 que tengan que esperar en la estación. La proporción de las estaciones de destino entre los pasajeros esperando y entre los pasajeros que suben a bordo es la misma. Esto es lo que implica nuestra suposición.

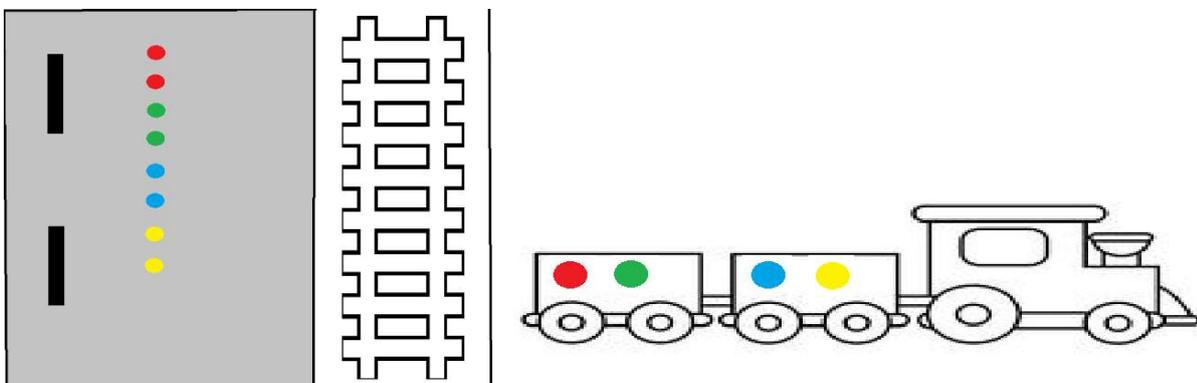


Ilustración 1: 2ª suposición

7.1.3 Hipótesis 3.

La demanda total de pasajeros es menor que la capacidad total de los trenes, para poder garantizar que todos los pasajeros puedan realizar su viaje.

Datos requeridos

Para poder resolver el modelo matemático necesitaremos un conjunto de datos que permitirán formular de forma consistente todas las restricciones. Estos datos se refieren a la información relacionada tanto con los horarios de los distintos servicios como con el flujo de pasajeros a lo largo de la línea.

Sin embargo, habrá otros muchos datos que serán totalmente arbitrarios. Estos datos nos serán útiles para modelar ciertas restricciones, además de sernos útiles a la hora de valorar la importancia de cada parámetro.

Todos los datos que usaremos en el entorno Python/GUROBI serán extraídos de un archivo Excel. Este archivo Excel contendrá tanto datos unitarios (parámetros del modelo) como diversas tablas que proporcionan información de horarios y flujo de pasajeros.

7.1.3.1 Conjuntos

En primer lugar, como hemos detallado en la introducción del modelo, necesitaremos una serie de conjuntos que recorreremos en Python, para modelar nuestras restricciones.

Estos conjuntos serán:

P , que consistirá en los grupos de pasajeros que llegan a una hora determinada a cada una de las estaciones.
 S , que contiene todas las estaciones desde la 1ª hasta la última en la dirección seleccionada. Además tendremos que declarar el conjunto I , que representará los diferentes servicios que recorren las distintas estaciones.

Por último, el conjunto T se referirá al tiempo. Nuestro horizonte temporal contemplará una hora, dividiendo esta en períodos de minutos.

7.1.3.2 Parámetros

Los distintos parámetros necesarios para formular el modelo son:

n_p : Se refiere al número de pasajeros en un grupo $p \in P$.

o_p : La estación de origen de un grupo de pasajeros $p \in P$.

t_p : La hora de llegada del grupo de pasajeros del grupo $p \in P$, a la estación de origen o_p .

d_p : La estación de destino del grupo de pasajeros p .

La Tabla 1 muestra la forma que que estos datos se estructuran en el libro de Excel que contiene los datos del problema.

P	Numero de pasajeros	Hora de llegada	Estación de origen	Estación de destino
1	1.00	1.00	1.00	2.00
2	2.00	1.00	1.00	3.00
3	3.00	1.00	1.00	4.00
4	3.00	1.00	1.00	5.00
5	1.00	1.00	1.00	6.00
6	2.00	1.00	1.00	7.00
7	1.00	1.00	1.00	8.00
8	1.00	1.00	2.00	3.00
9	2.00	1.00	2.00	4.00
10	2.00	1.00	2.00	5.00

Tabla 1: n_p, d_p, o_p, t_p representados en Excel

Estos datos se obtendrán directamente de la demanda. Esta demanda se representa en una matriz distribuida de acuerdo con el trayecto realizado (origen, destino) y el instante de tiempo dentro del horizonte de planificación.

Origen	Destino		1	2	3	4	5
1	2	1.2	1.00	1.00	1.00	2.00	1.00
1	3	1.3	2.00	2.00	1.00	1.00	3.00
1	4	1.4	2.00	5.00	4.00	4.00	6.00
1	5	1.5	1.00	1.00	1.00	3.00	1.00
1	6	1.6	1.00	2.00	2.00	1.00	3.00

Tabla 2: Demanda en los primeros 5 minutos

La notación correspondiente a estos datos y otros parámetros que se usará en la formulación del problema se muestra a continuación.

d_{ik} : La hora de salida de un determinado servicio i desde la estación.

H_k : El máximo aforo permitido en una determinada estación k .

U_k : El máximo número de pasajeros esperando que se considere seguro en una determinada estación k .

Φ : Usaremos esta variable en la función a minimizar. Será el coeficiente de riesgo de acumulación de pasajeros.

θ_1, θ_2 : Pesos de los 2 componentes de la función objetivo, el riesgo de aglomeración de pasajeros r_k y el número de pasajeros esperando w_k . Ver sección 7.1.3.4.

M positivo muy grande.

n_{max} : Número máximo de vagones reservables.

Q : Número total de vagones en cada tren.

C : Capacidad de cada vagón.

Después de definir los distintos datos necesarios pasamos a definir las variables de decisión del modelo.

En realidad, la única variable de decisión real será el número de vagones reservados por cada tren y estación. El resto de las variables dependerán directamente de esta decisión. El número de vagones lo definiremos como n_{ik} refiriéndonos al número de vagones reservados en un servicio i a la salida de cierta estación k . A partir del número de vagones que reservamos en cada tren y estación, aparecerán distintas variables intermedias que serán relevantes sobre todo en la función objetivo.

La función objetivo tratará de minimizar tanto el riesgo de aglomeración de pasajeros en las distintas estaciones como el número de pasajeros esperando. Ambos términos aparecerán multiplicados por los parámetros θ_1 y θ_2 definidos previamente.

Por lo tanto, las variables que deberemos declarar son:

7.1.3.3 Variables de decisión

n_{ik} : Serán las únicas variables realmente sujetas a decisión. Son números enteros que podemos definir como el número de vagones reservados en un el servicio i al salir de la estación k .

7.1.3.4 Variables intermedias

c_{ik} : Como hemos dicho con anterioridad, a partir de n_{ik} aparecerán distintas variables. La variable c_{ik} será en concreto la capacidad restante del tren que realiza el servicio i a su llegada a la estación k .

w_{ik} : Número de pasajeros esperando el servicio i en la estación k .

b_{ik} : Número de pasajeros que se montan en el tren que realiza el servicio i cuando llega a la estación k .

a_{ik} : Número de pasajeros que bajan en la estación k desde el servicio i .

v_{ik} : El número de pasajeros a bordo del tren que realiza el servicio i cuando sale de la estación k .

l_{ik} : Número de pasajeros que no han podido montarse en el tren que realiza el servicio i y han sido dejados en tierra en la estación k .

w_{iks} : En este caso haremos una nueva distinción relacionada con la estación de destino. w_{iks} será el número de pasajeros que esperan el servicio i en la estación k , con destino a la estación s .

l_{iks} : Al igual que con w_{iks} , l_{iks} será el número de pasajeros dejados en tierra por el servicio i en la estación k , cuyo destino era la estación s .

$w_k(t)$: El número de pasajeros que esperan en la estación k a lo largo del tiempo.

$r_k(t)$: El riesgo de aglomeración en la estación k a lo largo del tiempo.

w_k : El número total de pasajeros que esperan en la estación k .

r_k : El valor total de riesgo de aglomeración en la estación k .

Todas estas variables serán declaradas en Python mediante la función propia de GUROBI llamada *addvar*. Más adelante se darán detalles de cómo usaremos esta funcionalidad.

7.1.4 Restricciones

Todos estos datos y variables nos servirán para desarrollar las restricciones necesarias para llevar a cabo el modelo. Mientras que la mayoría de las restricciones estarán relacionadas con darle forma y sentido al problema, habrá algunas que harán mención a las distintas hipótesis que planteamos con anterioridad.

7.1.4.1 Restricciones 1

$$0 \leq n_{ik} \leq n_{m\acute{a}x}, \quad \forall i \in I, k \in S$$

El primer conjunto de restricciones hace referencia al número de vagones que podemos reservar. Para evitar que se reserven un número insostenible de vagones, hemos declarado el parámetro n_{max} . En esta restricción

simplemente limitamos el número de vagones que pueden ser reservados en cada servicio i al salir de la estación k a un valor máximo n_{max} .

7.1.4.2 Restricciones 2

$$n_{ik} \geq n_{i,k+1}, \forall i \in I, k, k+1 \in S$$

El segundo conjunto de restricciones fuerza que el número de vagones reservados en un tren a la salida de una cierta estación sea igual o menor al número de vagones reservados que en la estación anterior. Es decir, el número de vagones que se reservan en un cierto tren van disminuyendo conforme este circula por las distintas estaciones.

El objetivo del modelo consiste en evitar la aglomeración de pasajeros en ciertas estaciones. Para ello, se reservan vagones en estaciones previas, permitiendo así que pasajeros en estaciones posteriores puedan subirse a bordo. Por este motivo, el el número de vagones reservados irá disminuyendo conforme el tren haga su recorrido, abriendo estos vagones al público para disminuir el número de pasajeros esperando.

7.1.4.3 Restricciones 3

$$v_{ik} = \begin{cases} b_{ik} & \text{si } k = 1 \\ v_{ik-1} - a_{i,k} + b_{ik} & \text{en cualquier otro caso} \end{cases}, \forall i \in I, k \in S$$

En esta restricción nos encontramos con variables intermedias, es decir, variables que son necesarias para desarrollar el modelo pero que no constituyen la decisión principal

Este conjunto de restricciones hace referencia al número de pasajeros a bordo del tren que realiza el servicio i al salir de la estación k . Puede ser explicado de forma muy sencilla si se trata de la primera estación del recorrido: El número de pasajeros a bordo del tren serán todos aquellos que se monten en él.

Si, por el contrario, se trata de una estación superior tendremos que añadir todos aquellos pasajeros que estaban previamente a bordo (los que venían en el tren desde la estación anterior $k-1$) y restar todos aquellos que bajen del tren en dicha estación.

7.1.4.4 Restricciones 4

$$w_{ik} = \begin{cases} \sum_{p \in \{p' \in P | o_{p'} = k, 0 \leq t_{p'} \leq d_{1k}\}} n_p & \text{si } i = 1 \\ l_{i-1,k} + \sum_{p \in \{p' \in P | o_{p'} = k, d_{i-1,k} \leq t_{p'} \leq d_{ik}\}} n_p & \text{si } i \neq 1 \end{cases} \quad \forall i \in I, k \in S, p \in P$$

Al igual que en el conjunto anterior de restricciones, el conjunto 4 estará formado únicamente por variables intermedias. En este caso las restricciones hacen referencia al número de pasajeros que están esperando un determinado servicio i en la estación k .

Este número será igual a la cantidad de pasajeros que no han podido subirse en el tren anterior (por falta de espacio) más los pasajeros que llegaron a la estación k después de que el anterior tren partiera.

En el primer tren, obviamente, el número de pasajeros dejados en tierra anteriormente será igual a 0. En cuanto a la sumatoria de pasajeros, deberemos de tener en cuenta varias condiciones.

La primera de ellas especifica que el origen del grupo de pasajeros p debe de ser la estación k . Por último, la hora de llegada del grupo de pasajeros debe de ser posterior al instante en que el servicio $i - 1$ llega la estación k y, además, anterior a la hora de llegada del servicio i .

7.1.4.5 Restricciones 5

$$w_{i,k,s} = \begin{cases} \sum_{p \in \{p' \in P | o_{p'} = k, 0 \leq t_{p'} \leq d_{1k}\}} n_p & \text{si } i = 1 \\ l_{i-1,k,s} + \sum_{p \in \{p' \in P | o_{p'} = k, d_{i-1,k} \leq t_{p'} \leq d_{ik}\}} n_p & \text{si } i \neq 1 \end{cases} \quad \forall i \in I, \forall k, s \in S, k < s$$

El conjunto de restricciones 5 y 4 están muy relacionadas, ya que en esta ocasión queremos calcular el número de pasajeros esperando al servicio i en la estación k , cuyo destino es la estación s .

Es por ello que la restricción será prácticamente igual. Variará únicamente que los pasajeros que no han podido subirse en el anterior tren, deben además tener como destino la estación s .

7.1.4.6 Restricciones 6

$$c_{i,k} = \begin{cases} (Q - n_{ik}) \times C & \text{si } k = 1 \\ (Q - n_{ik}) \times C - (v_{i,k-1} - a_{ik}), & \text{si } k \neq 1 \end{cases} \quad \forall k \in S, \forall i \in I$$

En este conjunto de restricciones calcularemos la capacidad restante de cada servicio i al llegar a la estación k . En este caso, usaremos la variable de decisión n_{ik} pues la capacidad dependerá en gran medida del número de vagones disponibles y reservados.

La capacidad será igual al número de vagones que no están reservados multiplicado por la capacidad C de cada vagón en el caso de la estación 1. Si se trata de la segunda estación en adelante habrá que sumarle además la diferencia entre los pasajeros que abandonan el tren que realiza el servicio i en la estación k y los que se habían subido a bordo en la estación anterior.

7.1.4.7 Restricciones 7

$$b_{i,k} = \min\{c_{ik}, w_{ik}\}, \quad \forall i \in I, \forall k \in S$$

El conjunto de restricciones siete permite calcular de forma sencilla el número de pasajeros que pueden subir a bordo en el servicio i al llegar a la estación k . Dependerá de la capacidad del tren a llegar a la estación k y del número de pasajeros esperando el servicio i , siendo b_{ik} el mínimo entre estas dos cantidades.

7.1.4.8 Restricciones 8

$$l_{ik} = w_{ik} - b_{ik}, \quad \forall i \in I, k \in S$$

Las restricciones número ocho muestran el número de pasajeros que son dejados en tierra por el servicio i en la estación k . Lógicamente, este número será la diferencia entre el número de pasajeros esperando el servicio i en la estación k y los que tienen la oportunidad de montarse en el tren que realiza el servicio.

7.1.4.9 Restricciones 9

$$b_{iks} = \frac{w_{iks}}{w_{ik}} \times b_{ik}, \quad \forall i \in I, k, s \in S, k < s$$

Las restricciones número nueve están estrechamente relacionadas con una de las hipótesis planteadas con anterioridad, en concreto la [Hipótesis 2](#). Según esta hipótesis, la proporción entre el total de pasajeros esperando en una estación y los pasajeros esperando con un cierto destino será la proporción que se aplica a los pasajeros que suben al tren que realiza el servicio.

7.1.4.10 Restricciones 10

$$l_{iks} = w_{iks} - b_{iks}, \quad \forall i \in I, \forall k, s \in S, k < s$$

El décimo conjunto de restricciones, al igual que el conjunto ocho de restricciones muestra el número de pasajeros dejados en tierra por el servicio n i en la estación k . En este caso, sin embargo, también se especifica la estación de destino de estos pasajeros.

7.1.4.11 Restricciones 11

$$a_{ik} = \sum_{s=1}^{k-1} b_{isk}, \forall i \in I, k \in S$$

En este caso, el conjunto de restricciones 11 muestra cómo calcular el número de pasajeros que se bajarán con la llegada del servicio i a la estación k . Este número se puede calcular como la suma de todos aquellos que se montaron en este tren en una estación anterior, cuyo destino era la estación k .

7.1.4.12 Restricciones 12

$$l_{|I|k} = 0, \forall k \in S$$

$l_{|I|k}$ hace referencia al número de pasajeros que no han podido acceder al último tren en la estación k . Uno de los requisitos que planteamos en este problema es que todos los pasajeros puedan subirse a bordo de algún tren y llegar a su destino. Debido a este requisito, el último tren no debería dejar a ningún pasajero en tierra.

7.1.4.13 Restricciones 13

$$w_k(t) = \sum_{p \in \{p' \in P | o_{p'} = k, 0 \leq t_{p'} \leq t\}} n_p - \sum_{i \in \{i' \in I | d_{i'} \leq t\}} b_{ik}, \forall k \in S, t \in [0, T]$$

$w_k(t)$ es una variable muy relevante en nuestro el modelo, pues se encuentra dentro de nuestra la función objetivo. En esta restricción, queremos especificar el número de pasajeros que estará esperando en una estación k a lo largo del tiempo.

Para ello, debemos calcular la diferencia entre los pasajeros que han llegado a la estación (entendiendo esta como su estación de origen) antes del instante t y los que han tenido la oportunidad de subirse a bordo de algún tren, antes de t .

Para ello, sumaremos todos los pasajeros cuya estación de origen sea k y cuyo horario de llegada sea anterior a t . A esto le restaremos la suma de todos aquellos pasajeros que hayan subido a bordo de un tren en la estación k , antes de t .

7.1.4.14 Restricciones 14

$$r_k(t) = \begin{cases} 0, & \text{si } 0 \leq w_k(t) \leq U_k \\ \frac{w_k(t) - U_k}{H_k - U_k} \cdot \varepsilon, & \text{si } U_k < w_k(t) < H_k, \\ M, & \text{si } w_k(t) \geq H_k \end{cases} \quad \forall k \in S, t \in [0, T]$$

La otra variable relevante dentro de nuestra función objetivo sería el riesgo de aglomeración $r_k(t)$. Este riesgo será igual a 0 si no superamos el umbral de riesgo U_k y un valor muy grande M si superamos un valor que consideramos inaceptable como sería H_k .

Para los valores que se encuentren entre estos 2, se propone una fórmula que hará el valor de $r_k(t)$ mayor conforme nos acerquemos a H_k .

7.1.4.15 Restricciones 15

$$r_k = \int_0^T r_k(t) dt, \quad \forall k \in S$$

Integrando $r_k(t)$ podemos calcular el área de esta función, que es una forma mucho más efectiva y clara de comprobar el riesgo que se corre a lo largo del tiempo, en una estación k determinada.

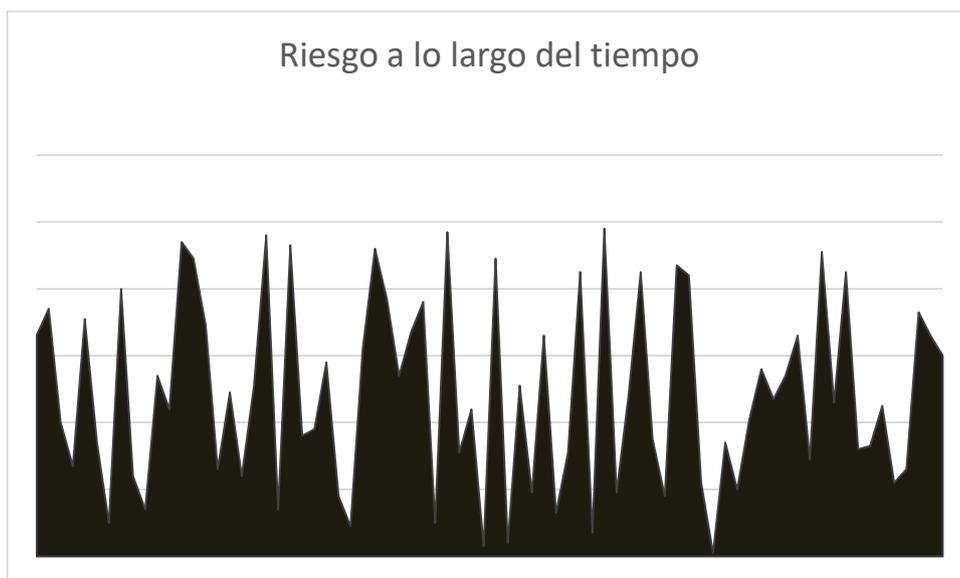


Gráfico 2: Ejemplo de riesgo de aglomeración

Como podemos ver en el gráfico superior, puede ocurrir que el riesgo de aglomeración en una cierta estación varíe de manera continua. Por ejemplo, una estación de metro puede estar relativamente tranquila

durante todo el horario de tarde. Sin embargo, puede abarrotarse repentinamente en una hora punta en la que se acaba la jornada laboral.

Es por ello que usar la integral de $r_k(t)$ puede hacer que el modelo cumpla el objetivo propuesto de una forma más precisa.

7.1.4.16 Restricciones 16

$$w_k = \int_0^T w_k(t) dt, \quad \forall k \in S$$

Al igual que con el riesgo de aglomeración, para poder calcular de manera más precisa la cantidad de pasajeros que esperan en cada estación la integral de $w_k(t)$ resultará mucho más útil.

7.1.5 Función Objetivo

$$\text{Min } \theta_1 \cdot \sum_{k \in S} w_k + \theta_2 \cdot \sum_{k \in S} r_k$$

El objetivo del modelo consistirá en minimizar tanto el número de pasajeros que espera en las distintas estaciones, como el riesgo de que en una estación se aglomeren demasiadas personas. Dependiendo de los valores que demos tanto a θ_1 como a θ_2 , a la hora de optimizar, el modelo dará más importancia al riesgo de aglomeración o a la cantidad de pasajeros esperando.

El problema con el que nos encontramos, como muchos otros, es difícil de resolver por métodos exactos. Tanto por la cantidad de soluciones posibles como por el tiempo que requeriría existen otros métodos que pueden ser más interesantes a la hora de buscar una solución óptima. Métodos metaheurísticos o aproximados como algoritmos genéticos pueden ser útiles para resolver este tipo de problemas.

En este caso, como se comentó anteriormente, se usará un GUROBI como solver de optimización. Un software que puede resolver modelos LP, QP y MIP. Es decir, modelos tanto lineales como cuadráticos además de mixtos enteros.

Algunas de las restricciones del modelo resultan no lineales. Para poder resolver el problema con el solver GUROBI habrá que linealizar alguna de estas restricciones, aquellas que no resulten cuadráticas, con el coste de aumentar el número de restricciones y variables del problema.

7.1.6 No linealidades

Una ecuación será no lineal cuando no cumpla con el principio de superposición. Este teorema consiste en lo siguiente: El principio de superposición se puede definir como “el efecto de la suma de acciones es la suma de cada efecto de cada acción”. En definitiva, que el efecto de la suma es la suma de los efectos.

El principio de superposición se puede definir como “el efecto de la suma de acciones es la suma de cada efecto de cada acción”. En definitiva, que el efecto de la suma es la suma de los efectos.

En este caso, el principio de superposición nos permite descomponer una ecuación lineal en dos o más ecuaciones más sencillas. Para que una ecuación cumpla el principio de superposición, es decir, sea lineal debe cumplir las siguientes propiedades:

- Proporcionalidad: $f(y \cdot a) = y \cdot f(a)$

- Aditividad: $f(a + b) = f(a) + f(b)$

En el modelo que se ha propuesto, encontramos varias restricciones no lineales que pueden dificultar su resolución. El conjunto de restricciones 7, por ejemplo, en la que se define el número de pasajeros que se suben a bordo de cada servicio i en la estación k .

Además de estas, el conjunto de restricciones 9 sería no lineal, ya que contiene variables que aparecen dividiendo. Por último, las [restricciones 14](#), en las que se calcula el riesgo de aglomeración para un instante t en cada estación k , son restricciones no lineales.

Tanto las restricciones número 7 como las restricciones número 14 se pueden linealizar añadiendo una cota M de valor muy grande y un conjunto nuevo de variables.

Por ejemplo, las restricciones número 7, que calculan el número de pasajeros que se suben a bordo del servicio i en la estación k :

$$b_{i,k} = \min\{c_{i,k}, w_{i,k}\}, \quad \forall i \in I, \forall k \in S$$

Se podría formular así:

$$\begin{cases} b_{i,k} \leq c_{i,k} \\ b_{i,k} \leq w_{i,k} \\ b_{i,k} \geq c_{i,k} - M \cdot (1 - u_{i,k}) \\ b_{i,k} \geq w_{i,k} - M \cdot u_{i,k} \end{cases}$$

siendo u_{ik} una variable binaria, que tomará valor 1 si c_{ik} es mayor que w_{ik} y 0 si ocurre lo contrario. De esta manera, b_{ik} siempre será el mínimo entre c_{ik} y w_{ik} .

Sin embargo, el programa GUROBI permite usar la función $\min_-(b_{ik}, c_{ik})$ por lo que no hemos tenido que descomponer o modificar la restricción.

7.1.6.1 Linealización de la restricción 9

$$b_{i,k,s} = \frac{w_{i,k,s}}{w_{i,k}} \times b_{i,k}, \quad \forall i \in I, k, s \in S, k < s$$

La restricción 9 puede ser descompuesta de la siguiente manera, para que pueda ser procesada por el optimizador GUROBI:

$$\begin{cases} \sum_{k+1}^S X_{i,k,s} = 1 \quad \forall i \in I, \forall k \in S \\ \sum_{k+1}^S Y_{i,k,s} = 1 \quad \forall i \in I, \forall k \in S \\ w_{i,k,s} = X_{i,k,s} \cdot w_{i,k}, \quad i \in I, k, s \in S, s > k \\ b_{i,k,s} = Y_{i,k,s} \cdot b_{i,k}, \quad i \in I, k, s \in S, s > k \end{cases}$$

En este caso tanto X como Y serán variables continuas que irán de 0 a 1. Con $X_{i,k,s}$ se estará definiendo el porcentaje de pasajeros esperando a un tren i en la estación k , cuya estación de destino es s . Así pues, la suma de todos esos porcentajes debe dar 1.

7.1.6.2 Linealización de la restricción 14

$$r_k(t) = \begin{cases} 0, & \text{si } 0 \leq w_k(t) \leq U_k \\ \frac{w_k(t) - U_k}{H_k - U_k}, & \text{si } U_k < w_k(t) < H_k, \\ M, & \text{si } w_k(t) \geq H_k \end{cases} \quad \forall k \in S, t \in [0, T]$$

La restricción 14 es claramente no lineal pues dependiendo del rango de valores en el que se encuentre $w_k(t)$ tendrá unos valores u otros. Se puede representar de la siguiente manera:

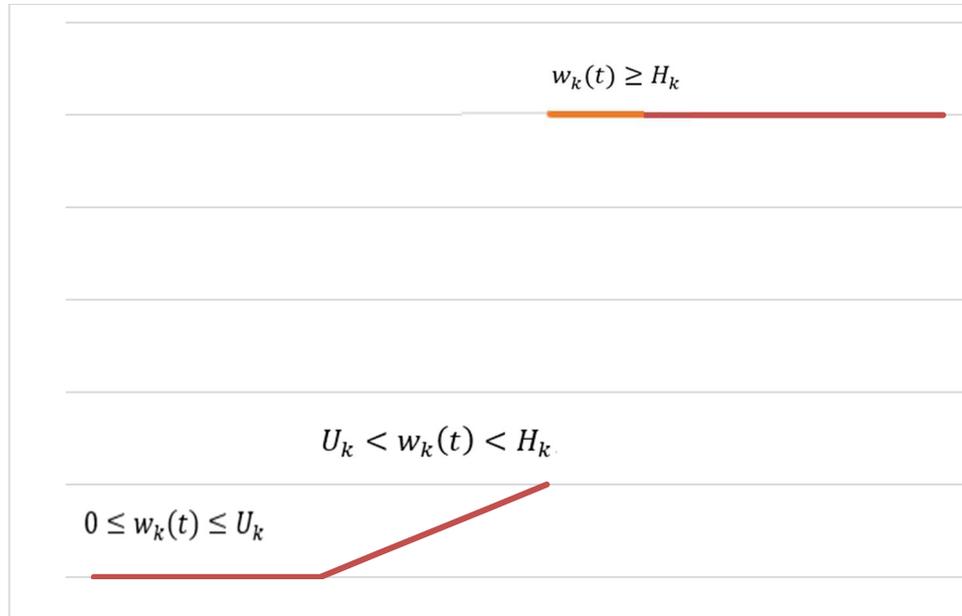


Gráfico 3: Riesgo de aglomeración en función de pasajeros

Para poder linealizar esta restricción se debe crear una única fórmula que adquiera un valor u otro dependiendo del valor que tome $w_k(t)$.

Para ellos se declaran nuevas variables complementarias:

$$\begin{cases} \delta_{1k}(t) + \delta_{2k}(t) + \delta_{3k}(t) = 1 & \forall k, t \in S, T \\ w_{1k}(t) + w_{2k}(t) + w_{3k}(t) = w_k(t) & \forall k, t \in S, T \\ w_{1k}(t) \geq 0 & \forall k, t \in S, T \\ \delta_{2k}(t) * U_k \leq w_{2k}(t) \leq \delta_{2k}(t) * H_k & \forall k, t \in S, T \\ M \cdot \delta_{3k}(t) \leq w_{3k}(t) \leq 2 \cdot M \cdot \delta_{3k}(t) & \forall k, t \in S, T \end{cases}$$

Donde δ_{1k} , δ_{2k} y δ_{3k} serán variables binarias. Estas restricciones harán que una de estas 3 variables sea igual a 1 dependiendo del valor de $w_k(t)$.

Finalmente la función que da el valor del riesgo $r_k(t)$ será:

$$r_k(t) = 0 + \frac{w_{2k}(t) - U_k \cdot \delta_{2k}(t)}{H_k - U_k} \cdot \varepsilon + M * \delta_{3k}(t) \quad \forall k \in S, \forall t \in T$$

Formulación del modelo usando notación simbólica

Conjuntos	
S	Conjunto de estaciones
I	Conjunto de trenes
$[0, T]$	Tiempo(Una hora en minutos)
P	Conjunto de demanda de pasajeros
Subíndices	
i	Índice de trenes
k, s	Índices de estaciones
t	Índice de tiempo
p	Índice de demanda de pasajeros
Parámetros	
n_p	Demanda de pasajeros en el grupo de pasajeros p.
o_p	Estación de origen del grupo de pasajeros p.
t_p	Hora de llegada a la estación de origen del grupo de pasajeros p.
d_p	Estación de destino del grupo de pasajeros p.
d_{ik}	Hora de llega del tren i a la estación k.
H_k	Aforo máximo de estación k.
U_k	Máximo número de pasajeros seguro en la estación k.
Φ	El coeficiente de riesgo de la acumulación de pasajeros.
θ_1, θ_2	Coeficientes de peso de pasajeros esperando y riesgo en la función objetivo.

M	Valor muy grande
$n_{m\acute{a}x}$	Número máximo de vagones que pueden ser reservados
Q	Número de vagones en cada tren
C	Capacidad máxima de cada vagón.
Variables	
n_{ik}	Número de vagones reservados en un tren i al llegar a la estación k .-Variable de decisión
c_{ik}	Capacidad restante de un tren i al llegar a la estación k .
w_{ik}	Pasajeros esperando al tren i en la estación k .
b_{ik}	Pasajeros que suben a bordo del tren i en la estación k .
a_{ik}	Número de pasajeros que se bajan del tren i en la estación k .
v_{ik}	Número de pasajeros a bordo del tren i al salir de la estación k .
l_{ik}	Número de pasajeros dejados atrás por el tren i al salir de la estación k .
w_{iks}	Número de pasajeros esperando al tren i en la estación k con destino a la estación s .
l_{iks}	Número de pasajeros dejados atrás por el tren i al salir de la estación k cuyo destino es la estación s .
$w_k(t)$	Número de pasajeros esperando en la estación k a lo largo del tiempo.
$r_k(t)$	Riesgo de aglomeración en la estación k a lo largo del tiempo.
w_k	Número total de pasajeros esperando en la estación k .
r_k	Riesgo de aglomeración total en la estación k .

$$\text{Min } \theta_1 \cdot \sum_{k \in S} w_k + \theta_2 \cdot \sum_{k \in S} r_k$$

Sujeto a:

- 1) $0 \leq n_{ik} \leq n_{m\acute{a}x}, \quad \forall i \in I, k \in S$
- 2) $n_{ik} \geq n_{i,k+1}, \quad \forall i \in I, k, k+1 \in S$

$$3) \quad v_{ik} = \begin{cases} b_{ik} & \text{si } k = 1 \\ v_{ik-1} - a_{ik} + b_{ik} & \text{en cualquier otro caso} \end{cases}, \quad \forall i \in I, k \in S$$

$$4) \quad w_{ik} = \begin{cases} \sum_{p \in \{p' \in P | o_{p'} = k, 0 \leq t_{p'} \leq d_{1,k}\}} n_p & \forall i = 1 \\ l_{i-1,k} + \sum_{p \in \{p' \in P | o_{p'} = k, d_{i-1,k} \leq t_{p'} \leq d_{i,k}\}} n_p & \forall i \neq 1 \end{cases}, \quad \forall i \in I, k \in S, p \in P$$

$$5) \quad w_{iks} = \begin{cases} \sum_{p \in \{p' \in P | o_{p'} = k, 0 \leq t_{p'} \leq d_{1,k}\}} n_p & \text{si } i = 1 \\ l_{i-1,k,s} + \sum_{p \in \{p' \in P | o_{p'} = k, d_{i-1,k} \leq t_{p'} \leq d_{i,k}\}} n_p & \text{si } i \neq 1 \end{cases}, \quad \forall i \in I, \forall k, s \in S, k < s$$

$$6) \quad c_{ik} = \begin{cases} (Q - n_{ik}) \times C & \text{si } k = 1 \\ (Q - n_{ik}) \times C - (v_{i,k-1} - a_{ik}), & \text{si } k \neq 1 \end{cases}, \quad \forall k \in S, \forall i \in I$$

$$7) \quad b_{ik} = \min\{c_{ik}, w_{ik}\}, \quad \forall i \in I, \forall k \in S$$

$$8) \quad l_{ik} = w_{ik} - b_{ik}, \quad \forall i \in I, k \in S$$

$$9) \quad b_{iks} = \frac{w_{iks}}{w_{ik}} \times b_{ik}, \quad \forall i \in I, k, s \in S, k < s$$

$$10) \quad l_{iks} = w_{iks} - b_{iks}, \quad \forall i \in I, \forall k, s \in S, k < s$$

$$11) \quad a_{ik} = \sum_{s=1}^{k-1} b_{isk}, \quad \forall i \in I, k \in S$$

$$12) \quad l_{|I|k} = 0, \quad \forall k \in S$$

$$13) \quad w_k(t) = \sum_{p \in \{p' \in P | o_{p'} = k, 0 \leq t_{p'} \leq t\}} n_p - \sum_{i \in \{i' \in I | d_{i',k} \leq t\}} b_{ik}, \quad \forall k \in S, t \in [0, T]$$

$$14) \quad r_k(t) = \begin{cases} 0, & \text{si } 0 \leq w_k(t) \leq U_k \\ \frac{w_k(t) - U_k}{H_k - U_k} \cdot \varepsilon, & \text{si } U_k < w_k(t) < H_k \\ M, & \text{si } w_k(t) \geq H_k \end{cases}, \quad \forall k \in S, t \in [0, T]$$

$$15) \quad r_k = \int_0^T r_k(t) dt, \quad \forall k \in S$$

$$16) \quad w_k = \int_0^T w_k(t) dt, \quad \forall k \in S$$

Este sería el modelo al completo que nos disponemos a resolver. Para desarrollar tanto las restricciones como la función objetivo, se ha usado *Python* y funciones propias de la librería de *GUROBI*, como *addVar* o *addConstr*.

Con la función *addVar* se podrá declarar todas las variables necesarias para resolver el problema. En cuanto a la función *addConstr*, nos permitirá desarrollar todas las [restricciones](#) anteriormente explicadas.

Los [datos previos](#) e información necesaria estarán recogidos en un archivo Excel. En este archivo se encontrarán tanto los horarios de los distintos trenes, como otra serie de datos relevantes para nuestro problema (número de vagones en cada tren, capacidad de cada vagón...)

Para poder usar estos datos dentro de nuestro modelo en Python, se usará la librería *openpyxl*, con la que se podrán leer todos los datos provenientes de un archivo Excel.

8 IMPLEMENTACIÓN EN PYTHON Y GUROBI

En primer lugar, se hablará de la lectura de datos provenientes de Excel. De este archivo Excel se leerán datos unitarios, en formato lista y por último en formato matriz.

Para ello se importarán 2 librerías:

```
from openpyxl import *  
from IOfunctionsExcel import *
```

Ilustración 6: Librerías usadas en Python

La segunda de ellas se trata de un código que nos permite leer datos de Excel en formato lista y matriz. Por ejemplo, para leer una matriz de Excel en Python se usaría la siguiente función:

```
def Read_Excel_to_NesteDic(sheet, Range1, Range2): # Los datos en la hoja de cálculo deben haber sido formateados primero  
    dict1={}  
    multiple_cells = sheet[Range1:Range2]  
    Aux={}  
    Aux.update({0: 'Empty'})  
  
    #Primero vamos a leer la fila que contiene las cabeceras de las columnas, que seran claves del diccionario interno  
    Column=0  
    for cell in multiple_cells[0]:  
        if Column >=1:  
            Aux.update({Column:cell.value})  
            Column=Column+1  
    # Ahora pasamos a leer por filas desde la primera  
  
    RowNumber=len(multiple_cells)  
    for Row in range(1,RowNumber):  
        dict2 = {}  
        Column=0  
        key = multiple_cells[Row][Column].value  
        for cell in multiple_cells[Row]:  
            if Column>=1:  
                dict2.update({Aux[Column]:cell.value})  
                Column=Column+1  
            dict1.update({key:dict2})  
  
    return dict1
```

Ilustración 7: IOFunctionsExcel en Python

Para después en el código principal usarse de la siguiente manera:

```
Horario=excel_document['Horario']
d=Read_Excel_to_NesteDic(Horario,'B2','J10')
```

Ilustración 8:Uso de IOFunctionsExcel

Donde d será d_{ik} y la matriz B2:J10 de la hoja Horario será el horario que siguen estos trenes en las distintas estaciones:

	A	B	C	D	E	F	G	H	I	J
1	Horario	Trenes								
2		1	2	3	4	5	6	7	8	
3	Estaciones	1	9.00	13.00	19.00	26.00	29.00	32.00	38.00	43.00
4		2	11.68	15.68	21.68	26.68	30.68	34.68	38.68	45.68
5		3	12.24	17.24	23.24	30.24	34.24	36.24	41.24	47.24
6		4	14.84	18.84	24.84	31.84	35.84	38.84	42.84	50.84
7		5	17.40	20.40	27.40	33.40	36.40	42.40	43.40	52.40
8		6	18.89	22.89	29.89	35.89	39.89	43.89	45.89	54.89
9		7	21.23	25.23	33.23	38.23	40.23	46.23	47.23	58.23
10		8	22.75	25.75	33.75	40.75	43.75	47.75	49.75	60.75

Tabla 3: Horario del servicio

Una vez todos los datos necesarios han sido llamados, se declaran las variables de decisión e intermedias del modelo. Para ello se usará la fórmula addvar:

```

nreserved={} #Numero de vagones reservados en tren i que sale de la estacion k
for i in Tren:
    for k in Estacion:
        nreserved[i,k]=tr.addVar(lb=0.0,ub=nmax, vtype=GRB.INTEGER, name="nreserved_%s_%s" % (i,k))

clibre={} #capacidad restante para un tren i que llega a la est k <----- OJO!!!!
#Según la restriccion 6 la capacidad es a la llegada np a la salida!!!!!!

for i in Tren:
    for k in Estacion:
        clibre[i,k]=tr.addVar(lb=0.0,ub=GRB.INFINITY, vtype=GRB.CONTINUOUS, name="clibre_%s_%s" % (i,k))

waiting={} #numero de pasajeros esperando a entrar al tren i en la estacion k
for i in Tren:
    for k in Estacion:
        waiting[i,k]=tr.addVar(lb=0.0,ub=GRB.INFINITY, vtype=GRB.CONTINUOUS, name="waiting_%s_%s" % (i,k))

```

Ilustración 9: Variables declaradas en Python

Para construir las restricciones de nuestro problema se usará `addConstr`. A continuación, se mostrará cómo se han desarrollado las 16 restricciones:

$$1) \quad 0 \leq n_{ik} \leq n_{\max}, \forall i \in I, k \in S$$

```

nreserved={} #Numero de vagones reservados en tren i que sale de la estacion k
for i in Tren:
    for k in Estacion:
        nreserved[i,k]=tr.addVar(lb=0.0,ub=nmax, vtype=GRB.INTEGER, name="nreserved_%s_%s" % (i,k))

```

Ilustración 10: Restricciones 1 en Python

En este caso no haría falta desarrollar la Restricciones pues se puede especificar el límite superior cuando se añade la variable con `ub=nmax`.

$$2) \quad n_{ik} \geq n_{i,k+1}, \forall i \in I, k, k+1 \in S$$

```

for i in Tren:
    for k in Estacion:
        if k>1:
            tr.addConstr(nreserved[i,k]<=nreserved[i,k-1], name="Restriccion2_%s_%s" % (i,k))

```

Ilustración 11: Restricciones 2 en Python

$$3) v_{ik} = \begin{cases} b_{ik} & \text{si } k = 1 \\ v_{i,k-1} - a_{ik} + b_{ik} & \text{en cualquier otro caso} \end{cases}, \quad \forall i \in I, k \in S$$

```
for i in Tren:
    for k in Estacion:
        if k==1:
            tr.addConstr(onboard[i,k]==boarding[i,k],name="Restriccion3_%s_%s" %(i,k))
        else:
            tr.addConstr(onboard[i,k]==(onboard[i,k-1]+boarding[i,k]-alight[i,k]),name="Restriccion3_%s_%s" %(i,k))
```

Ilustración 12: Restricciones 3 en Python

$$4) w_{ik} = \begin{cases} \sum_{p \in \{p' \in P | o_{p'} = k, 0 \leq t_{p'} \leq d_{1k}\}} n_p & \forall i = 1 \\ l_{i-1,k} + \sum_{p \in \{p' \in P | o_{p'} = k, d_{i-1,k} \leq t_{p'} \leq d_{ik}\}} n_p & \forall i \neq 1 \end{cases} \quad \forall i \in I, k \in S, p \in P$$

$$5) w_{iks} = \begin{cases} \sum_{p \in \{p' \in P | o_{p'} = k, 0 \leq t_{p'} \leq d_{1k}\}} n_p & \text{si } i = 1 \\ l_{i-1,k,s} + \sum_{p \in \{p' \in P | o_{p'} = k, d_{i-1,k} \leq t_{p'} \leq d_{ik}\}} n_p & \text{si } i \neq 1 \end{cases} \quad \forall i \in I, \forall k, s \in S, k < s$$

```
llegadas = numpy.zeros((Tren[-1], Estacion[-1], Estacion[-1]))

for i in Tren:
    for k in Estacion:
        for s in Estacion:
            if s > k:
                if i == 1:
                    llegadas[i - 1][k - 1][s - 1] = 0
                    for p in Pasajeros:
                        if k == origenp[p - 1] and s == destinop[p - 1] and 0 <= llegadap[p - 1] and llegadap[p - 1] <= d[k][i]:
                            llegadas[i - 1][k - 1][s - 1] = llegadas[i - 1][k - 1][s - 1] + numerop[p - 1]
                else:
                    llegadas[i - 1][k - 1][s - 1] = 0
                    for p in Pasajeros:
                        if k == origenp[p - 1] and s == destinop[p - 1] and d[k][i - 1] <= llegadap[p - 1] and llegadap[p - 1] <= d[k][i]:
                            llegadas[i - 1][k - 1][s - 1] = llegadas[i - 1][k - 1][s - 1] + numerop[p - 1]

for i in Tren:
    for k in Estacion:
        for s in Estacion:
            if s > k:
                if i == 1:
                    tr.addConstr(Waiting_[i, k, s] == llegadas[i - 1][k - 1][s - 1], name="Restriccion5_%s_%s_%s" % (i, k, s))
                else:
                    tr.addConstr(Waiting_[i, k, s] == Left_[i - 1, k, s] + llegadas[i - 1][k - 1][s - 1],
                                name="Restriccion5_%s_%s_%s" % (i, k, s))
```

Ilustración 13: Restricciones 5 en Python

Como se puede apreciar en la imagen superior, para programar la 5ª restricción se debe crear una variable numpy llegadas. Esta variable será una matriz tridimensional de tamaño IxSxS.

En cada una de las posiciones de esta matriz se guarda el número de pasajeros con dirección estación s que llegan a una estación k desde que sale el tren anterior $i-1$ hasta que llega el tren i . En el caso de ser el primer tren, este sería igual a w_{1ks} y a partir del segundo tren se tendría que sumar, además, los pasajeros que no pueden montarse en el tren anterior.

También se aclara la relación entre w_{iks} y w_{ik} así como con b y l . Esto hace que no sea necesario desarrollar la restricción 4, pues ya quedaría desarrollada en la restricción 5.

```
Boarding_={} # BOARDING
for i in Tren:
    for k in Estacion:
        for s in Estacion:
            if s > k:
                Boarding_[i,k,s]=tr.addVar(lb=0.0,ub=GRB.INFINITY, vtype=GRB.CONTINUOUS, name="boardin3_%s_%s_%s" % (i,k,s))

Waiting_={} # WAITING
for i in Tren:
    for k in Estacion:
        for s in Estacion:
            if s > k:
                Waiting_[i,k,s]=tr.addVar(lb=0.0,ub=GRB.INFINITY, vtype=GRB.CONTINUOUS, name="waiting3_%s_%s_%s" % (i,k,s))

Left_={} #LEFT BEHIND
for i in Tren:
    for k in Estacion:
        for s in Estacion:
            if s > k:
                Left_[i,k,s]=tr.addVar(lb=0.0,ub=GRB.INFINITY, vtype=GRB.CONTINUOUS, name="Left3_%s_%s_%s" % (i,k,s))
```

Ilustración 14:Relación entre i, k, s y i, k en Python

$$6) \quad c_{ik} = \begin{cases} (Q - n_{ik}) \times C & \text{si } k = 1 \\ (Q - n_{ik}) \times C - (v_{i,k-1} - a_{ik}), & \text{si } k \neq 1 \end{cases} \quad \forall k \in S, \forall i \in I$$

```
for i in Tren:
    for k in Estacion:
        if k == 1:
            tr.addConstr(clibre[i, k] == (QVag - nreserved[i, k]) * CAP, name="Restriccion6_%s_%s" % (i, k))
        else:
            tr.addConstr(clibre[i, k] == ((QVag - nreserved[i, k]) * CAP - (onboard[i, k - 1] - alight[i, k])), name="Restriccion6_%s_%s" % (i, k))
```

Ilustración 15:Restricciones 6 en Python

$$7) b_{ik} = \min\{c_{ik}, w_{ik}\}, \quad \forall i \in I, \forall k \in S$$

En este caso, GUROBI tiene una función `min_` que permite escribir la restricción 7 de forma muy sencilla. De no haber sido así, la restricción hubiera sido no lineal y habría que desarrollarla de forma diferente.

```
for i in Tren:
    for k in Estacion:
        tr.addConstr(boarding[i, k] == min_(clicbre[i, k], waiting[i, k]), name="Restriccion7_%s_%s" % (i, k))
```

Ilustración 16:Restricciones 7 en Python

$$8) l_{ik} = w_{ik} - b_{ik}, \quad \forall i \in I, k \in S$$

```
for i in Tren:
    for k in Estacion:
        tr.addConstr(l[i, k] == w[i, k] - b[i, k], name="Restriccion8_%s_%s" % (i, k))
```

Ilustración 17:Restricciones 8 en Python

$$9) b_{iks} = \frac{w_{iks}}{w_{ik}} \times b_{ik}, \quad \forall i \in I, k, s \in S, k < s$$

Esta restricción, como se ha comentado anteriormente, sería no lineal y deberá ser desarrollada de forma diferente. Habrá que crear una variable Gurobi llamada X con un valor entre 0 y 1.

$$\left\{ \begin{array}{l} \sum_{k+1}^S X_{iks} = 1 \quad \forall i \in I, \forall k \in S \\ \sum_{k+1}^S Y_{iks} = 1 \quad \forall i \in I, \forall k \in S \\ w_{iks} = X_{iks} \cdot w_{ik}, \quad i \in I, k, s \in S, s > k \\ b_{iks} = Y_{iks} \cdot b_{ik}, \quad i \in I, k, s \in S, s > k \end{array} \right.$$

```

Xporcen = {}

for i in Tren:
    for k in Estacion:
        if k < len(Estacion):
            for s in Estacion:
                if s > k:
                    Xporcen[i, k, s] = tr.addVar(lb=0.0, ub=1.0, vtype=GRB.CONTINUOUS, name="Xporcen_%s_%s_%s" % (i, k, s))

for i in Tren:
    for k in Estacion:
        if k < len(Estacion):
            tr.addConstr(quicksum(Xporcen[i, k, s] for s in Estacion if s > k) == 1, name="Restriccion9sumadeX2_%s_%s" % (i, k))

for i in Tren:
    for k in Estacion:
        for s in Estacion:
            if k < s:
                tr.addConstr(Waiting_[i, k, s] == Xporcen[i, k, s] * waiting[i, k], name="Restriccion9conW_%s_%s_%s" % (i, k, s))

for i in Tren:
    for k in Estacion:
        for s in Estacion:
            if k < s:
                tr.addConstr(Boarding_[i, k, s] == Xporcen[i, k, s] * boarding[i, k], name="Restriccion9conB_%s_%s_%s" % (i, k, s))

```

Ilustración 18:Restricciones 9 en Python

$$10) l_{iks} = w_{iks} - b_{iks}, \quad \forall i \in I, \forall k, s \in S, k < s$$

```

for i in Tren:
    for k in Estacion:
        for s in Estacion:
            if k < s:
                tr.addConstr(Left_[i, k, s] == Waiting_[i, k, s] - Boarding_[i, k, s], name="Restriccion10_%s_%s_%s" % (i, k, s))

```

Ilustración 19::Restricciones 10 en Python

$$11) a_{ik} = \sum_{s=1}^{k-1} b_{isk}, \quad \forall i \in I, k \in S$$

```

for i in Tren:
    for k in Estacion:
        tr.addConstr(alight[i, k] == quicksum(Boarding_[i, s, k] for s in range(1, k)), name="Restriccion11_%s_%s" % (i, k))

```

Ilustración 20:Restricciones 11 en Python

$$12) l_{|k} = 0, \quad \forall k \in S$$

Esta restricción sólo se podrá aplicar si hay capacidad suficiente para suplir a toda la demanda. Por lo tanto, dependiendo del caso se usará o no.

```
for k in Estacion:
    tr.addConstr(1[Tren[-1], k] == 0, name="Restriccion12_%s" % (k))
```

Ilustración 21: Restricciones 12 en Python

$$13) w_k(t) = \sum_{p \in \{p' \in P \mid o_{p'} = k, 0 \leq t_{p'} \leq t\}} n_p - \sum_{i \in \{i' \in I \mid d_{i'} \leq t\}} b_{ik}, \quad \forall k \in S, t \in [0, T]$$

Para la restricción número 13 se crearán 2 conjuntos de variables paralelas: Sum_Pas_until_t y boarding_until_t.

La primera de ellas dará la suma de todos los pasajeros que han pasado por una estación k hasta el momento t. Boarding_until_t por el contrario muestra todos los que han podido subirse a un tren en una estación hasta el momento t. La diferencia entre ambas nos dará el número de pasajeros esperando en un instante t en la estación k.

```
"""R13: Queremos ver el número de pasajeros esperando en una cierta estación a lo largo del tiempo."""
waiting_Tiempo = {}
for k in Estacion[:-1]:
    for t in Tiempo:
        waiting_Tiempo[k, t] = tr.addVar(lb=0, ub=GRB.INFINITY, vtype=GRB.CONTINUOUS, name="wait_tiempo_%s(%s)" % (k, t))

boarding_until_t = {}
for k in Estacion[:-1]:
    for t in Tiempo:
        boarding_until_t[k, t] = tr.addVar(lb=0.0, ub=GRB.INFINITY, vtype=GRB.CONTINUOUS, name="board_until_t_%s(%s)" % (k, t))

for k in Estacion[:-1]:
    for t in Tiempo:
        tr.addConstr(boarding_until_t[k, t] == quicksum(boarding[i, k] for i in Tren if d[k][i] < t), name="boarding_until_t_%s_%s" % (k, t))

Sum_Pas_until_t = {}
for k in Estacion[:-1]:
    for t in Tiempo:
        Sum_Pas_until_t[k, t] = 0
        for p in Pasajeros:
            if origenp[p - 1] == k and llegadap[p - 1] <= t:
                Sum_Pas_until_t[k, t] = Sum_Pas_until_t[k, t] + numerop[p - 1]*Escala

print(Sum_Pas_until_t)

for k in Estacion[:-1]:
    for t in Tiempo:
        tr.addConstr(waiting_Tiempo[k, t] == Sum_Pas_until_t[k, t] - boarding_until_t[k, t], name="Waiting_with_time_R13_%s_%s" % (k, t))
```

Ilustración 22: Restricciones 13 en Python

$$14) r_k(t) = \begin{cases} 0, & \text{si } 0 \leq w_k(t) \leq U_k \\ \frac{w_k(t) - U_k}{H_k - U_k} \cdot \varepsilon, & \text{si } U_k < w_k(t) < H_k, \\ M, & \text{si } w_k(t) \geq H_k \end{cases} \quad \forall k \in S, t \in [0, T]$$

La restricción 14, como la restricción 9, será no lineal. En este caso habrá que crear 6 nuevas variables binarias:

```
for k in Estacion:
    for t in Tiempo:
        delta1[k, t] = tr.addVar(vtype=GRB.BINARY, name="delta1_%s(%s)" % (k, t))
        delta2[k, t] = tr.addVar(vtype=GRB.BINARY, name="delta2_%s(%s)" % (k, t))
        delta3[k, t] = tr.addVar(vtype=GRB.BINARY, name="delta3_%s(%s)" % (k, t))

for k in Estacion:
    for t in Tiempo:
        w2[k, t] = tr.addVar(lb=0.0, ub=GRB.INFINITY, vtype=GRB.CONTINUOUS, name="w%s_1(%s)" % (k, t))
        w1[k, t] = tr.addVar(lb=0.0, ub=GRB.INFINITY, vtype=GRB.CONTINUOUS, name="w%s_2(%s)" % (k, t))
        w3[k, t] = tr.addVar(lb=0.0, ub=GRB.INFINITY, vtype=GRB.CONTINUOUS, name="w%s_3(%s)" % (k, t))
```

Ilustración 23: Variables previas en Restricciones 14 en Python

$$\begin{cases} \delta_{1k}(t) + \delta_{2k}(t) + \delta_{3k}(t) = 1 & \forall k, t \in S, T \\ w_{1k}(t) + w_{2k}(t) + w_{3k}(t) = w_k(t) & \forall k, t \in S, T \\ w_{1k}(t) \geq 0 & \forall k, t \in S, T \\ \delta_{2k}(t) \cdot U_k \leq w_{2k}(t) \leq \delta_{2k}(t) \cdot H_k & \forall k, t \in S, T \\ M \cdot \delta_{3k}(t) \leq w_{3k}(t) \leq 2 \cdot M \cdot \delta_{3k}(t) & \forall k, t \in S, T \end{cases}$$

```
M=10**6
for k in Estacion[:-1]:
    for t in Tiempo:
        tr.addConstr((delta1[k, t] + delta2[k, t] + delta3[k, t]) == 1, name="Suma_deltas_%s_%s" % (k, t))
        tr.addConstr(w1[k, t] + w2[k, t] + w3[k, t] == w[k, t], name="waiting_as_sum_%s_%s" % (k, t))
        tr.addConstr(0 <= w1[k, t], name="w1_lower_%s_%s" % (k, t))
        tr.addConstr(w1[k, t] <= Uk[k - 1] * delta1[k, t], name="w1_upper_%s_%s" % (k, t))
        tr.addConstr(delta2[k, t] * Uk[k - 1] <= w2[k, t], name="w2_lower_%s_%s" % (k, t))
        tr.addConstr(w2[k, t] <= Hk[k - 1] * delta2[k, t], name="w2_upper_%s_%s" % (k, t))
        tr.addConstr(Hk[k - 1] * delta3[k, t] <= w3[k, t], name="w3_lower_%s_%s" % (k, t))
        tr.addConstr(w3[k, t] <= M * delta3[k, t], name="w3_upper_%s_%s" % (k, t))

for k in Estacion[:-1]:
    for t in Tiempo:
        tr.addConstr(R[k, t] == 0 + 100 * (w2[k, t] - Uk[k - 1] * delta2[k, t]) / (Hk[k - 1] - Uk[k - 1]) + epsilon * delta3[k, t], name="Restriccion14_%s_%s" % (k, t))
```

Ilustración 24: Restricciones 14 en Python

$$r_k(t) = 0 + \frac{w_{2k}(t) - U_k \cdot \delta_{2k}(t)}{H_k - U_k} \cdot \varepsilon + M \cdot \delta_{3k}(t) \quad \forall k \in S, \forall t \in T$$

$$15) r_k = \int_0^T r_k(t)dt, \quad \forall k \in S$$

$$16) w_k = \int_0^T w_k(t)dt, \quad \forall k \in S$$

En cuanto a las restricciones 15 y 16, en Python bastará con hacer el sumatorio a lo largo del tiempo. Pese a no ser esto la integral exacta, cumple la misma misión.

```
Riesgo = {}
Espera = {}
for k in Estacion[:-1]:
    Riesgo[k] = tr.addVar(lb=0.0, ub=GRB.INFINITY, vtype=GRB.CONTINUOUS, name="R_%s" % (k))
    Espera[k] = tr.addVar(lb=0.0, ub=GRB.INFINITY, vtype=GRB.INTEGER, name="W_%s" % (k))

for k in Estacion[:-1]:
    tr.addConstr(Riesgo[k] == quicksum(R[k, t] for t in Tiempo), name="Restriccion15_%s" % (k))
    tr.addConstr(Espera[k] == quicksum(waiting_Tiempo[k, t] for t in Tiempo), name="Restriccion16_%s" % (k))
```

Ilustración 25:Restricciones 15 y 16 en Python

17) Función Objetivo

```
tr.setObjective(Theta1*(quicksum(Espera[k] for k in Estacion[:-1]))+Theta2*(quicksum(Riesgo[k] for k in Estacion[:-1])))
tr.modelSense=GRB.MINIMIZE
tr.optimize()
```

Ilustración 26:Función objetivo en Python

9 CASO DE ESTUDIO

Habr  ciertos valores como la demanda o la capacidad de los trenes que podr n ir variando para experimentar con distintos escenarios.

Para poder crear un escenario que se pueda aplicar a la realidad se deben observar las distintas estaciones de metro del mundo, su flujo de pasajeros diario y la capacidad tanto de estas estaciones como de los trenes que las recorren.

Por ejemplo, en Londres la capacidad media de cada tren est  alrededor de 1000 pasajeros (Transport For London, 2017). El flujo de pasajeros, sin embargo, var  mucho de una estaci n a otra pudiendo superar los 100.000 pasajeros diarios en Camden, o no llegar a 10.000 pasajeros en zonas menos urbanizadas como Wimbledon.

Habr  que considerar tambi n, mirando al ejemplo de Londres, que el sistema de metro tiene 270 estaciones y en hora punta puede llegar a tener 573 trenes funcionando. Esto quiere decir que habr  ocasiones en la que el n mero de trenes sea el doble que el de estaciones. (Telegraph UK, 2017)

Con el objetivo de poder simplificar el modelo, todas las estaciones consideradas ser n del mismo tama o. Esto implica tambi n que tanto la capacidad como los valores de riesgo de cada estaci n ser n similares.

En este caso, el modelo siempre estar  formado por 8 estaciones y 8 trenes. El horizonte de tiempo estar  fijado en 1 hora. Durante ese tiempo todos los trenes habr n pasado por las 8 estaciones fijadas.

		TIMETABLE									
E s t a c i o n e s	Salida	8	23.75	26.75	32.75	41.75	48.75	51.75	57.75	56.75	
	Llegada	8	22.75	25.75	31.75	40.75	46.75	50.75	55.75	55.75	
	Salida	7	22.225	25.225	31.225	40.225	46.225	50.225	55.225	55.225	
	Llegada	7	20.225	24.225	30.225	38.225	44.225	48.225	53.225	54.225	
	Salida	6	19.8875	23.8875	29.8875	37.8875	43.8875	47.8875	52.8875	53.8875	
	Llegada	6	17.8875	21.8875	27.8875	35.8875	41.8875	46.8875	50.8875	52.8875	
	Salida	5	17.4	21.4	27.4	35.4	41.4	46.4	50.4	52.4	
	Llegada	5	16.4	20.4	26.4	34.4	39.4	44.4	49.4	51.4	
	Salida	4	15.8375	19.8375	25.8375	33.8375	38.8375	43.8375	48.8375	50.8375	
	Llegada	4	14.8375	18.8375	24.8375	31.8375	36.8375	41.8375	46.8375	48.8375	
	Salida	3	14.2375	18.2375	24.2375	31.2375	36.2375	41.2375	46.2375	48.2375	
	Llegada	3	12.2375	17.2375	22.2375	29.2375	34.2375	40.2375	44.2375	47.2375	
	Salida	2	11.675	16.675	21.675	28.675	33.675	39.675	43.675	46.675	
	Llegada	2	9.675	15.675	20.675	27.675	32.675	38.675	41.675	44.675	
	Salida	1	9	15	20	27	32	38	41	44	
	Llegada	1	8	13	19	25	30	37	40	43	
			1	2	3	4	5	6	7	8	
			Servicios								

Tabla 4: Ejemplo de horarios

Se experimentará únicamente en una dirección, desde la estación 1 hasta la estación 8, ya que simplificará mucho los cálculos y añadir las direcciones de ida y vuelta no aporta ningún valor al modelo.

Esto es debido a que en una estación de metro bidireccional la demanda que cubre cada tren es unidireccional. Por lo tanto, la dirección contraria en este modelo se resolvería de la misma manera.

En el caso de estudio que nos encontramos también es muy importante la relación entre capacidad y demanda. La restricción 12 implica que la capacidad final sea mayor que la demanda, para así poder satisfacer esta demanda en su totalidad.

La demanda se dividirá por minutos, estación de origen y destino. Es decir, habrá un un número de pasajeros por cada minuto, origen y destino como se muestra en la siguiente tabla:

		tiempo (minutos)																																			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34		
1	2	3.00	1.00	2.00	3.00	2.00	5.00	3.00	1.00	2.00	9.00	5.00	3.00	2.00	2.00	5.00	3.00	10.00	15.00	5.00	3.00	22.00	10.00	4.00	25.00	20.00	15.00	14.00	12.00	4.00	2.00	5.00	20.00	20.00	14.00		
1	3	1.00	1.00	4.00	2.00	6.00	3.00	8.00	1.00	7.00	4.00	8.00	6.00	5.00	6.00	8.00	6.00	1.00	5.00	14.00	31.00	33.00	42.00	37.00	7.00	31.00	18.00	6.00	5.00	5.00	26.00	18.00	24.00	26.00			
1	4	5.00	3.00	5.00	10.00	11.00	12.00	15.00	27.00	23.00	24.00	12.00	10.00	10.00	13.00	1.00	7.00	19.00	19.00	7.00	65.00	49.00	21.00	13.00	50.00	33.00	1.00	7.00	5.00	39.00	35.00	2.00	15.00	33.00	55.00		
1	5	7.00	2.00	6.00	8.00	3.00	21.00	12.00	21.00	20.00	19.00	4.00	35.00	30.00	29.00	21.00	5.00	2.00	23.00	38.00	35.00	32.00	19.00	38.00	28.00	1.00	105.00	10.00	8.00	70.00	68.00	7.00	15.00	83.00	31.00	24.00	
1	6	6.00	3.00	8.00	8.00	14.00	1.00	8.00	19.00	21.00	21.00	13.00	5.00	20.00	8.00	18.00	2.00	2.00	2.00	6.00	38.00	32.00	68.00	83.00	41.00	39.00	44.00	3.00	11.00	21.00	46.00	50.00	24.00	16.00	47.00		
1	7	2.00	6.00	4.00	4.00	8.00	10.00	5.00	12.00	9.00	7.00	15.00	14.00	10.00	14.00	16.00	1.00	4.00	18.00	1.00	36.00	35.00	38.00	38.00	34.00	56.00	60.00	6.00	28.00	17.00	37.00	35.00	1.00	9.00	11.00		
1	8	4.00	4.00	3.00	1.00	2.00	8.00	9.00	3.00	14.00	6.00	10.00	4.00	3.00	15.00	5.00	11.00	21.00	10.00	18.00	6.00	3.00	50.00	30.00	4.00	37.00	7.00	11.00	30.00	21.00	30.00	3.00	16.00	3.00	13.00		
2	3	1.00	2.00	3.00	1.00	3.00	4.00	2.00	3.00	6.00	7.00	5.00	4.00	8.00	7.00	7.00	1.00	3.00	13.00	3.00	22.00	8.00	11.00	23.00	7.00	27.00	20.00	1.00	15.00	2.00	8.00	6.00	6.00	10.00	11.00		
2	4	2.00	6.00	1.00	1.00	1.00	10.00	1.00	3.00	12.00	3.00	5.00	6.00	5.00	9.00	10.00	28.00	3.00	9.00	36.00	3.00	40.00	58.00	43.00	19.00	60.00	21.00	8.00	45.00	28.00	42.00	19.00	24.00	30.00			
2	5	3.00	9.00	10.00	5.00	3.00	9.00	13.00	19.00	7.00	10.00	6.00	7.00	5.00	9.00	21.00	20.00	19.00	28.00	37.00	34.00	17.00	8.00	34.00	43.00	5.00	33.00	68.00	89.00	32.00	17.00	18.00	31.00	64.00	8.00		
2	6	6.00	7.00	4.00	8.00	11.00	15.00	7.00	20.00	6.00	14.00	3.00	14.00	3.00	21.00	4.00	11.00	31.00	8.00	35.00	10.00	8.00	34.00	86.00	83.00	16.00	4.00	12.00	28.00	4.00	15.00	40.00	26.00	37.00	52.00		
2	7	2.7	4.00	1.00	3.00	1.00	7.00	5.00	2.00	2.00	4.00	5.00	3.00	9.00	5.00	2.00	15.00	7.00	5.00	11.00	4.00	7.00	32.00	7.00	21.00	28.00	26.00	20.00	27.00	33.00	12.00	4.00	24.00	3.00	10.00	3.00	
2	8	2.8	2.00	2.00	1.00	2.00	2.00	5.00	5.00	3.00	7.00	8.00	9.00	3.00	6.00	8.00	5.00	8.00	6.00	5.00	14.00	18.00	17.00	23.00	23.00	29.00	6.00	24.00	12.00	16.00	3.00	9.00	17.00	1.00	2.00	16.00	
3	4	3.4	1.00	1.00	2.00	1.00	5.00	2.00	5.00	1.00	1.00	6.00	5.00	3.00	3.00	4.00	8.00	11.00	5.00	15.00	28.00	3.00	9.00	18.00	18.00	7.00	23.00	8.00	1.00	15.00	3.00	4.00	7.00	16.00	5.00		
3	5	3.5	4.00	4.00	6.00	4.00	4.00	3.00	9.00	7.00	15.00	13.00	6.00	8.00	15.00	11.00	19.00	10.00	20.00	24.00	10.00	25.00	29.00	21.00	53.00	52.00	60.00	40.00	41.00	2.00	20.00	48.00	8.00	30.00	38.00	24.00	
3	6	3.8	8.00	5.00	9.00	3.00	5.00	20.00	12.00	13.00	8.00	28.00	26.00	8.00	3.00	19.00	9.00	25.00	21.00	14.00	37.00	51.00	52.00	61.00	119.00	94.00	19.00	12.00	51.00	24.00	71.00	38.00	57.00	3.00	85.00	40.00	
3	7	2.7	2.00	1.00	2.00	5.00	3.00	2.00	8.00	3.00	6.00	4.00	11.00	3.00	6.00	7.00	6.00	3.00	12.00	10.00	23.00	28.00	9.00	8.00	35.00	20.00	17.00	22.00	5.00	15.00	27.00	38.00	15.00	7.00	33.00	2.00	
3	8	3.8	3.00	1.00	3.00	3.00	1.00	5.00	5.00	8.00	6.00	2.00	4.00	5.00	4.00	5.00	3.00	9.00	3.00	7.00	14.00	5.00	19.00	20.00	33.00	8.00	16.00	4.00	8.00	5.00	5.00	10.00	21.00	3.00	5.00	3.00	
4	5	4.5	3.00	1.00	3.00	2.00	1.00	6.00	5.00	3.00	10.00	7.00	7.00	2.00	4.00	4.00	8.00	6.00	3.00	16.00	16.00	30.00	25.00	4.00	20.00	38.00	23.00	36.00	29.00	24.00	14.00	14.00	27.00	25.00	17.00	5.00	
4	6	4.6	5.00	2.00	3.00	5.00	5.00	1.00	3.00	5.00	12.00	4.00	11.00	7.00	1.00	4.00	15.00	11.00	12.00	4.00	21.00	11.00	28.00	10.00	50.00	25.00	32.00	9.00	18.00	1.00	6.00	30.00	8.00	14.00	10.00	15.00	
4	7	4.7	3.00	1.00	1.00	4.00	5.00	8.00	4.00	6.00	1.00	5.00	4.00	12.00	3.00	3.00	2.00	6.00	8.00	18.00	12.00	39.00	9.00	14.00	23.00	22.00	25.00	36.00	20.00	28.00	12.00	11.00	7.00	20.00	5.00	20.00	
4	8	4.8	2.00	3.00	3.00	2.00	5.00	5.00	7.00	8.00	3.00	9.00	3.00	2.00	8.00	4.00	11.00	14.00	7.00	29.00	30.00	25.00	26.00	5.00	29.00	19.00	7.00	5.00	12.00	7.00	1.00	20.00	9.00	6.00			
5	6	5.6	3.00	2.00	1.00	3.00	5.00	4.00	2.00	1.00	2.00	2.00	1.00	3.00	4.00	3.00	6.00	2.00	11.00	14.00	14.00	17.00	28.00	22.00	12.00	18.00	3.00	17.00	7.00	2.00	15.00	21.00	4.00	18.00	21.00	4.00	
5	7	5.7	2.00	1.00	4.00	2.00	4.00	2.00	7.00	4.00	10.00	1.00	11.00	2.00	12.00	10.00	12.00	5.00	6.00	14.00	17.00	38.00	37.00	25.00	18.00	13.00	40.00	31.00	28.00	7.00	30.00	23.00	27.00	28.00	2.00		
5	8	5.8	1.00	3.00	3.00	1.00	4.00	3.00	6.00	4.00	3.00	9.00	1.00	2.00	8.00	9.00	7.00	8.00	3.00	10.00	5.00	16.00	21.00	28.00	10.00	22.00	19.00	2.00	18.00	8.00	21.00	18.00	20.00	21.00	1.00	6.00	
6	7	6.7	2.00	1.00	2.00	1.00	4.00	2.00	3.00	2.00	6.00	4.00	3.00	5.00	3.00	2.00	2.00	3.00	3.00	2.00	7.00	5.00	8.00	9.00	15.00	8.00	20.00	10.00	6.00	8.00	12.00	14.00	10.00	8.00	10.00	13.00	
6	8	6.8	1.00	1.00	1.00	1.00	1.00	1.00	2.00	2.00	2.00	2.00	2.00	1.00	2.00	2.00	3.00	2.00	3.00	1.00	3.00	5.00	2.00	9.00	4.00	1.00	8.00	10.00	3.00	5.00	6.00	3.00	5.00	4.00	3.00	5.00	7.00
7	8	7.8	1.00	1.00	1.00	1.00	1.00	1.00	2.00	3.00	2.00	3.00	3.00	2.00	3.00	1.00	1.00	3.00	5.00	5.00	2.00	2.00	2.00	8.00	8.00	11.00	9.00	1.00	3.00	2.00	8.00	8.00	5.00	2.00	7.00	4.00	

Tabla 5: Ejemplo de demanda

A partir de aquí se experimentará con distintos valores como el número máximo de vagones que pueden ser reservados, los rangos de riesgo de cada estación... buscando valores que den soluciones coherentes y que nos ayuden a valorar la efectividad del modelo.

Descripción y explicación de datos

Todos los datos que se usarán en el modelo provienen de un archivo Excel. En él se tendrán diversas hojas que mostrarán matrices como la que muestra los horarios de los trenes, como valores de los parámetros del modelo como $n_{m\acute{a}x}$ o el valor de ϵ .

En primer lugar, la demanda dependerá de un factor de escala. Habrá un número de pasajeros para cada minuto y trayecto (estación de origen y destino). Esta información estará reflejada en 1680 grupos de

pasajeros que formarán parte del conjunto P . Cada grupo de pasajeros se referirá a una hora, estación de origen y estación de destino determinada.

Grupo pasajeros	Numero de pasajeros	Hora de llegada	Estación de origen	Estación de destino
1	3.00	1.00	1.00	2.00
2	3.00	1.00	1.00	3.00
3	2.00	1.00	1.00	4.00
4	8.00	1.00	1.00	5.00
5	7.00	1.00	1.00	6.00
6	5.00	1.00	1.00	7.00
7	3.00	1.00	1.00	8.00
8	3.00	1.00	2.00	3.00
9	1.00	1.00	2.00	4.00

Tabla 6: Grupos de pasajeros p

Dentro de los experimentos a realizar, podremos variar este factor de escala simulando distintos flujos de pasajeros.

El horario definido por los instantes de salida de los trenes de cada estación d_{ik} será una matriz de 8×8 en la que cada elemento será la hora en la que parte un tren i de la estación k . Esta matriz tendrá valores que comprenderán entre 0 y 60 min, la hora que comprende el horizonte temporal del modelo propuesto.

		Trenes							
		1	2	3	4	5	6	7	8
Estaciones	1	9.00	13.00	19.00	26.00	29.00	32.00	38.00	43.00
	2	11.68	15.68	21.68	26.68	30.68	34.68	38.68	45.68
	3	12.24	17.24	23.24	30.24	34.24	36.24	41.24	47.24
	4	14.84	18.84	24.84	31.84	35.84	38.84	42.84	50.84
	5	17.40	20.40	27.40	33.40	36.40	42.40	43.40	52.40
	6	18.89	22.89	29.89	35.89	39.89	43.89	45.89	54.89
	7	21.23	25.23	33.23	38.23	40.23	46.23	47.23	58.23
	8	22.75	25.75	33.75	40.75	43.75	47.75	49.75	60.75

Tabla 7: Horario

Para poder desarrollar la restricción 14 será necesario decidir cuál va a ser la capacidad (H_k) de cada estación, así como el valor a partir del cual existe un cierto riesgo (U_k). Este valor puede ser considerado igual para todas las estaciones, con el fin de simplificar el modelo, o diferenciar entre unas estaciones y otras.

En Londres, siguiendo con el ejemplo anterior, hay estaciones por las que discurren diversas líneas como la estación Victoria, por la que pasan más de 150 millones de pasajeros al año. También hay estaciones mucho más limitadas como Kensington, cuyo flujo es de aproximadamente 2 millones de pasajeros al año.

Aun así, si nos referimos únicamente a la zona en la que se espera la llegada del subterráneo, la capacidad no suele variar mucho de una estación a otra.



Ilustración 27: Estación de metro en Madrid

Por esta razón, en los distintos experimentos tanto H como U se considerarán iguales para todas las estaciones.

Por último, habrá varios datos unitarios cuyo valor puede cambiar mucho el desarrollo del modelo. Estos datos estarán recogidos en una hoja llamada “DATOS” y podremos modelarlos para simular distintas situaciones.

DATOS PREFIJADOS	
nmax	4.00
Q	4
C	200
epsilon	0.50
Theta1	0.5
Theta2	0.5
M	1000000.00

Tabla 8: Datos prefijados

n_{max} (número máximo de vagones a reservar) y Q (número de vagones) estarán muy relacionados, ya que aumentar el número de vagones por tren dará la posibilidad de aumentar, como es lógico, el máximo número de vagones que se pueden reservar.

Otros valores como ε o θ dependerán del peso y la importancia que se le dé a parámetros muy importantes de la función objetivo y las restricciones. Dependiendo del objetivo del modelo estos parámetros tomarán unos valores u otros.

El modelo puede buscar simplemente disminuir el riesgo de aglomeración para mejorar la seguridad de las estaciones. Si, por el contrario, la motivación del problema es liberar las estaciones con el fin de cumplir distancias de seguridad, reducir tiempos de espera y aumentar la eficiencia de las estaciones de metro estos valores deberán ser mucho más ambiciosos.

10 EXPERIMENTACIÓN

A continuación, se modificarán los datos explicados anteriormente (demanda, capacidad de trenes...) buscando resultados que aporten valor y que nos ayuden a comprender la eficacia del sistema.

Demanda

Para decidir tanto la demanda como la capacidad de los trenes se observará la red de metro de una gran ciudad como Londres. Esta demanda será nula en el momento en que el último tren de nuestro modelo pase por cada estación k , pues suponemos que esa demanda será satisfecha por los trenes siguientes.

Origen	Destino	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
1	2	7.00	5.00	8.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	3	9.00	9.00	9.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	4	5.00	16.00	5.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	5	26.00	21.00	21.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	6	2.00	18.00	6.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	7	8.00	9.00	7.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	8	3.00	3.00	12.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	3	2.00	2.00	7.00	8.00	11.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	4	13.00	4.00	11.00	8.00	15.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	5	3.00	9.00	9.00	27.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	6	1.00	15.00	18.00	14.00	24.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	7	12.00	10.00	5.00	18.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	8	7.00	2.00	1.00	7.00	4.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	4	2.00	8.00	7.00	2.00	5.00	8.00	12.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	5	12.00	6.00	3.00	21.00	2.00	14.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	6	9.00	15.00	5.00	39.00	30.00	25.00	33.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	7	6.00	11.00	3.00	5.00	8.00	12.00	5.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	8	4.00	6.00	5.00	9.00	10.00	6.00	5.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	5	1.00	3.00	10.00	8.00	11.00	8.00	7.00	20.00	12.00	7.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	6	7.00	2.00	2.00	5.00	5.00	19.00	15.00	17.00	6.00	12.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	7	5.00	9.00	1.00	16.00	7.00	14.00	1.00	10.00	15.00	12.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	8	4.00	8.00	4.00	2.00	3.00	2.00	10.00	14.00	7.00	9.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	6	7.00	4.00	7.00	4.00	4.00	6.00	4.00	10.00	15.00	12.00	2.00	6.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	7	10.00	8.00	6.00	6.00	13.00	10.00	9.00	7.00	19.00	7.00	6.00	8.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	8	7.00	1.00	3.00	11.00	7.00	8.00	7.00	6.00	4.00	3.00	6.00	4.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	7	5.00	2.00	3.00	4.00	3.00	2.00	5.00	4.00	2.00	2.00	5.00	3.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
6	8	1.00	1.00	2.00	3.00	1.00	3.00	4.00	4.00	4.00	5.00	1.00	1.00	2.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
7	8	2.00	2.00	2.00	2.00	3.00	4.00	2.00	1.00	3.00	1.00	2.00	2.00	1.00	1.00	2.00	1.00	1.00	1.00	1.00	1.00

Ilustración 28: Demanda tras pasar servicio 8.

La cantidad de pasajeros estará condicionada por un factor de escala. A continuación, se estudiará cuánto debe valer este factor de escala para que el modelo se aproxime lo máximo posible a la realidad.

Tomando como referencia la estación 1, aproximadamente 7500 pasajeros parten desde la estación 1 durante los 60 minutos que se tienen de horizonte temporal con un factor de escala 1. Pero, ¿Es este valor lógico y proporcional al número de servicios de nuestro modelo? Para poder responder a esta pregunta se puede observar el flujo de pasajeros en diversas estaciones de Londres.

Según datos del ayuntamiento de Londres (referencia), la demanda diaria de pasajeros en sus estaciones varía desde 5.000 pasajeros en estaciones de extrarradio como Queens, hasta superar los 100.000 pasajeros en la estación de London Bridge.

Si se usa como referencia una estación con un alto flujo de pasajeros como esta última, la estación London Bridge tuvo un flujo de pasajeros diario de 113.606 en el año 2017. Este número dividido entre las 19h en las que el metro londinense está funcionando serían unos 6.000 pasajeros por hora.

No obstante, este valor debe ser modificado por varias razones:

- El modelo busca reducir los tiempos de espera en hora punta. El flujo de pasajeros en una ciudad es muy diferente a las 08:00am y por ejemplo, a las 22:00pm.
- El modelo contempla una sola línea de metro. Estaciones grandes, como la que se están estudiando, suelen albergar más de una línea de metro.

La estación London Bridge, por ejemplo, tiene un total de 16 andenes. Está conectado a 2 líneas de metro, Northern Line y Jubilee Line, y a otras tantas líneas de tren a nivel nacional. Dado que la gran mayoría de andenes estén destinados a viajes nacionales resulta difícil saber cuántos usuarios usan la línea de metro diariamente.

Por otro lado, se observa la estación King's Cross-Saint Pancras, en Camden (Transport For London, 2017). Esta estación tuvo un flujo de pasajeros diario de 149.150 personas, y alberga hasta 6 líneas de metro.

Este número de pasajeros dividido entre las 6 líneas de metro, y las 19 horas en las que el metro se encuentra activo nos da una media de 1308 pasajeros por hora y línea.

Aunque este número deba ser mucho mayor si se trata de una hora concurrida, se puede concluir con total seguridad que el factor de escala 1 de nuestro modelo es demasiado alto. Con un factor de escala de 0.6 la demanda en la estación 1 para una hora rondaría los 3000 pasajeros. Un valor que se acerca más a la realidad que se quiere representar.

Otro aspecto importante a decidir es la capacidad de cada tren. Para ello también el modelo de Londres servirá de gran ayuda. En Londres hay distintos tipos de trenes y todos ellos rondan las 1100 personas de capacidad. Considerando que esta capacidad es orientativa y que en numerosas ocasiones se supera, el máximo será 1200 personas. Esta capacidad estará dividida en 6 vagones de 200 pasajeros cada uno.

Definición de experimentos

Hay dos objetivos mínimos que deben conseguirse a través de la experimentación:

1. H_k , la capacidad de una estación k , nunca debe de ser superada. Esto dispararía el valor de la función objetivo y a nivel práctico, comprometería la seguridad de los pasajeros.
2. De forma ideal nos gustaría que toda la demanda fuera satisfecha. Sin embargo, esto dependerá de la relación entre capacidad y demanda total.

10.1.1 Primera experimentación

Factor de escala	0.6
Capacidad de vagón	200
Número de vagones	6
Capacidad de estación	1000
Umbral de seguridad de estación	400
Máximo número de vagones reservables	6
θ_1	0.5
θ_2	0.5
ϵ	100

Tabla 9: Datos iniciales para la 1ª experimentación

Para la primera experimentación los datos que se usarán serán los mencionados arriba. Serán datos estándar pues se le dará la misma importancia tanto al riesgo de aglomeración como a la espera (véase θ_1 y θ_2) y el factor de escala 0.6 dará una demanda que se acerca a la realidad que se quiere representar.

En este primer experimento no se podrán reservar vagones, simulando una situación en la que el modelo no surtiría efecto. De esta manera se comprobará el efecto de reservar vagones para una demanda con un factor de escala 0,6.

Los resultados para este caso son los siguientes:

$$\text{Valor Función Objetivo: } \theta_1 \cdot \sum_{k \in S} w_k + \theta_2 \cdot \sum_{k \in S} r_k = 22.349$$

Este valor tan alto se debe a que el valor de riesgo de aglomeración máximo se ha superado en varias ocasiones.

RIESGO DE AGLOMERACIÓN	Estación 1	Estación 2	Estación 3	Estación 4	Estación 5	Estación 6	Estación 7	Estación 8
Tren 1	0	0	0	0	0	0	0	0
Tren 2	0	0	0	0	0	0	0	0
Tren 3	0	14	25	42	8	0	0	0
Tren 4	96	36	45	94	3	0	0	0
Tren 5	20	45	72	200	0	0	0	0
Tren 6	2	0	71	83	0	0	0	0
Tren 7	35	0	12	65	0	0	0	0
Tren 8	13	6	18	200	3	0	0	0

Tabla 10:Riesgo de aglomeración en 1ª experimentación

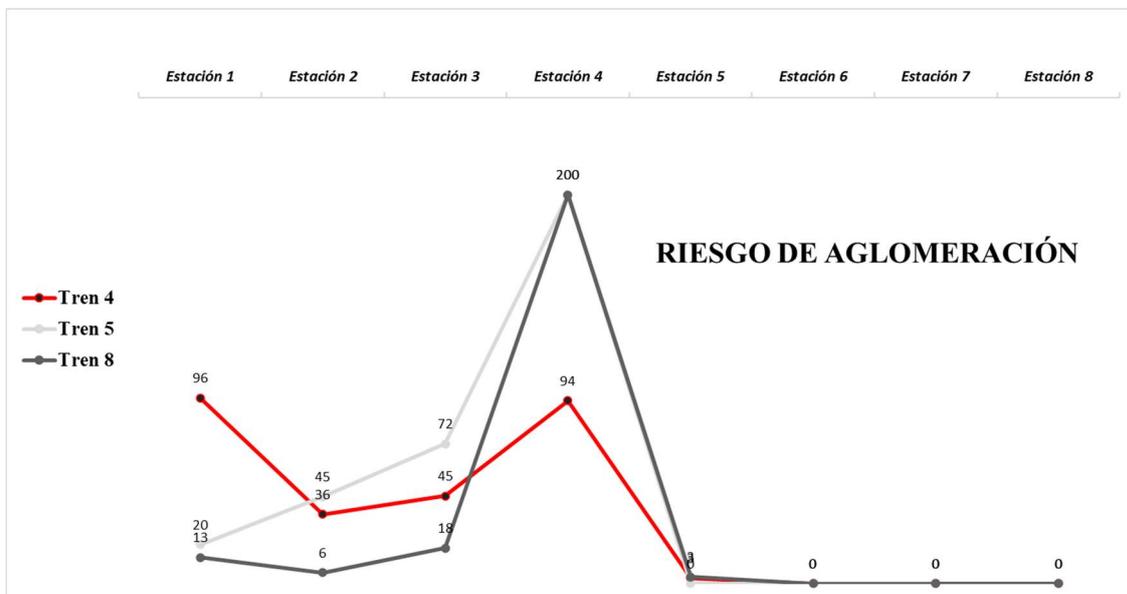


Gráfico 4:Riesgo de aglomeración en 1ª experimentación

Cuando el número de pasajeros esperando supera el valor de 1000 personas, que es el valor elegido como riesgo máximo, el riesgo de aglomeración toma el valor de $M=200$. Para un factor de escala de 0,6 la acumulación de pasajeros es la siguiente:

PASAJEROS ESPERANDO	Estación 1	Estación 2	Estación 3	Estación 4	Estación 5	Estación 6	Estación 7	Estación 8
Tren 1	241	295	328	296	270	275	32	0
Tren 2	193	161	176	173	168	158	8	0
Tren 3	345	482	549	653	447	321	19	0
Tren 4	973	614	670	964	416	146	13	0
Tren 5	520	668	831	1014	121	147	4	0
Tren 6	409	252	827	898	227	128	16	0
Tren 7	611	296	471	790	35	117	1	0
Tren 8	480	433	509	1226	420	286	20	0

Tabla 11:Flujo de pasajeros en 1ª experimentación

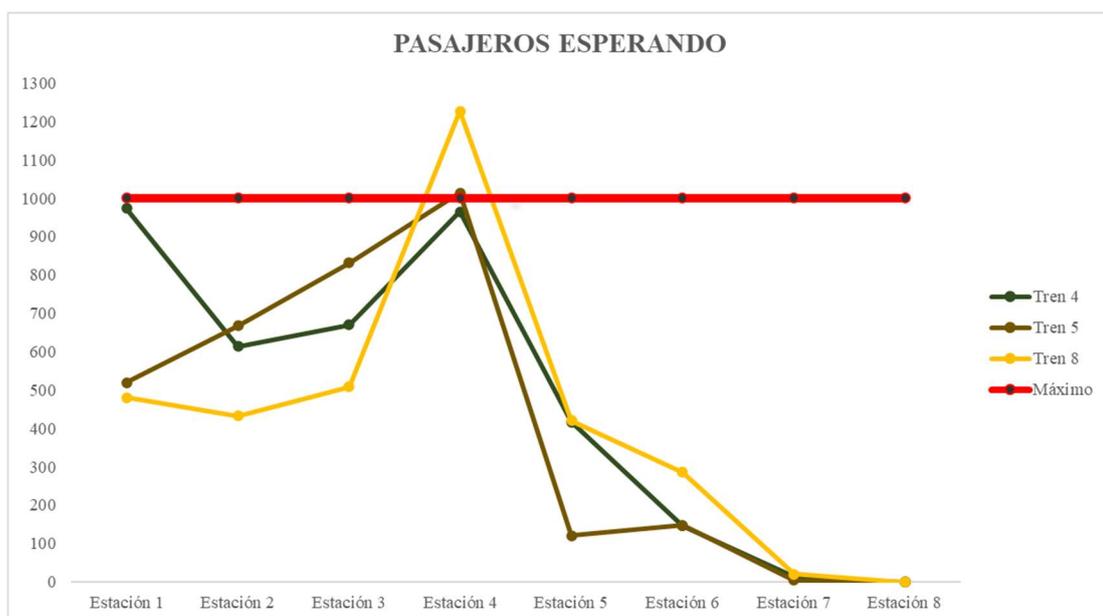


Gráfico 5 :Flujo de pasajeros en 1ª experimentación

Como se puede imaginar por el valor que adquiere la función objetivo, no sobrepasar este umbral máximo es una prioridad para el modelo. En el siguiente experimento se comprobará si la reserva de vagones consigue paliar esta sobrecarga de pasajeros y disminuir significativamente la función objetivo.

10.1.2 Segunda experimentación

En este segundo experimento el único dato inicial que variará será el máximo número de vagones reservables. De este modo podrá comprobarse si, para una demanda con un factor de escala 0.6, la reserva de vagones aporta valor y es capaz de descongestionar las estaciones del modelo.

Factor de escala	0.6
Capacidad de vagón	200
Número de vagones	6
Capacidad de estación	1000
Umbral de seguridad de estación	400
Máximo número de vagones reservables	6
θ_1	0.5
θ_2	0.5
ϵ	100

Tabla 12: Datos iniciales en 2ª experimentación

Los resultados obtenidos son los siguientes:

$$\text{Valor Función Objetivo: } \theta_1 \cdot \sum_{k \in S} w_k + \theta_2 \cdot \sum_{k \in S} r_k = 22.331$$

Lo que, comparado con el valor de la función objetivo para las mismas condiciones, pero sin la posibilidad de reservar vagones supone una mejora en 18 puntos. Esta mejora se debe en gran medida a que el umbral de riesgo máximo no se ha superado en esta ocasión.

Comparando el número de pasajeros esperando para ambos casos (con y sin reserva de vagones) se puede apreciar el efecto de esta reserva:

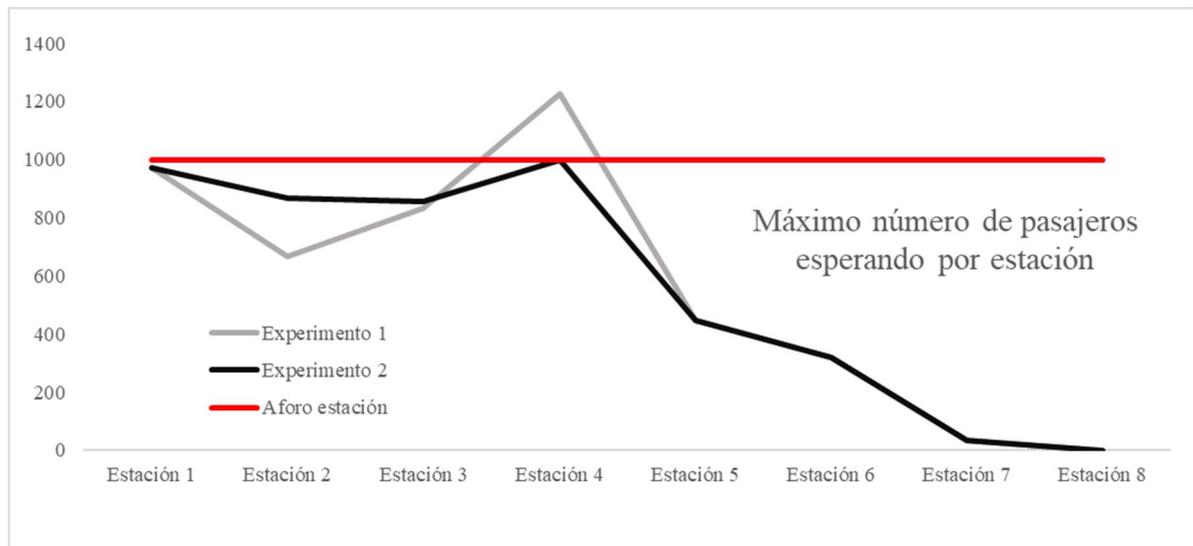


Gráfico 6: Máximo número de pasajeros por estación para factor de escala 0.6

En la gráfica se puede observar que en el experimento con reserva de vagones el umbral máximo no se llega a superar en ningún instante. Los vagones se han reservado de la siguiente manera:

VAGONES LIBRES	Estación 1	Estación 2	Estación 3	Estación 4	Estación 5	Estación 6	Estación 7	Estación 8
Tren 1	2	6	6	6	6	6	6	6
Tren 2	6	6	6	6	6	6	6	6
Tren 3	2	4	6	6	6	6	6	6
Tren 4	5	5	6	6	6	6	6	6
Tren 5	5	6	6	6	6	6	6	6
Tren 6	3	3	5	6	6	6	6	6
Tren 7	4	5	5	6	6	6	6	6
Tren 8	6	6	6	6	6	6	6	6

Tabla 13: Vagones disponibles en 2ª experimentación

En la situación en la que la reserva de vagones estaba deshabilitada el umbral máximo se superaba en 2 ocasiones: Cuando el tren 5 llegaba a la estación 4 y cuando el último tren 8 llegaba a la 4ª estación.

Para ambos casos, la reserva de vagones se antoja clave para que este límite no se sobrepase. La reserva de vagones en los servicios 3, 4 y 5 permiten que en la estación 4 el número de pasajeros esperando al quinto tren se reduzca de **1014** a **997**, suficiente para que el aforo no se supere. Esta demanda debe por supuesto repartirse y ser satisfecha por otros servicios. Por ejemplo, en la estación 3 el número de pasajeros esperando al 5º tren aumenta en **190** pasajeros.

EXPERIMENTO 1	Estación 1	Estación 2	Estación 3	Estación 4	EXPERIMENTO 2	Estación 1	Estación 2	Estación 3	Estación 4
Tren 1	241	295	328	296	Tren 1	241	295	328	296
Tren 2	193	161	176	173	Tren 2	193	161	176	173
Tren 3	345	482	549	653	Tren 3	345	482	549	653
Tren 4	973	614	670	964	Tren 4	973	614	670	964
Tren 5	520	668	831	1014	Tren 5	520	868	641	997

Tabla 14: Comparación de flujo de pasajeros con factor de escala 0.6

En el servicio 8, sin embargo, para no sobrepasar el aforo los vagones en los trenes 7 y 8 no se desbloquean por completo hasta llegar a la estación 4. En esta estación el número de pasajeros esperando a este tren si no se hiciera uso del modelo sería de **1.226** pasajeros.

La estación 4 soporta una alta demanda durante el total del experimento y además, los trenes suelen llegar a esta estación con pocos asientos o espacio libre:

Estación 4	Pasajeros esperando	Capacidad restante
Tren 1	296	552
Tren 2	173	839
Tren 3	653	275
Tren 4	964	250
Tren 5	1014	303
Tren 6	898	323
Tren 7	790	266
Tren 8	1226	259

Tabla 15: Flujo de pasajeros y capacidad restante para factor de escala 0.6

En segundo lugar, la demanda entre la llegada del tren 4 y el 8 sufre una subida que también aumenta considerablemente la aglomeración de pasajeros:



Gráfico 7: Demanda en estación 4 para factor de escala 0.6

10.1.3 Tercera experimentación

Con los 2 anteriores experimentos se ha podido comprobar el efecto de la reserva de vagones para una demanda alta o con un riesgo de aglomeración alto. A continuación, se tratará una demanda más baja que no implique bloqueos o situaciones de riesgo, pero cuyo flujo de pasajeros y tiempo de espera tengan un alto margen de mejora.

Factor de escala	0.5
Capacidad de vagón	200
Número de vagones	6
Capacidad de estación	1000
Umbral de seguridad de estación	600
Máximo número de vagones reservables	0
θ_1	0.5
θ_2	0.5
ε	100

Tabla 16: Datos iniciales para 3ª experimentación

Para el tercer y cuarto experimento se aplicarán 2 cambios fundamentales:

- El factor de escala ha sido disminuido hasta un valor de 0.5 evitando así una acumulación de pasajeros superior al umbral máximo de 1000.
- El umbral de seguridad de cada estación (U_k) se cambia de 400 a 600. Con este cambio se pretende que el modelo busque evitar llegar a valores entre 600 y 1000 pasajeros.

Sin la posibilidad de reservar vagones el resultado es el siguiente:

Valor Función Objetivo: $\theta_1 \cdot \sum_{k \in S} w_k + \theta_2 \cdot \sum_{k \in S} r_k = 14.971$

Como se puede apreciar en la siguiente tabla, el número de pasajeros esperando nunca es superior a la capacidad de la estación:

PASAJEROS ESPERANDO	Estación 1	Estación 2	Estación 3	Estación 4	Estación 5	Estación 6	Estación 7	Estación 8
Tren 1	221	263	236	281	255	224	33	0
Tren 2	158	132	167	103	128	168	14	0
Tren 3	324	364	403	442	438	247	19	0
Tren 4	746	478	631	437	449	205	10	0
Tren 5	371	339	745	469	214	126	3	0
Tren 6	329	298	279	385	129	89	12	0
Tren 7	483	222	217	159	6	58	2	0
Tren 8	359	427	367	557	376	268	25	0

Tabla 17:Flujo de pasajeros en 3ª experimentación

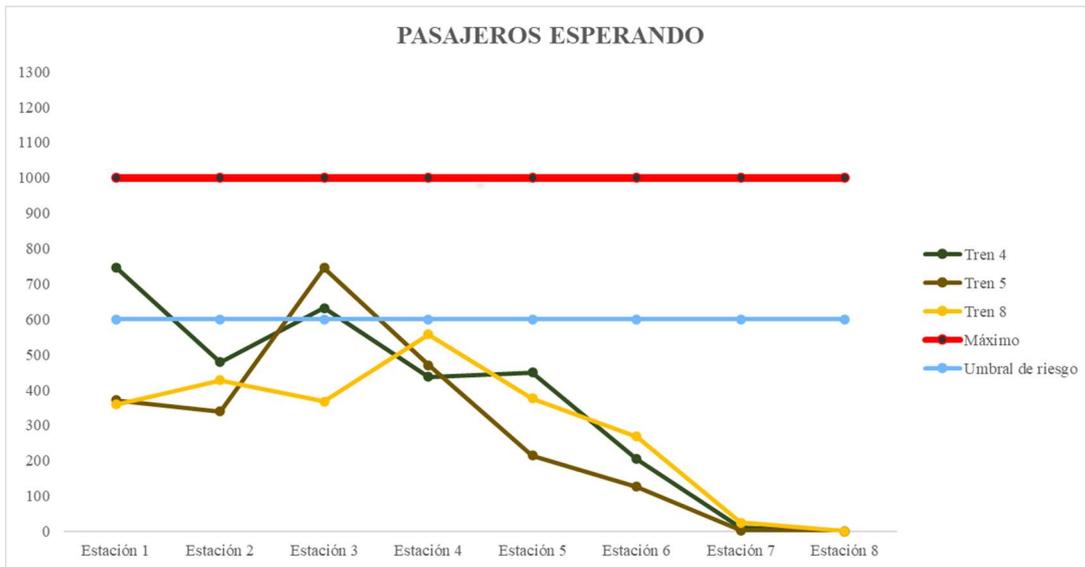


Gráfico 8:Flujo de pasajeros en 3ª experimentación

Sin embargo, superar la barrera de 600 pasajeros esperando entraña un riesgo en este modelo, y este umbral es superado hasta en 3 ocasiones cuando la reserva de vagones está inhabilitada.

RIESGO DE AGLOMERACIÓN	Estación 1	Estación 2	Estación 3	Estación 4	Estación 5	Estación 6	Estación 7	Estación 8
Tren 1	0	0	0	0	0	0	0	0
Tren 2	0	0	0	0	0	0	0	0
Tren 3	0	0	0	0	0	0	0	0
Tren 4	37	0	8	0	0	0	0	0
Tren 5	0	0	36	0	0	0	0	0
Tren 6	0	0	0	0	0	0	0	0
Tren 7	0	0	0	0	0	0	0	0
Tren 8	0	0	0	0	0	0	0	0

Tabla 18: Riesgo de aglomeración en 3ª experimentación

En concreto, al considerar el valor de ε igual a 100, los valores en la tabla superior reflejan la probabilidad de aglomeración para un servicio y estación concreta.

En el siguiente experimento se comprobará si la reserva de vagones es capaz de reordenar el flujo de pasajeros mitigando el riesgo de aglomeración.

10.1.4 Cuarta experimentación

Factor de escala	0.5
Capacidad de vagón	200
Número de vagones	6
Capacidad de estación	1000.00
Umbral de seguridad de estación	600.00
Máximo número de vagones reservables	6
θ_1	0.5
θ_2	0.5
ε	10000.00

Tabla 19: Datos iniciales para 4ª experimentación

Bajo las mismas condiciones que en el tercer experimento, se habilita ahora la reserva de vagones. Los resultados son los siguientes:

$$\text{Valor Función Objetivo: } \theta_1 \cdot \sum_{k \in S} w_k + \theta_2 \cdot \sum_{k \in S} r_k = 24.273$$

El valor de la función objetivo se ha reducido en un **15,7%**, gracias a la siguiente reserva de vagones:

VAGONES LIBRES	Estación 1	Estación 2	Estación 3	Estación 4	Estación 5	Estación 6	Estación 7	Estación 8
Tren 1	6	6	6	6	6	6	6	6
Tren 2	5	6	6	6	6	6	6	6
Tren 3	3	3	6	6	6	6	6	6
Tren 4	4	5	6	6	6	6	6	6
Tren 5	3	6	6	6	6	6	6	6
Tren 6	6	6	6	6	6	6	6	6
Tren 7	5	6	6	6	6	6	6	6
Tren 8	1	2	5	5	5	6	6	6

Tabla 20: Vagones disponibles en 4ª experimentación

Sin embargo, el umbral de riesgo sigue siendo superado hasta en 3 ocasiones.

PASAJEROS ESPERANDO	Estación 1	Estación 2	Estación 3	Estación 4	Estación 5	Estación 6	Estación 7	Estación 8
Tren 1	184	226	224	330	280	222	27	0
Tren 2	164	116	137	128	118	157	13	0
Tren 3	281	339	408	431	385	284	26	0
Tren 4	881	567	684	511	313	163	14	0
Tren 5	451	554	611	418	110	110	4	0
Tren 6	350	291	358	287	184	75	13	0
Tren 7	495	221	264	154	22	44	4	0
Tren 8	310	414	381	485	382	263	18	0

Tabla 21: Flujo de pasajeros en 4º experimentación

Disminuir el número de pasajeros esperando al tren número 5 a la llegada a la estación 3 en 182 pasajeros ha disminuido el riesgo de aglomeración considerablemente:

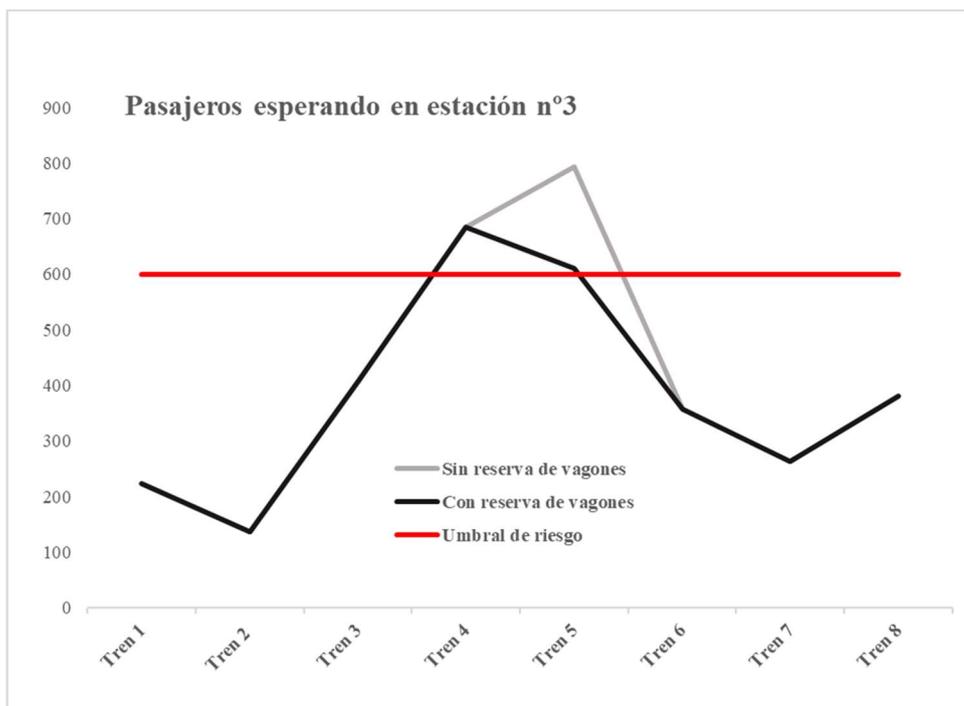


Tabla 22: Flujo de pasajeros en estación nª3 con factor de escala 0.5

En la gráfica superior se puede apreciar como la reserva de vagones también ayuda a reducir el tiempo que permanece la estación número 3 en situación de riesgo.

11 CONCLUSIONES

El mayor valor del modelo propuesto reside en la fácil implementación de las soluciones obtenidas. En modelos anteriormente mencionados como (Currie, 2010) la implementación de las soluciones al problema de congestión de las plataformas dependía de diversos factores externos que podían dificultarla. Entre otros, un cierto número de consumidores debían tomar la decisión de cambiar su horario para que el modelo obtuviera resultado. Otros modelos mencionados anteriormente (Niu & Zhou, 2013) se basan en un cambio del horario en función de la demanda, que da lugar a modelos más complejos y aplicables a instancias de pequeño tamaño.

Para la reserva de vagones la única premisa necesaria sería obtener datos sobre la demanda con cierta fiabilidad. Si los datos previos son fiables, el modelo que se propone resulta de desarrollo sencillo y sus soluciones pueden ser puestas en práctica de forma fácil en circunstancias reales. Factores externos como horarios o la predisposición del consumidor no influirían en las soluciones a implementar.

A partir de los experimentos realizados se puede comprobar la utilidad del modelo en diversas situaciones. Resulta muy efectivo si el objetivo es evitar sobrepasar valores de riesgo que pongan en peligro al pasajero. En esencia, al reservar vagones, el modelo reparte la demanda de pasajeros entre los distintos trenes que se detienen en las estaciones. Si el flujo de pasajeros en una estación toma valores muy elevados o de riesgo, la reserva de vagones garantizará que los trenes entrantes tengan la capacidad suficiente para aliviar la demanda.

Un ejemplo de este caso serían los Experimentos 1 y 2. Gracias a que los trenes mantienen varios de sus vagones cerrados el número de pasajeros esperando nunca llega a superar la capacidad de las estaciones. Sin esta reserva, por el contrario, alguna de las estaciones habrían superado el aforo máximo en distintos momentos.

Además, esta misma casuística puede ser usada para disminuir la acumulación de pasajeros aun cuando no suponen un riesgo. Es decir, el modelo es capaz de repartir la demanda de forma que ninguna estación sufra una acumulación de pasajeros, y por ende unos tiempos de espera mucho mayores que las demás.

Aun así, el modelo sufre también ciertas limitaciones. Existen diversos modelos (Xu, et al., 2016) que contemplan la posibilidad de una demanda incierta, es decir, de una situación inusual que dispara la demanda a valores atípicos. La reserva de vagones tal como se ha planteado no contempla más que el caso determinista y sería necesario modificar el modelo propuesto para afrontar una situación de demanda aleatoria.

Otro aspecto a tener en cuenta son las estaciones con un flujo de pasajeros bajo, dado que el modelo propuesto se basa en equilibrar la demanda a lo largo de la línea para evitar aglomeraciones, en el caso de que la demanda sea muy baja, los pasajeros en esas estaciones podrían verse perjudicados en lo que respecta sus tiempos de espera. Volviendo al ejemplo de la red de transporte subterráneo en Londres, existen estaciones cuyo flujo diario está entorno a los 1000 pasajeros. Si el modelo considera que en estas estaciones no existe riesgo de aglomeración podría no habilitar vagones, y por tanto un aumento en los tiempos de espera de los pasajeros esperando en la estación.

En la mayoría de grandes ciudades, las estaciones con un flujo menor de pasajeros se suelen encontrar en el extrarradio o periferia de la ciudad. Impedir a un pasajero del extrarradio que suba a bordo de un tren para evitar aglomeraciones en zonas más concurridas en la ciudad puede ocasionar problemas de injusticia o exclusión social.

Para concluir, la reserva de vagones puede ser un elemento muy interesante como solución a la aglomeración en estaciones de ferrocarril de grandes ciudades. Su fácil implementación y adaptabilidad hacen que sea una opción a tener en cuenta. Como se ha comentado, la estrategia de reserva de vagones presenta ciertas limitaciones que dificultan el control de los tiempos de espera o la respuesta ante situaciones remotas. A pesar de ello, se ha mostrado como una solución válida en ciertas ocasiones. Otro aspecto interesante es la posible combinación del mecanismo de reserva con otras estrategias utilizadas para mejorar la experiencia de los usuarios, tales como Stop-Skipping o Short-Turning.

12 BIBLIOGRAFÍA

ABC, 2018. *El éxito de Inditex: al mal tiempo, un buen algoritmo*, s.l.: ABC.

Barrena, E., Canca , D., Coelho, L. & Laporte, G., 2014. Single-line rail rapid transit timetabling under dynamic passenger demand. *Transportation Research Part B: Methodological*, Volumen 70, pp. 134-150

Cadarso, L., Marín, Á. & Maroti, G., 2013. Recovery of disruptions in rapid transit networks. *Transportation Research Part E Logistics and Transportation Review* , 53(1), p. 15–33.

Canca, D., Barrena, E. & Algaba , E., 2014. Design and analysis of demand-adapted railway timetables. *Journal of Advanced Transportation*, 48(2), pp. 119-137.

Chapman, L., 2007. Transport and climate change: a review. *Journal of Transport Geography*, 15(5), pp. Pages 354-367.

Currie, G., 2010. Quick and Effective Solution to Rail Overcrowding: Free Early Bird Ticket Experience in Melbourne, Australia. *Transportation Research Record*, Volumen 2146, pp. 35-42.

El País, 2021. Transición Ecológica lanza una guía para que los ayuntamientos creen sus zonas de bajas emisiones. *EL PAÍS*, 19 Noviembre, p. 2.

freejournal.org, 2020. [En línea].

Jiang, Z. y otros, 2018. Reinforcement learning approach for coordinated passenger inflow control of urban rail transit in peak hours. *Transportation Research Part C: Emerging Technologies*, Volumen 88, pp. 1-16.

Montañés, É., 2018. La población crece solo en las grandes ciudades. *ABC*.

Niu, H. & Zhou, X., 2013. Optimizing urban rail timetable under time-dependent demand and oversaturated. *Transportation Research Part C: Emerging Technologies*, Volumen 36, pp. 212-230.

Roser, M., 2013. Future Population Growth. *Our World In Data*.

Shi, J., 2020. Dynamic train capacity allocation for an overcrowded metro line: An effective passenger flow control approach. *the National Natural Science Foundation of China*, p. 35.

Shi, Z., Zian, Z., Ran, J. & Jiaer, L., 2020. The association between domestic train transportation and novel coronavirus (2019-nCoV) outbreak in China from 2019 to 2020: A data-driven correlational report. *Travel Medicine and Infectious Disease: Elsevier BV*, Volumen 33.

Telegraph UK, 2017. *telegraph.co.uk*. [En línea]
Available at: <https://www.telegraph.co.uk/travel/destinations/europe/united-kingdom/england/london/articles/London-Underground-150-fascinating-Tube-facts/>
[Último acceso: 21 Octubre 2021].

Transport For London, 2017. *London Underground Performance Reports*, Londres: s.n.

Xu, X.-j., Liu, J., Lin, H.-y. & Yiang, M., 2016. Capacity-oriented passenger flow control under uncertain demand: Algorithm development and real-world case study. *Transportation Research Part E: Logistics and Transportation Review*, Volumen 87, pp. 130-148.

Yang , H. & Tang, Y., 2018. Managing rail transit peak-hour congestion with a fare-reward scheme. *Transportation Research Part B: Methodological*, 110(C), pp. 122-136.