



ESCUELA POLITÉCNICA SUPERIOR
UNIVERSIDAD DE SEVILLA



TRABAJO DE FIN DE GRADO

PROGRAMACIÓN MATEMÁTICA PARA EL DISEÑO DE LÍNEAS DE TRANSPORTE

Grado en Ingeniería de Diseño Industrial y Desarrollo del Producto

Curso: 2023/2024

Autor: Marisa Paz Fernández

Tutor: Federico Perea Rojas-Marcos

ÍNDICE

1. Introducción.....	4
2. Revisión bibliográfica.....	15
2.1. Problemas en la planificación del diseño de una red de transporte de tránsito rápido.	17
2.2. Robustez de una red de tránsito rápido.	34
2.3. Formas de evaluar una red de transporte de tránsito rápido..	36
3. Modelos matemáticos para el diseño de una red de tránsito rápido.....	39
3.1. Modelo matemático 1.....	40
3.1.1. Datos y notación.	41
3.1.2. Variables.	43
3.1.3. Función objetivo y restricciones.	43
3.2. Modelo matemático 2.	48
3.2.1. Datos y notación.....	49
3.2.2. Variables.....	50
3.2.3. Función objetivo y restricciones.....	50
4. Experimentos. Resultados y análisis.....	53
4.1. Datos introducidos.	54
4.2. Resultados.	59

5. Conclusiones	69
6. Bibliografía	71
7. Anexos.....	78
7.1. Anexo 1: Modelo 1 de Programación Matemática para el Diseño de Líneas de Tránsito rápido.....	78
7.2. Anexo 2: Modelo 2 de Programación Matemática para el Diseño de Líneas de Tránsito Rápido.	86
7.3. Anexo 3: Bases de datos.....	91

1. INTRODUCCIÓN.

El diseño de una buena red de transporte es de suma importancia en las ciudades, dado que es uno de los factores clave para garantizar una movilidad eficiente y sostenible. Su buen diseño es esencial para promover una ciudad más sostenible, dado que su uso hace que disminuya la contaminación ambiental producida por las emisiones de gases de efecto invernadero, ruido, y otros contaminantes, que se producen por el uso masivo del transporte de vehículos individuales. Al proporcionar una infraestructura de transporte eficaz se reduce el número de vehículos en las calles, el tiempo de viaje, y se evita la congestión del tráfico, ayudando de esta forma a reducir la contaminación del aire y mejorar la salud pública. Por otro lado, la red debe estar diseñada teniendo en cuenta la eficiencia en el uso del espacio y los recursos. Esto significa que los planes de transporte deben tener en cuenta la densidad poblacional, la distribución de los lugares de trabajo y los espacios de ocio de las ciudades más transitadas,

además de la cantidad de espacio disponible para la infraestructura del transporte. De esta forma, brindará a los habitantes de la ciudad un acceso fácil y eficiente a lugares de trabajo, lugares de entretenimiento y servicios públicos, mejorando la calidad de vida de las personas y la equidad social, en la medida en que se fomente el acceso a oportunidades para todos los miembros de la comunidad.

La movilidad urbana sostenible es uno de los principales retos que afrontan las ciudades de la Unión Europea y un motivo de preocupación para muchos de sus ciudadanos. Conforme nuestra sociedad se hace más móvil, la Unión Europea colabora con las ciudades y regiones, para elaborar una estrategia de movilidad urbana sostenible que engloba la creación de sistemas de transporte público eficientes y una conectividad sólida. Según el Informe especial: Movilidad urbana, 2023 [18] se considera que el problema más grave que afecta a la movilidad urbana es la congestión de tráfico, el cual se trata de uno de los desafíos más difíciles de afrontar. Diversas investigaciones económicas señalan que los efectos negativos de la congestión para la sociedad son de alto coste. Además, se observa que en aquellas zonas urbanas donde el tráfico fluye con mayor facilidad, existe una mayor probabilidad de impulsar el crecimiento económico. Actualmente, la principal fuente de financiación de la movilidad urbana procede de varios fondos públicos de inversión, destacando: Fondos Estructurales y de Inversión Europeos (Fondos EIE), el Fondo Europeo de Desarrollo Regional (FEDER) y el Fondo de Cohesión (FC), los cuales van aumentando en cada periodo de revisión el importe disponible, para aumentar la

eficacia de los proyectos de transporte. Por otro lado, los fondos de la UE para expandir sus redes de transporte público en líneas de metro y tranvía han tenido en cuenta otros factores como la facilidad de acceso, frecuencia, rapidez y un grado elevado de conectividad de la red.

Concretamente el Ministerio de Transportes, Movilidad y Agenda Urbana (MITMA), que ha desarrollado la Estrategia de Movilidad Segura, Sostenible y Conectada 2030 [10], pretende gestionar los retos y oportunidades de la transformación revolucionaria de la movilidad, impulsada por tres motivos, siendo el primero la urgencia en avanzar en la descarbonización del transporte y luchar contra la emergencia climática, el segundo es la necesidad de incorporación masiva de las nuevas tecnologías, y el tercer motivo se relaciona con la concentración demográfica en zonas urbanas y zonas periurbanas, generando problemas como congestión y pérdida de población en áreas rurales y medianas, lo que dificulta atender las necesidades de movilidad. Este último, conlleva desafíos importantes de gestión tanto en áreas urbanas de gran tamaño (distribución del espacio público, efectos en la salud debido a la calidad del aire y ruido, congestión, entre otros), como cuestiones de las zonas rurales y ciudades de tamaño medio, que lidian con la despoblación y necesitan soluciones de movilidad adaptadas, mientras también se enfrentan a retos demográficos como el envejecimiento de la población.

En la era actual, el panorama de la movilidad ha experimentado un profundo cambio de paradigma. Las transformaciones en la sociedad y en la

tecnología han llevado a una ampliación del concepto de movilidad, extendiéndose más allá de su interpretación tradicional. Ya no se trata simplemente de mover personas o mercancías entre puntos, sino de abordar una red de complejidades interconectadas. Esta extensión de concepto abarca aquellos condicionantes, necesidades, motivaciones y percepciones de los individuos, que sitúan al ciudadano y al usuario del transporte en el centro del sistema (ver Ilustración 1).

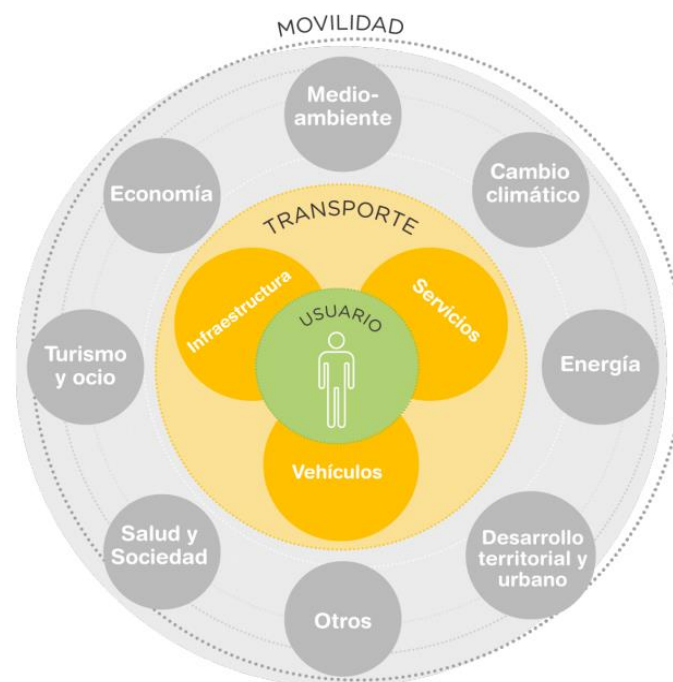


Ilustración 1: Las relaciones multidisciplinares de la movilidad. Foro: MITMA

Esta nueva visión de movilidad implica, además de una mejora en la calidad de los servicios de transportes, una optimización de la planificación urbana y un enfoque hacia la sostenibilidad, lo que tendría un impacto sumamente positivo en las ciudades, aumentando el uso de los servicios de transporte públicos y abordando uno de los problemas más significativos actualmente: la emisión de gases de efecto invernadero provenientes del uso de vehículos

individuales en las ciudades, lo cual a su vez, supondría una mejora de calidad del aire y salud pública. Es por estas razones que el problema de diseño de redes de transporte público presenta un proceso de planificación compleja. Una de las fases en las que se puede dividir la planificación de una red de transporte, una vez su infraestructura ha sido construida, puede ser la siguiente.

1. Planificación global de la red de líneas.
2. Establecimiento de horarios
3. Programación de vehículos.
4. Programación de tripulaciones.

Entre los transportes públicos, los sistemas de metro son los más eficientes, ya que consumen menos energía y pueden transportar a más pasajeros por unidad de superficie que cualquier otra forma de transporte. Además, ayudan a reducir el tráfico de automóviles privados, disminuyendo así la contaminación. Otra característica importante es su influencia estructuradora en las ciudades, ya que establecen la base para el desarrollo de áreas residenciales, comerciales y empresariales. A su vez, es importante destacar la perspectiva de los usuarios; el sistema debe satisfacer la demanda ofreciendo un servicio económico y directo a los pasajeros, además de aportar comodidad, regularidad, cobertura y un aceptable nivel de frecuencia. En contraste, la perspectiva del operador es crear un sistema que genere la mayor ganancia posible, minimizando costes. Esto conlleva realizar compensaciones entre

objetivos, y es en este punto donde entran en juego diversas técnicas de optimización.

La planificación general del transporte puede ser descrita y analizada mediante modelos matemáticos, siendo precisamente éste el enfoque de este proyecto. Este estudio se centra en los Sistemas de Metro o Sistemas de Transporte de Tránsito Rápido. Un transporte de tránsito rápido tiene un significado amplio. Según la literatura, se considera transporte rápido a los trenes de mercancías, tranvías, metros y autobuses. A pesar de esto, a veces, el término metro se usa indistintamente con “transporte de tránsito rápido”, según Mesa, J (2015) [31]. El objetivo principal de este documento es utilizar la Programación Lineal para el diseño de redes de tránsito rápido en ciudades, mediante aplicación de técnicas de optimización matemática, lo cual puede jugar un papel importante al abordar el diseño y la gestión de sistemas de transporte público, considerando aspectos como la eficiencia de rutas, la reducción de tiempos de viaje y la distribución equitativa de los servicios en toda la ciudad.

No obstante, resolver los problemas resultantes de la integración del diseño de una red y la planificación de líneas es una tarea compleja dado que incluso sus subproblemas considerados por separado ya son NP-duros, desde el punto de vista de la complejidad computacional. Un problema NP-duro (no polinómico duro) es un problema para el cual no se ha demostrado que exista un algoritmo de tiempo polinómico que lo resuelva de manera eficiente. Estos problemas son una categoría especial de problemas de optimización que son al

menos tan difíciles de resolver como los problemas NP (no existe un algoritmo determinístico polinomial que lo resuelva). Por otro lado, el proceso de diseñar una red de tránsito rápido se complica en una ciudad por factores como el presupuesto disponible, posibles retrasos, el encarecimiento de obras, la necesidad de infraestructuras etc. Esto sugiere que hay algunas variables que están más allá de la formulación o el control. No obstante, para un diseño realista y óptimo es necesario llevar a cabo un plan con una serie de fases integradas que conducirá a mejores soluciones. Diversos autores, como Ceder y Wilson (1986) [7], Schöbel (2011) [35], V. Guihaire y Hao (2008) [16] y Marín et al. (2009) [30], sostienen que con la integración de múltiples etapas de planificación se podrían obtener soluciones más efectivas, siempre que sean factibles.

Tradicionalmente, el proceso de planificación de transporte rápido se ha descompuesto en una sucesión de etapas:

1. El diseño de la red
2. El diseño de la línea
3. La programación de horarios
4. El material rodante.
5. La planificación del personal.

Según Marín et al. (2009) [30] se pueden identificar dos objetivos principales en conflicto que se persiguen a la hora de planificar un sistema de líneas: optimizar el servicio de los pasajeros y minimizar los costes del sistema

de trenes interurbanos. La mejora del servicio de pasajeros puede definirse desde diferentes puntos de vista; minimizar las transferencias, minimizar el tiempo total del viaje o maximizar la comodidad. También se podría maximizar el número de conexiones directas, pero esto suele dar como resultado líneas largas, las cuales normalmente producen retrasos con más facilidad y gasto de material rodante. Sin embargo, las líneas cortas pueden presentar el problema de obligar a los pasajeros a realizar transferencias con demasiada frecuencia.

Los distintos modelos creados, y estudios de los problemas de planificación de líneas ferroviarias se pueden clasificar en dos grupos: desde el punto de vista del operador y desde el punto de vista del usuario. Según el enfoque del primer grupo, el modelo diseñado de optimización debería minimizar los costes, Claessens et al. (1998) [9] utiliza un modelo de programación matemática no-lineal entera mixta con variables de decisión binarias, y variables para las frecuencias o longitudes de trenes. Desde la perspectiva del usuario, el modelo debe maximizar la cobertura, minimizar tiempos de viaje o minimizar el número de transferencias. En el artículo Scholl 2005 [36] explica qué implica el uso de rutas de pasajeros en el modelo (concepto definido como “Switch-and-Ride”), entre otros que se explicarán más detalladamente en el siguiente capítulo.

Concretamente, este documento presentará dos propuestas para integrar en las fases de diseño de red y planificación de líneas, abordando dos problemas de Programación Lineal Entera para el diseño y construcción de una línea de

tránsito rápido, con el fin de comparar su eficacia y aportar la mejor solución que se adapte a la realidad. Por otro lado, el objetivo de este proyecto es implementar técnicas de programación matemática para optimizar el proceso de planificación de transporte mencionado en párrafos anteriores. Además, ambos modelos tendrán en común el enfoque de optimizar la experiencia del usuario.

Los modelos que se exponen en el capítulo 3, están basados en el descrito en el artículo de García-Archilla, B., Lozano, A., Mesa, J. A., & Perea, F. 2011 [14], en el que se propone un algoritmo de Búsqueda Aleatoria Adaptativa Greedy GRASP, que proviene de las siglas de “Greedy Randomized Adaptive Search Procedure”. GRASP es una metaheurística introducida por Feo y Resende en 1995 [38] desarrollada con el objetivo inicial de resolver problema de cobertura de conjuntos, es generalmente robusta, y se basa en un procedimiento de búsqueda aleatorio y adaptativo que, aplicándolo a problemas de optimización, garantiza una buena solución, aunque no necesariamente la óptima. Además, es un procedimiento iterativo donde cada paso consiste en una fase de construcción, de mejora, y otra de actualización. No obstante, el artículo mencionado presenta un modelo de Programación Lineal Entera para el problema de diseño de una red ferroviaria interurbana, estaciones y enlaces que las conectan, que simplifica modelos demasiados complejos como por ejemplo en Laporte et al. (2010,2011) [23],[21].

En el presente trabajo se tomará como referencia el Trabajo de Fin de Máster de Manzanares Abásolo, M. A. (2022) [27], que trata el diseño de una

línea de tránsito rápido enfocado para que pueda servir como herramienta en la planificación y creación de una ruta específica que conecta más de dos puntos clave. Por lo general, implica determinar el recorrido exacto de la línea, las estaciones a lo largo de la ruta y las interconexiones de éstas, siendo el objetivo principal proporcionar una conexión rápida y eficiente entre puntos específicos de una ciudad o región. Por otro lado, el diseño de una red tiene un enfoque más amplio que abarca la planificación y creación de líneas interconectadas, implica la identificación y diseño de diferentes rutas y líneas que cubren una extensa área geográfica, conectando múltiples destinos y estaciones. El objetivo es crear una red integral y coherente de múltiples rutas.

El resto de esta memoria está organizada de la siguiente manera. En el Capítulo 2 se presenta una revisión de los problemas de diseño de redes de tránsito rápido y analizamos los diferentes modelos encontrados en la literatura de gestión y planificación de redes o líneas de transporte. En el Capítulo 3 se detallan los dos modelos matemáticos estudiados para la planificación de una línea de transporte de tránsito rápido en una ciudad. Se propone un modelo ya utilizado por varios autores, que maximiza el número de viajes de pasajeros entre estaciones, utilizando como datos de entrada una matriz Origen-Destino (OD) y teniendo en cuenta otro medio de transporte ya existente en la ciudad. Este modelo como veremos posteriormente puede complicarse computacionalmente, es por ello que se presenta un segundo modelo en el que se intenta simplificar, y se intenta que el transporte llegue al máximo número de habitantes por zonas. En el Capítulo 4, se obtienen los resultados de estos modelos mediante

simulación y se comparan las soluciones obtenidas, así como su eficacia.

Finalmente, se la memoria termina con las conclusiones.

2. REVISIÓN BIBLIOGRÁFICA.

El diseño de un modelo matemático exacto para la planificación de redes de tránsito rápido en una ciudad es posible, pero no es una tarea sencilla debido a la complejidad y la cantidad de variables involucradas. Como se ha introducido anteriormente, requiere una planificación cuidadosa y una consideración de varios factores, como las necesidades de transporte de la ciudad, el crecimiento futuro, las características topográficas y la demanda de transporte. El uso de modelos matemáticos en el diseño de una red de tránsito rápido en una ciudad es cada vez más común. Los modelos matemáticos pueden ayudar a los planificadores de transporte a analizar diferentes opciones de rutas, evaluar la demanda de pasajeros y determinar la capacidad necesaria para el sistema. Por otro lado, los modelos matemáticos también pueden ayudar a optimizar la ubicación de las estaciones, la frecuencia de los trenes y otros aspectos

importantes del diseño de la red de tránsito rápido. Además, estos modelos pueden ser útiles para simular diferentes escenarios y evaluar el impacto de posibles cambios en la red en términos de tiempos de viaje, capacidad, costes, y otros aspectos.

No obstante, los modelos matemáticos avanzados que se han desarrollado para la planificación de redes de transporte son de alto coste, y requieren una cantidad significativa de recursos y tiempo para desarrollarse y mantenerse actualizados. Adicionalmente, la utilización de modelos matemáticos en el diseño de una red de tránsito rápido depende en gran medida de la disponibilidad de datos precisos y actualizados, así como de los recursos financieros y técnicos necesarios para llevar a cabo la planificación y construcción del sistema. Esto hace, que se dificulte su aplicación matemática, dado que es importante tener en cuenta las limitaciones y posibilidades de los solvers que resuelven los modelos matemáticos en función de las condiciones específicas de cada proyecto. Es por esta razón, que a lo largo de este capítulo se exponen algunos de los modelos ya creados y puestos a prueba que se han realizado a lo largo del tiempo. Empezaremos resumiendo varias revisiones que resuelven los problemas más comunes que se presentan en el diseño de redes, por lo tanto, se explicará a continuación qué recogen algunos autores en sus artículos publicados sobre la evaluación de la efectividad de las redes diseñadas.

Los artículos de investigación descritos a continuación han sido seleccionados debido a su relevancia en relación con las contribuciones de la

investigación que se expone en este documento. Además, se debe aclarar que a pesar de que se exponen artículos en orden cronológico, se omiten por cuestiones de espacio muchas investigaciones y otros artículos que han sido fundamentos de los presentes en este trabajo.

2.1. Problemas en la planificación del diseño de una red de transporte de tránsito rápido.

Hasta donde sabemos, Dufourd et al. (1996) [12] proporcionaron el primer intento real de resolver el problema de ubicar una línea de tránsito teniendo en cuenta el espaciado máximo y mínimo entre estaciones y el número de estaciones permitidas para ubicar. En este artículo, el objetivo es maximizar la población cubierta por las estaciones. Esto se calcula utilizando varios niveles de captación. Los autores diseñaron un procedimiento de construcción tipo greedy para generar una solución inicial que luego se mejoró mediante búsqueda tabú. Un procedimiento de construcción tipo greedy implica tomar decisiones para construir la solución paso a paso, eligiendo en cada etapa la opción que parezca más beneficiosa en ese momento según algún criterio. En este caso, podría tratar de elegir las ubicaciones de las estaciones o determinar la ruta entre estaciones en función de ciertas consideraciones, como la demanda o la distancia. Tras este procedimiento, se realiza la búsqueda tabú, siendo una técnica de optimización que busca mejorar la solución inicial al explorar nuevas áreas del espacio de soluciones.

El estudio que se presenta en Laporte et al. (2005) [25] describe un modelo matemático y varias heurísticas para la construcción de una alineación de tránsito rápido, siendo su objetivo la maximización de la demanda total origen-destino cubierta por la línea de transporte. Además, identifica que la parte crucial en el proceso de planificación de un sistema de transporte de tránsito rápido consiste en dos problemas entrelazados; la determinación de rutas y la ubicación de estaciones. Utiliza dos enfoques para este problema, el primero localiza inicialmente las estaciones principales y luego se determina implícitamente la alineación ubicando las estaciones restantes entre aquellas que se fijaron en el primer paso (enfoque utilizado para el diseño del metro de Sevilla). En ambos enfoques, el nivel de tráfico pronosticado se puede determinar mediante el modelo clásico de análisis de cuatro etapas:

1. Análisis de Generación de Viajes: cálculo del número de viajes que comienzan en cada zona para cada propósito de viaje en particular.
2. Análisis de Distribución de Viajes: producción de una tabla que contiene el número de viajes que comienzan en cada zona y terminan en cada otra zona.
3. Análisis de Elección Modal: asignación de viajes entre los sistemas de transporte actualmente disponibles (autobús, tren, peatón y vehículos privados).
4. Análisis de Asignación de Viajes: asignación de flujos de viajes para las rutas específicas en cada sistema de transporte que serán seleccionadas por los usuarios.

Tras este proceso de cuatro etapas se podrá identificar una lista de posibles corredores o trayectos de transporte rápido, que se evaluarán en función de varios factores como la demanda de pasajeros esperada. Luego, se ordenan los corredores y los que se seleccionan se combinan en varias redes que dan lugar a diferentes escenarios. Este enfoque de planificación no necesita aplicar métodos de optimización, ya que la lista posible de corredores que se obtiene al principio es corta y manejable. Sin embargo, es probable que se eliminen buenas alternativas en etapas tempranas. Más adelante en este capítulo veremos cómo se puede optimizar este proceso.

Además, detalla que el objetivo principal al diseñar un sistema de transporte rápido es proporcionar los tiempos de viaje más cortos para mejorar la movilidad. Se asume que el número esperado de viajes está relacionado con la población cubierta por el sistema (personas que viven o trabajan cerca de las líneas) y se suele abordar con la maximización de la cobertura que puede proporcionar la línea. El estudio propone tres enfoques para medir la cobertura de la línea.

1. El primero es **trazar corredores** alrededor de la ruta y asignar factores de cobertura a cada corredor, enfoque que fue utilizado anteriormente por Chapleau et al. (1986) [8]. La cobertura disminuye gradualmente a medida que la distancia a pie a la alineación disminuye y finalmente llega a cero (ver Ilustración 2). No obstante, este enfoque es bastante insatisfactorio, ya que ignora el hecho de que alguien que vive cerca

de una línea de metro, pero lejos de una estación, no se sentirá atraído por la línea. Investigaciones posteriores, presentarán una metodología que combina alineaciones de varios corredores para el diseño de redes de metro, utilizando un programa lineal entero bi-criterio.

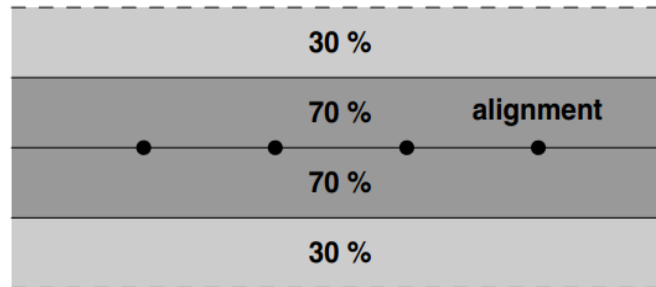


Ilustración 2: Alineación con dos corredores integrados alrededor. Los valores muestran el porcentaje de la población que vive en un corredor dado que probablemente usará el sistema de transporte. Fuente Laporte 2005.

2. Una segunda forma más precisa de medir la cobertura es mediante la construcción de **áreas de captación** alrededor de cada estación (ver Ilustración 3). Al igual que en el método anterior, construyen formas geométricas concéntricas alrededor de cada estación, y se estima el número de usuarios potenciales recolectando datos de población de los distritos censales que se superponen a estas formas. Sin embargo, este enfoque no considera la demanda de viajes entre estaciones, lo cual puede limitar su precisión. Por lo tanto, es poco probable que una persona que viaja a lo largo del eje Norte-Sur se sienta atraída por una estación ubicada en una alineación Este-Oeste (suponiendo que esta alineación no forma parte de una red interconectada más grande).

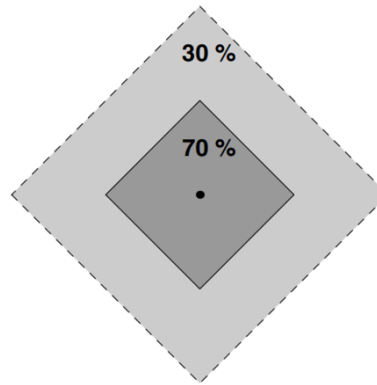


Ilustración 3: estación con zonas de captación integradas a su alrededor. Fuente Laporte 2005.

3. El tercer enfoque se basa en la **demanda real entre pares de origen/destino** (O/D). Los datos requeridos para este modelo (información O/D, distribución de población en cada distrito censal y forma geométrica de los distritos censales) son más voluminosos, pero generalmente están disponibles en agencias gubernamentales; además, estos datos pueden visualizarse mediante un Sistema de Información Geográfica (SIG) e integrarse con un software de optimización. Aunque este tercer enfoque requiere un esfuerzo mayor en la recopilación de datos y el cálculo, se basa en supuestos más realistas.

El artículo presenta una metodología de ubicación de rutas basada en información Origen-Destino, lo que incluye los tres primeros pasos de la metodología clásica de planificación de tránsito de cuatro etapas. Además, describe un modelo para la estimación de la demanda real entre dos estaciones,

se proponen heurísticas simples para el diseño de una línea que maximice la cobertura de viajes, y se resuelve el problema en presencia de un modo de transporte alternativo. Los autores utilizaron una variable continua que representa la distribución de la demanda del par asignada a cada modo, de acuerdo con una función Logit que depende de la diferencia entre los tiempos de viaje (o costes) de ambos modos. Y, se llega finalmente a la conclusión que la tercera forma de contemplar la cobertura de una línea es computacionalmente factible, a pesar de su dificultad.

Como se ha visto anteriormente, con el análisis clásico, se pueden llegar a eliminar buenas alternativas en etapas tempranas y la cantidad de información manejable puede llegar a ser insuficiente para obtener una solución óptima. En el artículo de Laporte et al. (2007) [20] se propone un proceso de planificación de sistemas de tránsito rápido de tres etapas para poder optimizar el proceso de análisis clásico que se resume en cuatro etapas para seleccionar una red de líneas de un sistema de transporte masivo, y ayudar a obtener una predicción de la demanda. Por otro lado, con el fin de poder considerar un mayor número de candidatos, intenta fomentar la introducción de métodos de optimización en el proceso de diseño, y se centra en la determinación de una sola ruta y la ubicación de las estaciones. Para lograrlo, se identifican tres etapas.

1. Selección de nodos claves: lugares claves donde se generan los viajes, ya sean como origen o destino, que son los principales puntos de atracción: estaciones de ferrocarril o autobuses y aeropuertos,

hospitales, campus universitarios, grandes tiendas, centros comerciales y áreas densamente pobladas lejos del área central de la ciudad, etc.

2. Diseño de la red principal: conectar las estaciones clave (nodos seleccionados previamente) para formar una red central.
3. Ubicaciones de estaciones secundarias. Una vez que las líneas están ampliamente decididas se ubicaran nuevas estaciones secundarias a lo largo de los tramos determinados por pares de estaciones clave, con el fin de aumentar óptimamente la cobertura de la línea.

La Ilustración 4 muestra la división que se realiza en el estudio y se encuentra clasificadas por la gama de grises que se puede observar, siendo las zonas más oscuras las que están situadas lejos del centro de la ciudad, como también se ve reflejadas aquellas zonas que ofrecen un gran número de empleos

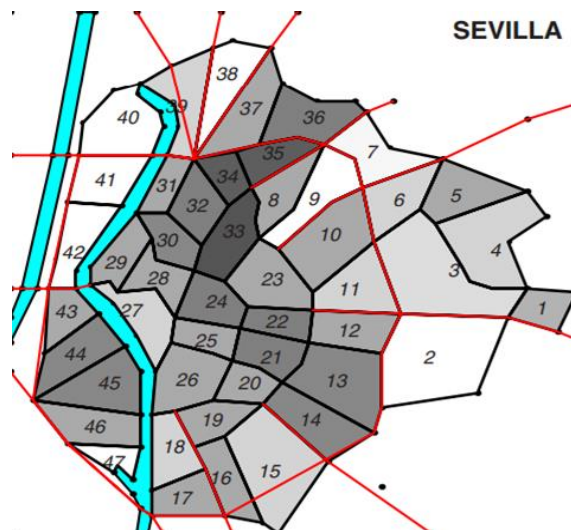


Ilustración 4: División en zonas de la ciudad de Sevilla en Laporte, 2007

o que contienen instalaciones importantes, que deben considerarse como zonas de atracción de viajes.

No obstante, en el artículo se aborda principalmente la segunda etapa, aunque se incluyen comentarios sobre la primera. Además, el modelo matemático presentado en el artículo utiliza la matriz Origen-Destino que representa el patrón de viajes. Se utilizó un modelo de programación lineal entera con el objetivo de maximizar la cobertura de viajes para resolver el problema de diseño de la red central en presencia de un modo de transporte alternativo. Importante destacar que, computacionalmente, el modelo no es capaz de manejar la cantidad de datos que supondría una situación real.

La tercera etapa la podemos ver reflejada en artículos como Bruno et al. (2002b) [4]. En este artículo se propone una estrategia de búsqueda más rápida y sólida que se puede utilizar para generar y optimizar una variedad de alineaciones potenciales. El método propuesto puede verse como un componente básico para el problema de diseño de redes multilíneas. En problemas de prueba, el método propuesto en este artículo produce soluciones óptimas o casi óptimas en unos pocos segundos, buscando maximizar la población total cubierta por la alineación. El método también se aplicó a datos de la ciudad de Milán y se obtuvo una buena alineación.

El artículo publicado por Mesa, J (2015) [31] resume que el proceso de planificación de un sistema de transporte rápido comienza analizando el área en

consideración y patrones de viaje, luego, se codifican los patrones en una matriz Origen-Destino, se identifican y combinan algunos corredores de tráfico amplios dando lugar a varios escenarios de red que pueden evaluarse desde diferentes puntos de vista, a menudo utilizando análisis de múltiples criterios. Esto permite considerar no sólo un único factor, sino varios factores importantes al mismo tiempo, siendo ejemplo de factores el tiempo de viaje, la reducción de impactos ambientales, la demanda de pasajeros, la integración de otros medios de transportes etc.

No obstante, estos documentos no tuvieron en cuenta las restricciones de capacidad, y las líneas resultantes no pudieron definirse completamente porque no se podía determinar la frecuencia. Marín, Á. M., & García-Ródenas, R. (2009) [28] van un paso más allá y consideran un modelo con una función de Logit para la cobertura de viajes y distribuir a los viajeros entre los modos de tránsito rápido y privado, manteniendo la linealidad del programa. Este enfoque permite capturar las preferencias de los individuos y predecir cómo tomarán decisiones de viaje, siendo el primer intento de tener en cuenta el comportamiento del usuario. Además, en su artículo explican que hasta ese momento en la literatura sólo se había estudiado la ubicación de la línea de tránsito del metro dentro de las zonas urbanas-suburbanas y el número y la ubicación de las estaciones. La principal crítica a estos modelos es que el comportamiento del usuario no se introduce explícitamente. Se supone implícitamente que una “buena” red de transporte promoverá su uso. Otros de los primeros intentos de tener en cuenta el comportamiento del usuario se da en Bruno (1998) [5], donde la elección del

modo se formula utilizando una asignación de modo todo-nada. Tras esto, Laporte et al. 2005 [23] introdujo la demanda Logit para modelar la elección de una sola línea de transporte. Por otro lado, expone que cuando el objetivo es implementar más de una línea de transporte se deben estudiar los siguientes dos problemas de diseño:

1. Problema de diseño de red física ferroviaria, donde el objetivo es determinar la infraestructura de la red, y su ubicación espacial.
2. Problema de diseño de la red de servicio, donde las líneas de tránsito y sus frecuencias se establecerán sobre la red física dada, y esto determinará dos aspectos claves para los usuarios: las posibilidades de conectividad y los tiempos de espera y traslado.

En el artículo se proponen diferentes modelos para resolver el problema de la red ferroviaria física, destacando dos aspectos relevantes: modelo de la demanda, donde se modelan las elecciones de ruta y modo de los usuarios y el modelo de oferta de tránsito, donde la red física ferroviaria está diseñada explícitamente, incluyendo aspectos del servicio y costes del transporte. El diseño de la red maximiza el transporte público y reduce la congestión privada.

A lo largo de los años, se han utilizado modelos de optimización y técnicas de Investigación Operativa para abordar algunos aspectos de la planificación de sistemas de tránsito masivo. Estos esfuerzos se han centrado principalmente en determinar una línea única y la ubicación de estaciones en función de ésta.

Principalmente, se han utilizado dos objetivos principales, en el caso de la ubicación de una sola línea: el primero es la cobertura de la población total, donde se busca maximizar el número de personas que se benefician directamente de la línea (a menudo definida como la población que vive dentro de un cierto umbral de distancia desde las estaciones), y el segundo objetivo es la cobertura de viaje, que busca maximizar la cantidad de viajes realizados en la red. Según el artículo de Mesa (2015) [31], el umbral de distancia desde las estaciones se ha establecido en 400 metros o cinco minutos andando en áreas densamente pobladas, pero puede aumentar a un kilómetro en regiones menos pobladas. Sin embargo, la cantidad de pasajeros no es solo una función de la distancia a la línea, sino también del diseño de la red, es decir, una estación ubicada en un área de alta densidad podría no ser eficiente debido a la dirección de la línea a la que pertenece. Es por esto por lo que el segundo objetivo proporciona una visión más realista, que puede ser evaluado a través de encuestas que evalúen el flujo de pasajeros a través de una matriz origen-destino, combinadas con modelos de equilibrio de tráfico.

Desde el primer modelo de programación matemática para el problema general de diseño de redes de tránsito rápido, que maximiza la cobertura de viaje esperada, se han desarrollado avances en diferentes funciones objetivo y algoritmos para la resolución del problema. Los modelos presentados en Laporte et al. (2010,2011) [23][21], son ejemplos de estudio de diseño de redes que intentan simplificar el problema, en los que se diseña un conjunto de líneas para optimizar una cierta función de utilidad y abordar la robustez de la red, no

obstante, son modelos demasiados complejos. Apenas podían manejar instancias de juguete compuestas por 10 estaciones posibles, y, además las líneas no estaban completamente definidas.

Si hablamos de expandir la capacidad de una red de transporte de tránsito rápido, Marín y Jaramillo (2008) [29], explica que la expansión de la capacidad es un problema que debe estudiarse dado que la construcción de la infraestructura es costosa y produce perturbaciones significativas en la vida de la ciudad, por lo tanto, se realiza en etapas a lo largo de varios años. Durante este período de planificación a largo plazo, los datos del problema suelen cambiar mucho, inflación, demanda de viajes, etc, y, recalca que estos cambios no se pueden ignorar en un modelo real. La formulación del modelo se basa en considerar las decisiones durante ciertos periodos de tiempo, se asume que los nodos y aristas pueden instalarse en cualquier momento dentro de un cierto período de tiempo, y los costes durante este tiempo se consideran los que tienen lugar al comienzo del periodo de tiempo. Y, para poner solución a la planificación de largo plazo, en su función objetivo, se combina la función objetivo de cada periodo de tiempo.

Estos mismos autores, en el mismo año aplicaron la Descomposición acelerada de Benders [31], técnica de optimización avanzada utilizada para abordar problemas de gran escala que involucran múltiples variables y restricciones. Esta técnica trata de resolver modelos de programación lineal entera mixta mediante la división del problema en subproblemas más pequeños

y manejables. En su artículo, resumen que diferentes autores han adaptado la descomposición acelerada de Benders para resolver problemas de diseño de redes. Sin embargo, proponen modelos de diseño de red sin capacidad y sin las restricciones de ubicación de línea y enrutamiento de demanda que se consideran en su artículo. No obstante, a pesar de plantear cuatro variantes del método de descomposición, recogen en la conclusión que resolver este tipo de problemas, cuando el tamaño de estos es elevado, es una tarea difícil.

El artículo de García-Archilla et al. (2011) [14] se presenta un modelo de Programación Lineal Entera, que tienen en cuenta restricciones de capacidad, pudiendo de esta forma determinar las frecuencias, abordando además la propiedad de robustez. Realizan una simplificación del problema, diseñando solo la red de infraestructuras, es decir, estaciones y enlaces que las conectan, y el diseño de las líneas (junto sus frecuencias) se realiza en una fase posterior, siendo su función objetivo maximizar la cobertura de viaje. Proponen como método de resolución del problema un algoritmo tipo GRASP (Greedy Randomized Adaptive Search Procedure). Este tipo de algoritmos se ha utilizado ampliamente para resolver problemas de optimización a gran escala desde el trabajo pionero de Feo y Resende (1989) [13]. Demostraron que los algoritmos GRASP son buenos y encuentran diseños de redes factibles de alta calidad en un tiempo de cómputo muy reducido, comparado con los resultados obtenidos por el solver de programación matemática. No obstante, a pesar de las simplificaciones, tanto los problemas de diseño redes como de planificación, fueron imposibles de manejar para instancias realistas.

Como se menciona anteriormente, los modelos de programación matemática para la planificación de líneas pertenecen a dos grupos: los relacionados con la empresa operadora cuya función objetivo suele depender de los costes, y los relacionados con los usuarios cuyos objetivos incluyen espera, tiempo en el vehículo, tiempo de transferencia y viajes directos de pasajeros. El actual documento mantiene su campo de estudio desde la perspectiva del usuario, es por ello que, hasta el momento, se han descrito sólo algunos artículos que se enfocan en el usuario. Pero como se introduce anteriormente, la perspectiva del operador no debe quedar olvidada, es por esta razón que se menciona a continuación algunas de las investigaciones recientes sobre el tema. Ramos, F.L (2014) [33], en su tesis desarrolla un modelo que ayuda a los operadores a tomar decisiones estratégicas y de planificación para redes de transporte público. Además, en esta tesis podemos ver un amplio estudio sobre la segunda fase de la planificación del transporte público, establecimiento de horarios, lo cual supone la frecuencia de los trenes. No obstante, trabajos que anteceden al mencionado anteriormente son la investigación de Baaj, 1991 [3] que desarrolló un enfoque basado en inteligencia artificial que consta de tres componentes principales. El primero involucra un algoritmo de construcción de líneas que equilibra los costes tanto para los usuarios como para los operadores. El segundo componente es un método de análisis de líneas que asigna frecuencia a las líneas, y el tercer componente optimiza la ruta de las líneas generada en el primer componente para que resulten más manejables desde la perspectiva del operador.

Por otro lado, destacamos el artículo de Gutiérrez-Jarpa et al. (2017) [17], donde se consideran tres funciones objetivo: coste, tiempo ahorrado y patrocinio. Utiliza una metodología diferente basada en la combinación de alineamientos. Proporciona la red de metro, aplica un modelo de programación lineal entero bi-criterio, basado en el enfoque de corredores para maximizar el tráfico capturado en presencia de un modo de transporte alternativo.

No obstante, podemos ver reflejado cierto equilibrio entre los pasajeros y las decisiones operativas en el modelo de diseño de redes en el artículo de Canca et al. (2019) [6]. Plantean un problema integrado que incorpora una asignación de múltiples pasajeros junto con el diseño de la red y la configuración de frecuencia y tipo de tren, mediante un procedimiento exacto, teniendo en cuenta la capacidad y modo de transporte alternativo. El procedimiento de solución es una técnica metaheurística que consta de dos niveles, el primero resuelve el problema de diseño de la red y el nivel inferior determina con precisión las frecuencias, los tipos de trenes, los flujos de pasajeros y los tamaños de flotas.

A continuación, se presenta una tabla (Tabla 1) a modo de resumen de las principales revisiones descritas, clasificadas por autor, objetivo, enfoque del artículo, función objetivo que se ha utilizado en el modelo matemático diseñado para dar solución al problema de diseño de líneas o redes de transporte de tránsito rápido, si en este modelo se ha tratado el tema de la capacidad o se ha

tenido en cuenta otro medio de transporte alternativo que competía con el diseñado. Y, por último, otras características relevantes de cada artículo.

Tabla 1: Revisión bibliográfica sobre los problemas del diseño de redes de transporte de tránsito rápido.

Autor	Objetivo	Enfoque	Función objetivo	Capacidad/Otros medios de transporte	Otras Características
Daufroud, 1996	Ubicación de una línea de tránsito rápido	Procedimiento de construcción codicioso y búsqueda tabú	Maximizar la población cubierta por estaciones	No	Tiene en cuenta el espaciado máximo y mínimo, y número de estaciones permitidas.
Laporte, 2005	Diseño de una línea de transporte	Enfoque de ubicación de estaciones principales y rutas implícitas	Maximización de la demanda total O/D cubierta por la línea de transporte	Transporte alternativo	Describe heurísticas para la construcción de una línea. Propone tres enfoques para medir la cobertura de la línea. Uso de la función Logit.
Laporte, 2007	Planificación de tránsito rápido	Determinación de rutas y ubicación de estaciones	Maximización de la cobertura de la línea	Transporte alternativo	Identificación de corredores y trayectos
Bruno, 2002	Diseño de líneas de tránsito rápido	Tercera fase de Laporte (2007). Ubicaciones estaciones secundarias.	Maximización de la población total cubierta	No	Propone una estrategia de búsqueda rápida y sólida para optimizar alineaciones
Marín, 2009	Diseño de Redes de Tránsito Rápido	Diseño de red y ubicación de estaciones. Incorporación del comportamiento del usuario en diseño de red	Maximización de transporte público y reducción de congestión privada	Sí, ambas	Modelo de Logit para elección de línea. Estudio para abordar el problema de diseño de varias líneas.
Marín y Jaramillo 2008	Expansión de capacidad de red	Planificación considerando los cambios en datos a lo largo del tiempo	Integración de decisiones en periodos de tiempo	Si	Aplica descomposición acelerada de Benders en diseño de redes.
García-Archilla et al., 2011	Diseño de Redes de Tránsito Rápido	Uso de algoritmo GRASP para diseño de red	Maximización de cobertura	Sí, ambas	Emplea algoritmo GRASP para diseño de red.
López Ramos, 2014	Planificación de transporte	Enfoque basado en inteligencia artificial para toma de decisiones estratégicas y planificación	Optimización de operaciones de transporte público	Sí, ambas	Aborda la toma de decisiones para operadores y establecimiento de horarios.
Baaj, 1991	Planificación de transporte	Algoritmo basado en IA para diseño de línea	Equilibrio de costes para usuarios y operadores	Capacidad	Utiliza algoritmo de construcción de líneas y optimización de rutas.
Gutiérrez-Jarpa et al., 2017	Diseño de Redes de Tránsito Rápido	Enfoque basado en corredores	Coste, tiempo ahorrado, patrocinio	Transporte alternativo.	Consideración de modo de transporte alternativo
Canca et al., 2019	Diseño de Redes Ferroviarias	Enfoque en cobertura de población y viajes	Maximización del tráfico capturado	Sí, ambas.	Analiza dos objetivos principales para diseño de líneas. Proporciona frecuencias precisas.

Debido al alto coste de construcción de una red de transporte de tránsito rápido o metro, es importante prestar atención a los problemas que afectan la eficacia y robustez al planificar nuevas líneas. Ya que las redes ferroviarias no siempre funcionan según lo previsto debido a la incertidumbre en los datos de entrada, eventos climáticos o incluso posibles ataques intencionales, es crucial que el diseño de la red sea no solo eficiente, sino también resistente y robusto. Por lo tanto, en los dos siguientes apartados se presenta una breve introducción a los sistemas de transporte robustos y a la evaluación de la eficacia, elementos que han sido objeto de investigación.

2.2. Robustez de una red de tránsito rápido.

Según la definición del Institute of Electrical and Electronics Engineers (1990) la robustez se puede definir como el grado en que un sistema o componente puede funcionar correctamente en presencia de entradas inválidas o condiciones ambientales estresantes. Según García-Archilla et al. (2011) [14], en el contexto que nos ocupa, una red se dice que es robusta cuando logra mantener su funcionalidad de manera cuasi-óptima incluso en el caso de que uno de sus enlaces falle. Es por esta razón que la robustez debe tenerse en cuenta en la etapa de planificación. La red puede fallar debido a eventos imprevistos, como accidentes, condiciones climáticas extremas, averías técnicas u otros factores que podrían afectar su operación normal. No obstante, también se ve afectada por la estocasticidad del flujo de pasajeros, que se relaciona con

la imprevisibilidad en la cantidad de pasajeros que utilizarán el sistema, dado que la demanda del transporte puede fluctuar debido a factores impredecibles, y, por lo tanto, afectar a la operación de la red.

La robustez, se ha tratado a lo largo de investigaciones, identificando similitudes entre las soluciones óptimas para diferentes escenarios, o buscando una solución que permanezca viable incluso para los parámetros del peor escenario. Un ejemplo de modelo robusto de diseño de redes de tránsito rápido que se ha mencionado anteriormente es el de Laporte et al. (2011) [21], que presenta un modelo en el que se introducen restricciones de robustez que no solo proporciona rutas alternativas en caso de que falle el enlace, sino que también ayuda a evitar la congestión en un conjunto restringido de enlaces.

Otro problema que afecta a la robustez se relaciona con las perturbaciones en las operaciones normales. El artículo de A. De-Los-Santos et al. (2012) [1] considera la robustez desde el punto de vista de los pasajeros en presencia de interrupciones. La función auxiliar aplicada para definir las medidas de robustez es el tiempo total de tránsito de los pasajeros. Se consideran dos casos: en el primer caso, los pasajeros afectados por la interrupción deben esperar a que se repare el fallo o tomar una ruta alternativa en la misma red; en el segundo caso, el operador proporciona un servicio de autobús puente. Un ejemplo es el sistema trenes de cercanías de Madrid ilustra la aplicabilidad de los índices de robustez desarrollados por los autores.

2.3. Formas de evaluar una red de transporte de tránsito rápido.

Para garantizar el funcionamiento eficiente de un sistema de metro, es importante evaluar la vulnerabilidad de la red de metro a posibles interrupciones e identificar líneas cuyos incidentes puedan tener un impacto crucial tanto en las redes de metro como en los viajeros. La vulnerabilidad y la resiliencia son dos índices ampliamente utilizados para medir el rendimiento de la red.

La vulnerabilidad de la red en el sistema de transporte se define como la susceptibilidad a incidentes que podrían ocasionar una notable disminución en la capacidad de funcionamiento de dicha red. En el ámbito de una red de metro, la resiliencia está vinculada a la capacidad de enfrentar de manera efectiva situaciones imprevistas con una disminución aceptable en el rendimiento operativo. Esta disminución por lo general se mide mediante la reducción de la capacidad y los esfuerzos destinados a recuperarse de las interrupciones. Por lo tanto, se puede decir que la vulnerabilidad tiene más que ver con la susceptibilidad de un sistema, y la resiliencia se refiere más a la respuesta de un sistema.

En diferentes revisiones podemos encontrar investigaciones que se centran en el flujo de pasajeros y el tiempo/distancia de viaje, enfoques basados en cuestiones topológicas, flujo o el coste para medir la vulnerabilidad. Otras investigaciones se centran desde el punto de vista del funcionamiento de la línea,

basado en el comportamiento del metro si se considera una línea no operativa durante una interrupción.

Según Mesa (2015) [31], una red de tránsito rápido existente se puede evaluar mediante medidas e indicadores de la red, pero las mismas medidas también se pueden usar para evaluar redes potenciales, en particular aquellas que resultan del proceso de combinar corredores. Además, estas medidas se pueden utilizar como funciones objetivo o como restricciones en modelos de programación matemática.

Musso y Vuchic (1988) [32] han formulado varios indicadores que analizan la topología de la red, incluyendo la disponibilidad de círculos, la complejidad de la estructura y la conectividad. Además, han evaluado medidas relacionadas con el servicio y los indicadores de utilización. Por otro lado, Laporte et al. (1997) [22] evaluaron la eficacia de las redes de tránsito rápido empleando medidas como pasajeros/red y pasajeros/plano. Por ejemplo, estos investigadores demostraron que, en una ciudad de forma circular, los diseños con forma de triángulo y rueda (conocidos como “cartwheel”) son más deseables que los diseños en forma de estrella (Ilustración 5) en términos de conectividad y dirección de viaje. En otro estudio, Saidi et al. (2016) [34] crearon un modelo analítico para determinar el número óptimo de líneas radiales en una disposición anillo-radial, que puede ser vista como una generalización de la forma de rueda (cartwheel), donde las líneas radiales no necesariamente se cruzan en un punto único.

Gattuso y Miriello (2005) [15] proporcionan un análisis comparativo de 13 redes de metro existentes en relación con 10 indicadores. Lee et al. (2008) [26] analizaron la red de metro de Seúl en términos de longitud característica del camino, radio, diámetro, coeficiente de agrupamiento, eficiencia de la red, peso de los bordes, fortaleza de los nodos y árbol de flujo máximo que abarca. Otros indicadores como regularidad, disponibilidad de servicio, puntualidad y confiabilidad se pueden encontrar en UITP, The International Association of Public Transport (2011) [39]. En la actualidad, los valores de algunos de estos indicadores a menudo se presentan en los informes técnicos de las empresas operadoras.

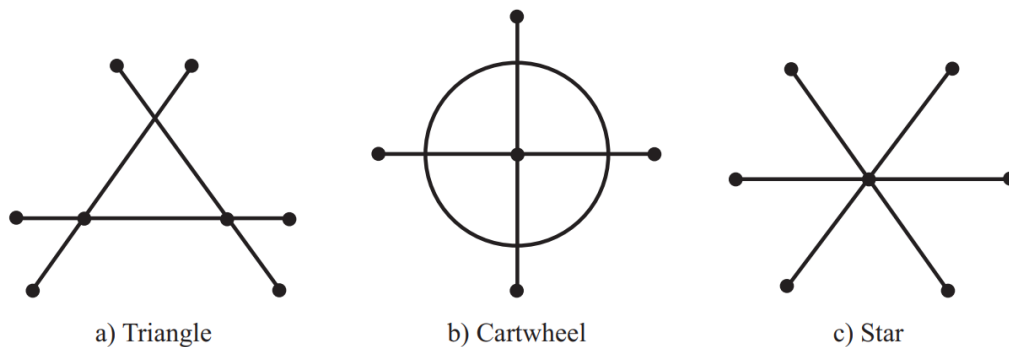


Ilustración 5: Los tres tipos de diseño de metro. Según Mesa (2015) [12]

En los últimos años hemos sido testigos de la aparición de nuevos índices basados en la evaluación de las redes de transporte desde el ángulo de la teoría de redes complejas y la robustez. En un estudio del sistema de metro de Sun y Guan (2016) [37] encontraron que las líneas de metro que transportan grandes volúmenes de pasajeros tienen un mayor impacto en la vulnerabilidad del sistema y que las líneas con una forma topológica circular tienen un alto impacto en la redistribución del flujo de pasajeros en caso de interrupciones.

3. MODELOS MATEMÁTICOS PARA EL DISEÑO DE UNA RED DE TRÁNSITO RÁPIDO.

El presente capítulo del trabajo propone dos modelos de programación matemática para el diseño de líneas de tránsito rápido en una ciudad. Como se ha comentado anteriormente, los modelos matemáticos de localización se pueden complicar mucho, y la mayoría de ellos son intratables para escenarios realistas debido a limitaciones computacionales. Es por esto, que los dos modelos matemáticos que se presentan a continuación se evaluarán con una base de datos manejables, y tratarán localizar las estaciones que se deberán construir para maximizar la demanda de pasajeros. Además, se introducirán funciones y restricciones, que permitan proporcionar el diseño de líneas que optimicen el tiempo de viaje (coste) a un elevado número de pasajeros que

respete las restricciones técnicas, y presupuestarias que se describirán en cada modelo.

Los algoritmos de resolución propuestos en este documento son exactos, más concretamente, modelos de programación matemática lineal entera mixta. Ambos devuelven una solución óptima garantizada, es decir, se demuestra que no existe otra solución con mejor valor objetivo (siempre y cuando el solver disponga de suficiente tiempo).

3.1. Modelo matemático 1.

El primer modelo matemático para el diseño de una red de tránsito rápido busca una solución óptima que maximiza el número de viajes captados, teniendo en cuenta otros medios de transporte alternativo.

El modelo que a continuación se describe se basa en Laporte et al. (2011) [24] y en Laporte y Mesa (2019) [6], así como en el mencionado trabajo de Manzanares Abásolo M.A (2022) [27] sin tener en cuenta los distintos efectos de la sobreocupación.

3.1.1. Datos y notación.

En este modelo se asume que el dato del número de pasajeros que viajan desde un origen a un destino viene dado. Por otro lado, también se asume que la localización de las distintas estaciones potenciales se conoce.

1. En primera instancia, el modelo seleccionará una ruta para la línea de metro. Las posibles estaciones o paradas $N = \{1, \dots, n\}$ y las conexiones potenciales entre las mismas $E \subset N \times N$, formarán una red subyacente $N = (N, E)$. Para cada arista no dirigida $e = \{i, j\}$ se consideran los arcos (dirigidos) $a = (i, j), \hat{a} = (j, i)$. La línea constará de dos extremos, uno se encontrará en el conjunto $O \subset N$, y el otro en $D \subset N$. Denotaremos $\delta(k)$ al conjunto de aristas incidentes en el nodo k , y $\delta^-(k)$ y $\delta^+(k)$ al conjunto de arcos entrantes y salientes de k , respectivamente.
2. Cada arco tiene asociado una distancia, d_a , la cual ha sido aproximada con la distancia euclídea entre las estaciones, y será calculada mediante las coordenadas de cada nodo o estación posible a construir en la ciudad de Sevilla, será un dato de entrada. No obstante, las distancias reales entre estaciones en todo momento no podrán ser distancias rectas, ya que habrá factores que considerar, como, pasos de ríos, montañas o parques naturales que no permitan edificar y obstaculicen el paso del transporte. El parámetro d_a puede ser también interpretado como el tiempo, o coste, necesitado para atravesar el arco $a \in A$.

3. La estructura de coste incluye el coste de construir una estación en el nodo $i \in N$ y cada eje $e \in E$, siendo c_i el coste de construir una estación en el nodo i , y , c_e el coste de construir una arista, el cual viene definido por la expresión $c_e = d_a \cdot c_v$, que es el resultado del producto de la distancia entre arcos y el coste base de construir una vía, los cuales serán definidos en el Capítulo 4, dado que forman parte de los datos utilizados para obtener los resultados de los modelos. El modelo también se verá limitado por c_{max} , el coste máximo disponible o presupuesto.

4. La demanda es por pares y se representa por el conjunto $W \subset N \times N$. Cada par $w = (w^s, w^t) \in W$ tiene asociado una demanda g_w y representa el número de viajeros de w^s a w^t . Por lo tanto, podemos llamar $G = (g_w)$ a la matriz origen-destino.

5. Este modelo tiene en cuenta un modo de tránsito alternativo. La utilidad por ese modo para el par w viene dado por la matriz $U = (u_w)$. Supondremos que el par w será captado por la línea si su utilidad usando esta línea es mejor que u_w . Por ejemplo, que el tiempo de viaje por la línea sea menor al tiempo de viaje por el modo alternativo.

6. Por último, denotamos t_a el tiempo del vehículo al recorrer el arco a , t_k el tiempo de parada en la estación k , y t_{wait} el tiempo de espera en el andén.

3.1.2. Variables.

- $y_i = 1$, si la estación $i \in N$ se construye, 0 en caso contrario.
- $x_a = 1$, si la vía que va de la estación i a j se construye, 0 en caso contrario.
- $f_a^w = 1$, si la demanda w atraviesa el arco a , 0 en caso contrario.
- $z_w = 1$, si la línea captura el par de demanda w , y 0 en caso contrario.

3.1.3. Función objetivo y restricciones.

- El objetivo del modelo será maximizar la demanda captada.

$$\text{Max} \sum_{w \in W} g_w z_w \quad (1)$$

- Restricción de presupuesto. La suma del coste de construir una estación, multiplicado por la variable que define si la estación existe o no, más la suma del coste de construir un tramo, multiplicado por la variable que define si el tramo se va a construir o no, tiene que ser menor o igual que el presupuesto máximo disponible para construir la línea de metro.

$$\sum_{i \in N} c_i y_i + \sum_{e \in E} c_e x_e \leq c_{max} \quad (2)$$

- Restricciones de diseño.

$$x_e \leq y_i, \quad e \in E, i \in e \quad (3)$$

$$\sum_{o \in O} y_o = 1, \quad \sum_{d \in D} y_d = 1 \quad (4)$$

$$\sum_{e \in E} x_e + 1 = \sum_{i \in N} y_i \quad (5)$$

$$\sum_{e \in \delta(k)} x_e = 2y_k, \quad k \in N \setminus (O \cup D) \quad (6)$$

$$\sum_{o \in O} \sum_{e \in \delta(o)} x_e = 1, \quad \sum_{d \in D} \sum_{e \in \delta(d)} x_e = 1 \quad (7)$$

La restricción (3) indica que, si se construye una vía, se deben construir las estaciones adyacentes a ella. Además, de todas las posibles estaciones, tiene que haber una que sea origen y una que sea destino de la línea, lo que queda reflejado en la restricción (4). Por otro lado, para conseguir una estructura de la línea coherente y conexa sin formar bucles o ciclos, el número de estaciones ha de ser igual al número de aristas más uno. Esto último lo asegura la restricción (5). En el diseño de líneas de un sistema de transporte, a veces, encontramos definida esta restricción como conseguir una estructura de árbol.

La restricción (6) asegura que el diseño de la estructura de la línea sea un “camino”, es decir, que la línea no tenga bifurcaciones ni intersecciones, sino que sea una ruta directa y unidireccional desde su inicio hasta su destino.

Por último, la restricción (7), impone que de la estación origen solo puede salir una arista, y al destino solo llega una.

- Conservación de flujo.

$$\sum_{a \in \delta_w^+(i)} f_a^w - \sum_{a \in \delta_w^-(i)} f_a^w = \begin{cases} z^w, & \text{si } i = w^s \\ -z^w, & \text{si } i = w^t \\ 0, & \text{en otro caso} \end{cases} \quad w \in W, i \in N \quad (8)$$

La línea de transporte debe garantizar un equilibrio en el flujo de pasajeros en la red, y se modela con la restricción (8), que presenta tres casos distintos. Si el par no es captado, la variable que representa la demanda, z^w , valdrá cero. No obstante, cuando el par es captado, el número de arcos que salen de un nodo menos el número de arcos que entran en el nodo tomarán el valor 1, -1 ó 0, dependiendo de si i es el nodo origen del par, el nodo destino o un nodo intermedio.

1. En el primer caso, se considera el nodo origen del par.

La diferencia entre el flujo entrante y el flujo saliente en

este nodo debe ser igual a la variable de captación z asociada a la demanda w .

2. El segundo caso se refiere al nodo destino del par. En este caso, la diferencia entre los flujos entrantes y salientes en este nodo debe ser igual a la variable de captación z asociada a la demanda w , pero con signo positivo.
3. El último caso se aplica a cualquier otro nodo de la red. En estos nodos, se requiere que el flujo sea igual a cero. Por lo tanto, el número de arcos entrante debe ser igual al número de arcos saliente.

De esta forma, aseguramos que el flujo de aristas esté en equilibrio en cada nodo de la red.

- Restricciones de localización.

$$f_a^w + f_{\bar{a}}^w \leq x_e, \quad e = \{i, j\} \in E, a = (i, j), \quad w \in \{w^s, w^t\} \quad (9)$$

$$\sum_{a \in \delta^+(w^s)} f_a^w \leq y_{w^s}, \quad \sum_{a \in \delta^-(w^t)} f_a^w \leq y_{w^t}, \quad w \in \{w^s, w^t\} \quad (10)$$

$$z_w \leq y_k, \quad k \in \{w^s, w^t\}, \quad w \in \{w^s, w^t\} \quad (11)$$

Para modelar la relación entre la demanda de transporte y la activación de la variable binaria, f_w , que indica la utilización de las conexiones específicas y permite capturar la presencia de flujo, se introduce la restricción (9) y la (10). La restricción (9) establece que, si el par w usa un arco, se debe construir la correspondiente arista. Así mismo, la restricción (10) vincula la construcción de una estación en el origen de un par de demanda con la presencia de flujo en la red, de tal forma que si se construye una estación en el origen de un par puede salir flujo, y si por el contrario no se construye, no puede salir flujo. Por último, se añade la restricción (11), la cual indica que, si un par va a utilizar la red, han de construirse las estaciones de origen y destino de ese par.

- Elección de modo.

$$\sum_{a \in A} t_a f_a^w + \sum_{k \in N \setminus \{O, D\}} t_k \sum_{a \in \delta^+(k)} f_a^w + t_{wait} z^w \leq u^w, w \in W \quad (12)$$

Esta última restricción se introduce para asegurar que el tiempo total de viaje para los pasajeros que utilizan la línea de transporte es menor que el tiempo del modo alternativo de transporte con el que compete, es por esta razón que se establece que la suma de los productos de flujos de pasajeros por cada arista multiplicados por el tiempo en recorrer al arco a y el tiempo de parada en la estación k , con la suma

del tiempo de espera correspondiente debe ser menor que el transporte alternativo. En otras palabras, si el tiempo de viaje total es mejor que u^w , entonces el par de demanda w será captado, ya que ofrecería un tiempo de viaje más eficiente en comparación con la alternativa.

3.2. Modelo matemático 2.

El diseño de líneas de transporte de tránsito rápido, como hemos visto, es una tarea difícil, que desemboca en modelos matemáticos complejos de resolver computacionalmente.

Un enfoque que puede simplificar el modelo matemático para el diseño de líneas es la maximización de la cobertura de la población que reside en las proximidades del sistema de metro, ya estudiado por varios autores como hemos visto en el capítulo anterior, en lugar de maximizar la cobertura por pares de nodos, función más compleja y utilizada en el modelo de la sección anterior. Esta aproximación se enfoca en brindar un acceso eficiente y conveniente a un mayor número de personas, lo que a su vez puede conducir a un mayor uso del sistema de transporte público y una reducción en la congestión vial.

Investigaciones anteriores demuestran que se encuentran soluciones más realistas si el modelo matemático maximiza la cobertura según la demanda origen-destino. No obstante, este segundo modelo matemático intenta simplificar este tipo de modelos mediante la maximización de la cobertura de la población. En este sentido, el trabajo de Laporte (2005) [25] ofrece una perspectiva valiosa al establecer una relación entre el número esperado de viajes y la población beneficiada por el sistema, es decir, personas que viven o trabajan en las áreas circundantes a las líneas de transporte. Al maximizar la cobertura de la población, se garantiza que un mayor número de personas pueda acceder y aprovechar los beneficios del sistema de tránsito rápido. Además, de simplificar la complejidad del modelo matemático al enfocarse en la conectividad de las líneas con las áreas pobladas, facilita la toma de decisiones y la planificación del sistema de transporte.

A continuación, se encuentra detallado el modelo en cuestión, describiendo primero los datos y notación, las variables, la función objetivo, y restricciones.

3.2.1. Datos y notación.

El diseño del Modelo Matemático 2 se fundamenta en los datos presentados en el Modelo 1, siendo los mismos las estaciones o paradas N , las conexiones E , y el conjunto de vías de la línea, siendo el conjunto de arcos dirigidos A . Por otro lado, la estructura del coste de construcción de estaciones

c_i , y vías c_e son las mismas que el Modelo 1, así como su presupuesto máximo c_{max} y la distancia o coste asociado a cada arco d_a .

1. La demanda en este modelo será representada por el número de habitantes que viven cerca de la estación (nodo) i , denotado como h_i , lo cual formará parte también de los datos de entrada del problema, los cuales han sido estimados según los Indicadores demográficos de Sevilla [11], en el que los servicios de estadística dividen la demografía de la ciudad por sus diferentes barrios (ver Anexo 3).
2. Por último, los grupos de estaciones que contendrán aquellos nodos que se deseen que pertenezca al grupo de estaciones serán definidos de la misma forma que en el Modelo 1.

3.2.2. Variables

- $y_i = 1$, si la estación $i \in N$ se construye, 0 en caso contrario.
- $x_a = 1$, si la vía $a = (i, j)$ se construye, 0 en caso contrario.

3.2.3. Función objetivo y restricciones

- Función objetivo.

$$Max \sum_{i \in N} h_i y_i \quad (1)$$

La función objetivo busca localizar dónde situar las estaciones del metro para maximizar el total de habitantes capturados.

- Restricción de presupuesto. Esta restricción limita la construcción de estaciones y vías, teniendo que ser la suma del coste de construir una estación por su variable, más la suma del coste de construir las vías multiplicado por su variable y por la distancia que recorrerá, debe ser menor o igual que el presupuesto o coste máximo, que será definido como un dato de entrada.

$$\sum_{i \in N} c_i y_i + \sum_{e \in E} c_e x_e \leq c_{max} \quad (2)$$

- Restricciones de diseño.

$$x_e \leq y_i, \quad \forall e \in E \quad (3)$$

$$x_e \leq y_j, \quad \forall e = (i, j) \quad (4)$$

$$\sum_{e \in E} x_e + 1 = \sum_{i \in N} y_i \quad (5)$$

Las restricciones (2) y (3) establecen una relación entre la construcción de las vías y estaciones correspondientes. Las restricciones (2) y (3) implican que, si se construye la vía entre dos nodos, también es necesario construir las estaciones asociadas a estos nodos.

Por otro lado, la restricción (5) como se explica en el Modelo 1, asegura que la línea sea coherente, restringiendo que el número de estaciones deber ser igual que el número de aristas más uno.

- Restricciones de localización.

$$\sum_{e \in E} x_e = y_i \quad ; \quad \sum_{i \in N} y_i = 1 \quad \forall i \in O \quad (6)$$

$$\sum_{e \in E} x_e = y_j \quad ; \quad \sum_{i \in N} y_j = 1 \quad \forall j \in D \quad (7)$$

$$\sum_{i \in N} x_{ik} = \sum_{j \in N} x_{kj} \quad \forall k \in K \quad (8)$$

Por último, se añaden tres restricciones que dividen las estaciones en tres grupos, las de partida del metro (O), las intermedias (K) y las finales (D). Las ecuaciones (6) y (7), imponen que número total de estaciones en ambos grupos es 1, es decir, habrá una única estación inicial y otra final. Por el contrario, la ecuación (8) define que por cada estación construida dentro del grupo K, debe salir y entrar una vía.

4. EXPERIMENTOS. RESULTADOS Y ANÁLISIS.

En este capítulo, se presentan los resultados y el análisis derivados de los diversos ensayos realizados con los dos modelos matemáticos de programación aplicados al diseño de una línea de transporte de tránsito rápido que se desarrollaron en el capítulo anterior. Recordamos que primer modelo busca maximizar la captación de pares origen-destino como solución, mientras que el segundo modelo maximiza en la cantidad de habitantes por estación (nodo) construida.

4.1. Datos introducidos.

La base de datos introducida en los modelos se caracteriza por una serie de coordenadas donde se situarían las estaciones, acompañadas de los habitantes por cada zona, y los arcos (vías) posibles que unen las estaciones. Esta base de datos se expandirá gradualmente con el propósito de evaluar los modelos.

La base de datos utilizada para probar los modelos parte de coordenadas reales y estimaciones de población para la ciudad de Sevilla. Se llevó a cabo un estudio previo para identificar los posibles lugares edificables en la ciudad, identificando un total de 28 ubicaciones (ver Ilustración 6). Además, se incorporaron datos de población por barrio proporcionados por el Ayuntamiento de Sevilla [2]. El objetivo fue reflejar de manera precisa la situación real en la ciudad de Sevilla.

No obstante, como se ha introducido anteriormente, la base de datos se reduce y se aumenta poco a poco, con el fin de ver cómo reaccionan ambos modelos con distintas cantidades de información (ver Anexo 3).

Por otro lado, los modelos fueron probados con una serie de datos estimados que se mantienen constantes en las diferentes pruebas: el coste por

estación, $c_i = 20 M$, el coste base de la construcción de las vías $c_v = 50 M$ y el coste máximo o presupuesto $c_{max} = 15\ 000 M$.

Dentro de los grupos de estaciones creados, los diferentes escenarios consideran que la última estación es la número 28, que corresponde con el aeropuerto, ya que se puede prever que agilizaría el tráfico y facilitaría el acceso de los ciudadanos. Esta restricción se añade para obtener una red con sentido en el Modelo 2, dado que, al optimizar el número de habitantes por zona captados, se eliminaría la opción de una posible estación cerca del aeropuerto. (Ver Ilustración 6).

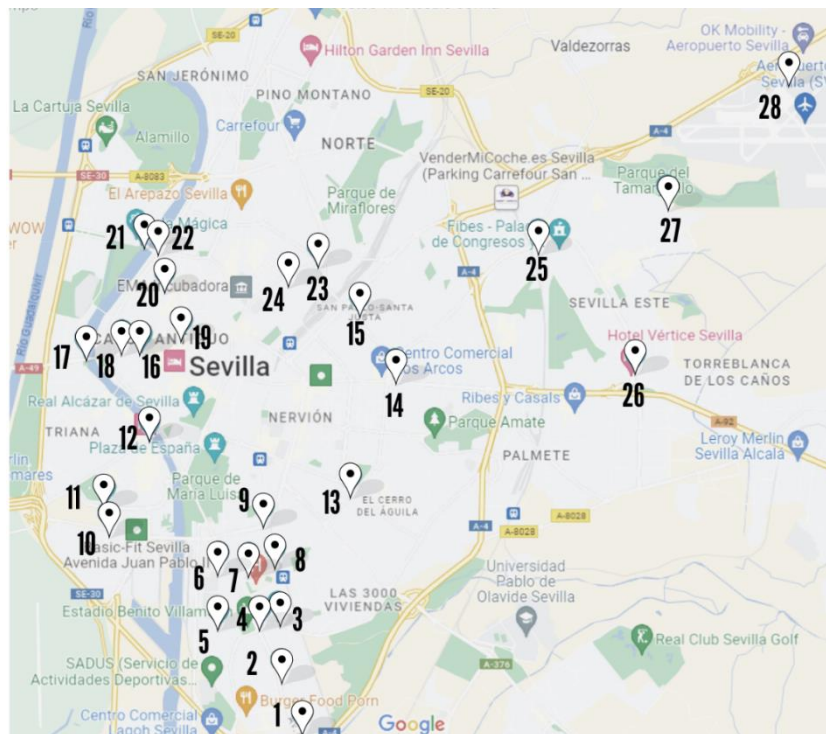


Ilustración 6: Localización de posibles estaciones para el diseño de una línea de metro en la ciudad de Sevilla.

El primer modelo de programación matemática, como se puede ver en el capítulo anterior, necesita más datos de entrada, como los tiempos o costes de recorrer cada arco (t_a), de parada (t_k) y de espera (t_{wait}), la matriz Origen-Destino (G) y la matriz del modo alternativo de transporte (U). El tiempo de parada y de espera han sido estimados, siendo; $t_k = 4$ minutos y $t_{wait} = 6$ minutos. El tiempo que tarda en recorrer cada arco se ha definido proporcional a la distancia, siendo un kilómetro igual a un minuto, es por esto que; $t_k = d_a$. Se debe destacar, que las dos matrices y los costes de recorrer cada arco han sido calculados en el modelo de programación según la base de datos introducida, por lo que, si aumenta, estos datos aumentarán en consecuencia.

A continuación, se muestran las matrices correspondientes a la distancia o costes de arcos d_a , la matriz Origen-Destino G , y aquella que describe el coste del método de transporte alternativo U . Las matrices que se presentan corresponden a la simplificación de la base de datos de las 28 estaciones, en grupos de 6 y 9 estaciones, que son las que se utilizarán en ambos modelos matemáticos para poder comparar sus resultados.

$$d_a = \begin{bmatrix} 0 & 8 & 15 & 15 & 19 & 24 \\ 8 & 0 & 7 & 7 & 12 & 17 \\ 15 & 7 & 0 & 2 & 9 & 11 \\ 15 & 7 & 2 & 0 & 7 & 10 \\ 19 & 12 & 9 & 7 & 0 & 7 \\ 24 & 17 & 11 & 10 & 7 & 0 \end{bmatrix}$$

d_a Para una base de datos de 6 estaciones.

$$d_a = \begin{bmatrix} 0 & 8 & 15 & 15 & 19 & 24 & 22 & 21 & 113 \\ 8 & 0 & 7 & 7 & 12 & 17 & 14 & 14 & 110 \\ 15 & 7 & 0 & 2 & 9 & 11 & 7 & 7 & 107 \\ 15 & 7 & 2 & 0 & 7 & 10 & 7 & 8 & 109 \\ 19 & 12 & 9 & 7 & 0 & 7 & 9 & 12 & 114 \\ 24 & 17 & 11 & 10 & 7 & 0 & 5 & 9 & 110 \\ 22 & 14 & 7 & 7 & 9 & 5 & 0 & 4 & 106 \\ 21 & 14 & 7 & 8 & 12 & 9 & 4 & 0 & 102 \\ 113 & 110 & 107 & 109 & 114 & 110 & 106 & 102 & 0 \end{bmatrix}$$

d_a Para una base de datos de 9 estaciones.

La matriz Origen-Destino, $G = (g^w)$, ha sido definida mediante las distancias entre nodos. Los valores de este conjunto de datos se han aproximado mediante la división de 1000 entre la distancia entre nodos, es decir, la distancia euclídea entre las coordenadas de las posibles estaciones introducidas en la base de datos del modelo. Es importante mencionar que la distancia euclídea calculada es una simplificación útil cuando se trabaja en un contexto donde se asume que los desplazamientos se realizan en un plano bidimensional, como mapas urbanos. No obstante, esta distancia no refleja la realidad, ya que hay situaciones topográficas que hacen imposible construir la línea de metro como una línea recta entre las dos estaciones.

$$G = \begin{bmatrix} 0 & 125 & 66.66 & 66.66 & 52.63 & 41.67 \\ 125 & 0 & 142.85 & 142.85 & 83.33 & 58.82 \\ 66.66 & 142.85 & 0 & 500 & 111.11 & 90.91 \\ 66.66 & 142.85 & 500 & 0 & 142.85 & 100 \\ 52.63 & 83.33 & 111.11 & 142.85 & 0 & 142.85 \\ 41.66 & 58.82 & 90.91 & 100 & 142.85 & 0 \end{bmatrix}$$

G Para una base de datos de 6 estaciones.

$$G = \begin{bmatrix} 0 & 125 & 66.66 & 66.66 & 52.63 & 41.67 & 45.45 & 47.62 & 8.85 \\ 125 & 0 & 142.85 & 142.85 & 83.33 & 58.82 & 71.42 & 71.42 & 9.09 \\ 66.66 & 142.85 & 0 & 500 & 111.11 & 90.91 & 142.85 & 142.85 & 9.34 \\ 66.66 & 142.85 & 500 & 0 & 142.85 & 100 & 142.85 & 125 & 9.17 \\ 52.63 & 83.33 & 111.11 & 142.85 & 0 & 142.85 & 111.11 & 83.33 & 8.77 \\ 41.66 & 58.82 & 90.91 & 100 & 142.85 & 0 & 200 & 111.11 & 9.09 \\ 45.45 & 71.42 & 142.85 & 142.85 & 111.11 & 200 & 0 & 250 & 9.43 \\ 47.62 & 71.42 & 142.85 & 125.0 & 83.33 & 111.11 & 250 & 0 & 9.80 \\ 8.85 & 9.09 & 9.34 & 9.17 & 8.77 & 9.09 & 9.43 & 9.80 & 0 \end{bmatrix}$$

G Para una base de datos de 9 estaciones.

La utilidad por el modo de tránsito alternativo viene dada por la matriz $U = (u^w)$. Matemáticamente, en el modelo, se ha calculado como el doble de distancia que se tardaría en metro, ya que tienden a ofrecer rutas más directas y eficientes, minimizando desvíos y rodeos que se experimentan en las carreteras o vías de la ciudad. Para ambas bases de datos, la matriz $U = (u^w)$ es:

$$U = \begin{bmatrix} 0 & 16 & 30 & 30 & 38 & 48 \\ 16 & 0 & 14 & 14 & 24 & 34 \\ 30 & 14 & 0 & 4 & 18 & 22 \\ 30 & 14 & 4 & 0 & 14 & 20 \\ 38 & 24 & 18 & 14 & 0 & 14 \\ 48 & 34 & 22 & 20 & 14 & 0 \end{bmatrix}$$

U Para una base de datos de 6 estaciones.

$$U = \begin{bmatrix} 0 & 16 & 30 & 30 & 38 & 48 & 44 & 42 & 226 \\ 16 & 0 & 14 & 14 & 24 & 34 & 28 & 28 & 220 \\ 30 & 14 & 0 & 4 & 18 & 22 & 14 & 14 & 214 \\ 30 & 14 & 4 & 0 & 14 & 20 & 14 & 16 & 218 \\ 38 & 24 & 18 & 14 & 0 & 14 & 18 & 24 & 228 \\ 48 & 34 & 22 & 20 & 14 & 0 & 10 & 18 & 220 \\ 44 & 28 & 14 & 14 & 18 & 10 & 0 & 8 & 212 \\ 42 & 28 & 14 & 16 & 24 & 18 & 8 & 0 & 204 \\ 226 & 220 & 214 & 218 & 228 & 220 & 212 & 204 & 0 \end{bmatrix}$$

U Para una base de datos de 9 estaciones.

4.2. Resultados.

Los resultados de las pruebas realizadas podemos ver que recopilan el número de estaciones a construir, las vías, la solución óptima o capacidad captada (z), y el tiempo de cómputo del programa en encontrar la solución (t), en segundos.

Ambos modelos, se ponen a prueba incrementando poco a poco la base de datos que se introduce. En primer lugar, se realiza una prueba en ambos modelos con una base de datos de 6 posibles estaciones (ver Tabla 2).

<i>Tabla 2: Resultado con una base de datos de 6 estaciones.</i>		
Tipo de modelo	Modelo 1	Modelo 2
Nº de estaciones	4	5
Vías	(1, 3), (3, 4), (4, 6)	(2, 3), (3, 4), (4, 5), (5, 6)
z	572,24 viajeros	85 296 habitantes
t(s)	2.95	0.0517

Como podemos observar, los modelos construyen una cantidad diferente de estaciones, con diferentes nodos, y, como era de esperar, diferentes soluciones objetivas (observar solución gráfica en la Ilustración 7).

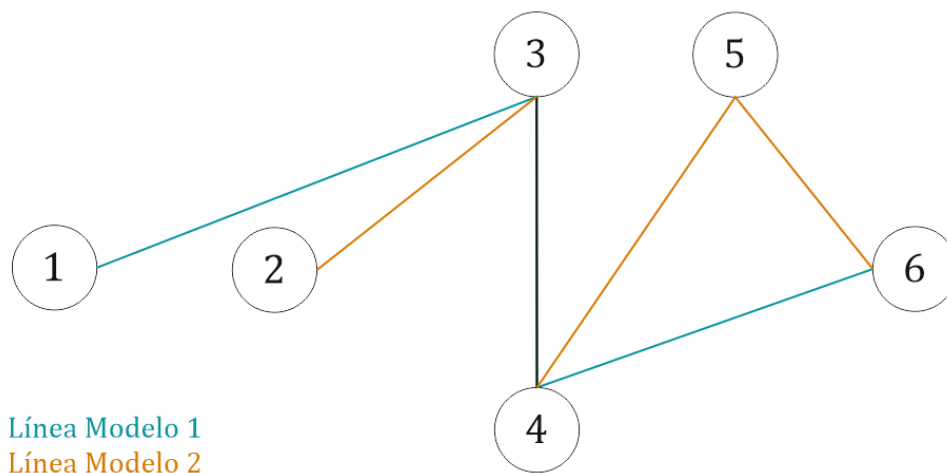


Ilustración 7: Comparación de los diseños de líneas resultantes de los modelos de programación matemática, para 6 nodos. Siendo los tramos negros los coincidentes.

Como se menciona previamente, los modelos tienen objetivos distintos, es por esto, que, ampliamos la tabla anterior para poder compararlos. Para abordar esta limitación, se ha realizado una modificación en el planteamiento del Modelo 1, el cual maximiza la cobertura por pares, y del Modelo 2 que maximiza la cobertura por nodos, con el fin de obtener las soluciones de los modelos que permita una comparación más adecuada. Las Tablas 3 y 5 muestran, respectivamente, la cobertura por nodos que presenta el Modelo 1, y la cobertura por pares del Modelo 2, con la base de datos de 6 y 9 estaciones posibles siendo la conversión de las soluciones de los modelos de programación expuestos.

<i>Tabla 3: Comparación de la cobertura de los modelos para una base de datos de 6 estaciones</i>				
Solución	Modelo utilizado	Cobertura por nodos	Cobertura por pares	Tiempo de cómputo
Solución 1	<u>Modelo 1</u> (maximiza cobertura por pares)	47 081 habitantes	572,24 viajeros	2,95 s
Solución 2	<u>Modelo 2</u> (maximiza cobertura por nodos)	85 296 habitantes	285,71 viajeros	0.0517 s

La conversión de las soluciones revela que a pesar de que el Modelo 2 obtiene una mayor cobertura por nodos (572), la cobertura entre las estaciones que se salen a construir es mucho menor a la del Modelo 1 (285 viajeros). Esta primera prueba revela que probablemente el Modelo 2 no es una buena opción que simplifica el segundo modelo para maximizar la cobertura por pares, ya que sus soluciones son bastante dispares. Además, el Modelo 1 obtiene una

solución con diferentes nodos a construir que el Modelo 2, podemos ver reflejado cómo estaciones que tienen pocos habitantes puede ser demandas por la población, y es importante tener esto en cuenta para un estudio de construcción de líneas de transporte de tránsito rápido.

Por otro lado, si comparamos el tiempo de cómputo entre los modelos, podemos afirmar que la simplificación que se buscaba con el Modelo 2, en cuanto a la dificultad del programa, ha sido adecuada, bajando dos órdenes de magnitud el tiempo de cómputo empleado.

La segunda prueba a la que se somete los modelos es con una base de datos mayor, de 9 estaciones potenciales (ver Tabla 4, Ilustración 8). El propósito de ir aumentando poco a poco la base de datos de entrada, si encuentra solución, es ver cómo se comporta el programa matemático al manejar una cantidad de datos mayor, esto se observa en el tiempo de cómputo.

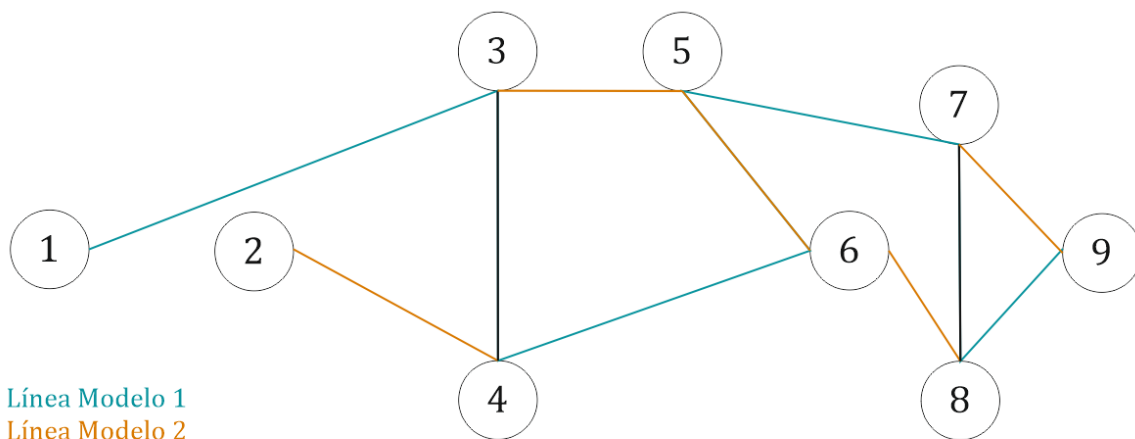


Ilustración 8: Comparación de los diseños de líneas resultantes de los modelos de programación matemática, para 9 nodos. Siendo los tramos negros los

<i>Tabla 4: Resultado con una base de datos de 9 estaciones.</i>		
Tipo de modelo	Modelo 1	Modelo 2
Nº de estaciones	8	8
Vías	(1, 3), (3, 4), (4, 6), (6, 5), (5, 7), (7, 8), (8, 9)	(2, 4), (4, 3), (3, 5), (5, 6), (6, 8), (8, 7), (7, 9)
z	612.27 viajeros	131 265 habitantes
t(s)	376,3808	0.0676

Como se menciona anteriormente, para mejorar la comprensión de las soluciones obtenidas, se realiza una conversión de las mismas para una base de datos de 9 estaciones (Tabla 5).

<i>Tabla 5: Comparación de la cobertura de los modelos para una base de datos de 9 estaciones</i>				
Solución	Modelo utilizado	Cobertura por nodos	Cobertura por pares	Tiempo de cómputo
Solución 1	<u>Modelo 1</u> (maximiza cobertura por pares)	131 220 habitantes	612.27 viajeros	376,3808 s
Solución 2	<u>Modelo 2</u> (maximiza cobertura por nodos)	131 265 habitantes	574.24 viajeros	0.0676 s

Las soluciones, en este caso, no muestran ser tan dispares como la obtenida para una base de datos de 6 estaciones. El Modelo 1 obtiene una mayor cobertura de la línea debido a la optimización de la cobertura por pares y no de habitantes, no obstante, la cobertura de los habitantes es bastante similar a la

del Modelo 2. Siendo diferente el recorrido, encontramos que se construyen el mismo número de estaciones y comparten nodos como el 3, 4, 5, 6, 7, 8 y 9 en sus soluciones.

Otro punto a destacar es el tiempo de cómputo, podemos observar que se ve incrementado notablemente si aumentamos tan solo un poco la base de datos de este primer modelo. Esto se debe a la naturaleza del problema, y el solver utilizado para encontrar la solución, que hace imposible seguir calculando soluciones de una base de datos mayor a la de 9 estaciones para el Modelo 1. Por el contrario, el Modelo 2 encuentra una solución óptima en décimas segundos, dado que, al ser más sencillo, puede manejar de forma más rápida una base de datos más grande. Nótese también que el aumento de tiempo al pasar de 6 a 9 nodos es apenas una décima de segundo.

Hasta el momento, se ha utilizado el solver de optimización de código abierto, *glpk*. Este solver mencionado demuestra no ser suficiente para resolver un modelo matemático en el que se aumentan las variables y restricciones. Es por esta razón que el estudio se ve limitado, y no se puede llegar a comparar los modelos con un número mayor de posibles estaciones. No obstante, se ha realizado una prueba con el solver comercial de optimización *gurobi*, y se obtiene la misma solución con un tiempo de cómputo reducido de forma considerable (ver Tabla 6).

<i>Tabla 6: Tiempos de cómputo utilizando el solver comercial gurobi para el Modelo 1.</i>	
Base de datos utilizada	Tiempo de cómputo
Base de datos de 6 estaciones	0.2424 s
Base de datos de 9 estaciones	0.6425 s

Por el mínimo tiempo de cómputo y su similitud en la cobertura por pares obtenidas en las diferentes soluciones, podemos llegar a la conclusión que el Modelo 2 creado, puede llegar a ser una buena simplificación del Modelo 1. Teniendo siempre en cuenta que no representa fielmente la realidad, y sería un fallo no tener en cuenta posibles nodos que pueden generar demanda, el Modelo 2 puede llegar a servir de apoyo para la planificación y construcción de líneas de transporte de tránsito rápido.

Por último, se realiza un estudio con la base completa de posibles estaciones en la ciudad de Sevilla, para comprobar la efectividad y rapidez del modelo de programación del Modelo 2. Las soluciones obtenidas para el mismo presupuesto máximo que se realiza en las pruebas anteriores (ver Tabla 7, Ilustración 9).

Tabla 7: Resultado con una base de datos de 28 estaciones. Modelo 2. Para $c_{max} = 15\ 000\ M.$	
Tipo de modelo	Modelo 2
Nº de estaciones	19
Vías	(2, 4), (4, 5), (5, 6), (6, 11), (11, 10), (10, 12), (12, 19), (19, 17), (17, 18), (18, 16), (16, 24), (24, 15), (15, 14), (14, 13), (13, 8), (8, 7), (7, 3), (3, 28)
z	277990 habitantes
t(s)	72.0386

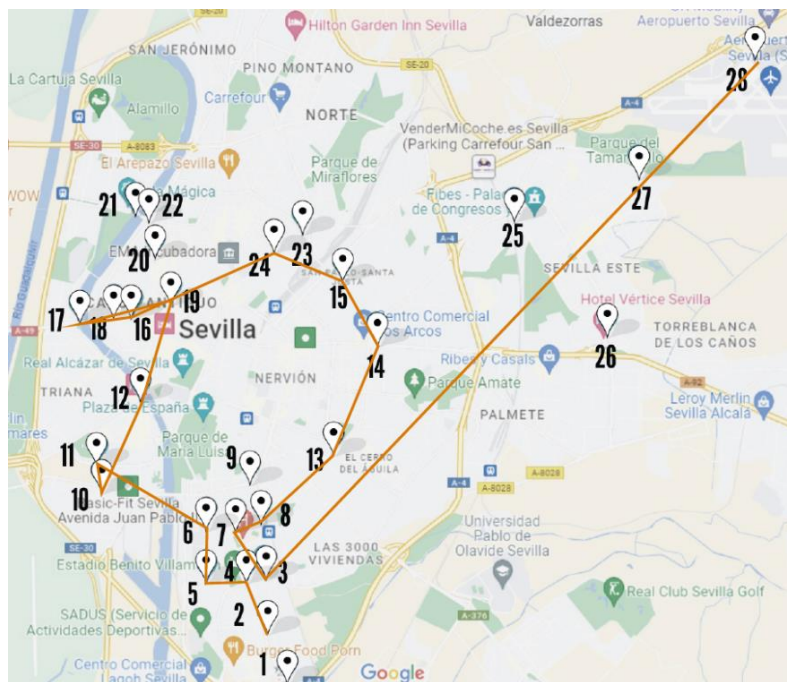


Ilustración 9: Recorrido de la línea de metro para una base de datos de 28 estaciones posibles, para $c_{max} = 15\ 000\ M.$

El modelo crea el 72.04% de las posibles estaciones, y esto se debe al alto coste máximo o presupuesto que se ha introducido en el modelo de una forma estimada. Para obtener un modelo más realista, reduciremos el

presupuesto inicial a máximo de forma decreciente con el fin de probar el modelo y obtener diferentes soluciones, y, además, encontraremos el de encontrar el presupuesto mínimo que se necesita para construir una línea de metro en Sevilla, considerando las áreas más pobladas (ver Tabla 8, Ilustración 10).

Tabla 8: Resultado con una base de datos de 28 estaciones. Modelo 2. Para $c_{max} = 14\ 500\ M.$	
$c_{max}(\text{€})$	14 500 M
Nº de estaciones	18
Vías	(2, 5), (5, 6), (6, 10), (10, 11), (11, 17), (17, 18), (18, 20), (20, 16), (16, 24), (24, 15), (15, 14), (14, 13), (13, 8), (8, 7), (8, 7), (7, 4), (4, 3), (3, 28)
z	271 311 habitantes
t(s)	752.55

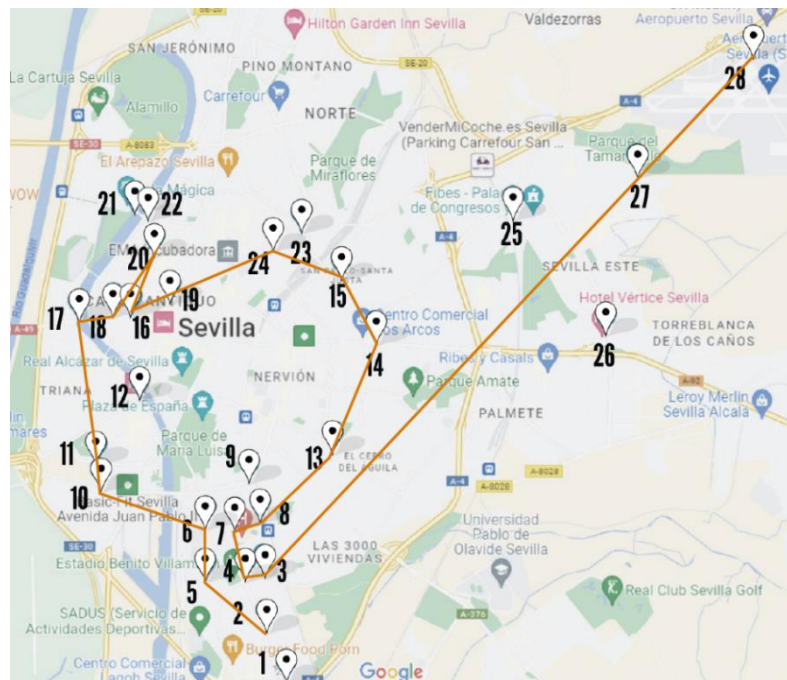


Ilustración 10: Recorrido de la línea de metro para una base de datos de 28 estaciones posibles, para $c_{max} = 14\ 500\ M\text{€}.$

Como podemos observar, que a pesar de que no haya diferencia en las estaciones o nodos construidos, si cambiamos el presupuesto máximo, el programa tarda demasiado tiempo en encontrar una solución factible. De nuevo, el Modelo de programación 2, se enfrenta al mayor conflicto estudiado por los científicos y expuesto en las revisiones del Capítulo 2 del presente trabajo. En este tipo de problemas matemáticos, el modelo no es capaz de producir una solución óptima en un tiempo de cómputo razonable utilizando el solver de código abierto para un escenario de tamaño realista. Es necesario acudir a un solver comercial, que sea capaz del manejo de una gran cantidad de variables, restricciones y una base de datos mayor. Por estas razones, a pesar de los esfuerzos realizados para encontrar un modelo que simplifique las restricciones y variables, se descarta la posibilidad de encontrar solución óptima para una base de datos que refleje la situación real en la ciudad de Sevilla.

5. CONCLUSIONES.

Este Trabajo de Fin de Grado se ha centrado en el estudio y desarrollo de modelos de programación matemática para el diseño de líneas de transporte de tránsito rápido. Se ha revisado la literatura existente sobre el problema de diseño de redes de tránsito rápido, y se ha puesto a prueba dos modelos generales que incluyen características de los modelos diferentes considerados en el ámbito de diseño de redes, con diferentes funciones objetivo y comparado las soluciones, siendo el segundo una contribución original de este trabajo.

Uno de los modelos desarrollados tiene como fin encontrar alguna solución al problema computacional que presentan los modelos complejos, simplificando de esta forma las variables y encontrando una solución óptima que sea válida para poder utilizarla como apoyo a la planificación de redes de

transporte de tránsito rápido. Es por esta razón que, a pesar de las simplificaciones que se hicieron, este tipo de análisis puede ayudar a comparar configuraciones básicas y se puede aplicar fácilmente a situaciones reales utilizando ajustes de parámetros adecuados.

Podemos afirmar, por lo tanto, que los modelos matemáticos aplicados al diseño de líneas de transporte de tránsito rápido proporcionan información valiosa sobre la eficacia de diferentes opciones de diseño, pero no puede reemplazar completamente el juicio y la experiencia de los expertos en transporte y urbanismo. Además, es importante tener en cuenta que la planificación de redes de transporte es un proceso iterativo que implica una serie de decisiones y compromisos que se deben tomar en función de las necesidades de la ciudad y las limitaciones de recursos.

6. BIBLIOGRAFÍA.

[1]. A. De Los Santos, G. Laporte, J.A. Mesa and F. Perea. (2012). Evaluating passenger robustness in a rail transit network. *Transportation Research Part C: Emerging Technologies*, 20(1), 34–46.

[2]. Ayuntamiento de Sevilla. (2023). Ayuntamiento de Sevilla.

[3]. Baaj, M. H., & Mahmassani, H. S. (1991). An AI-based approach for transit route system planning and design. *Journal of Advanced Transportation*, 25(2), 187-209-

[4]. Bruno, G., Gendreau, M., & Laporte, G. (2002b). A heuristic for the location of a rapid transit line. *Computers & Operations Research*, 29(1), 1-12.

[5]. Bruno, G., Ghiani, G., & Improta, G. (1998). A multi-modal approach to the location of a rapid transit line. *European Journal of Operational Research*, 104(2), 321-332.

[6]. Canca, D., De-Los-Santos, A., Laporte, G., & Mesa, J. A. (2019). Integrated Railway Rapid Transit Network Design and Line Planning problem with maximum profit. *Transportation Research Part E-logistics and Transportation Review*, 127, 1-30.

[7]. Ceder, A. and Wilson, N.H.M. (1986) *Bus Network Design*. *Transportation Research Part B Methodological*, 20, 331-344. – References – Scientific Research Publishing.

[8]. Chapleau, R., P. Lavigueur, and K. Baass. (1986). A Posteriori Impact Analysis of a Subway Extension in Montreal. *Transportation Research Record*, 1152

[9]. Claessens, M., Van Dijk, N., & Zwaneveld, P. (1998). Cost optimal allocation of rail passenger lines. *European Journal of Operational Research*, 110(3), 474-489.

[10]. Conoce la estrategia y sus ejes | Estrategia de movilidad Segura, sostenible y conectada 2030. (2023).

- [11]. Delegación de Hacienda y Administración Pública Excmo. Ayuntamiento de Sevilla. (2017). Indicadores demográficos de Sevilla.
- [12]. Dufourd, H., Gendreau, M., & Laporte, G. (1996). Locating a transit line using Tabu Search. *Location Science*, 4(1-2), 1-19.
- [13]. Feo, T. A., & Resende, M. G. C. (1989). A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8(2), 67-71.
- [14]. García-Archilla, B., Lozano, A. J., Mesa, J. A., & Perea, F. (2011). GRASP algorithms for the robust railway network Design Problem. *Journal of Heuristics*, 19(2), 399-422.
- [15]. Gattuso, D., & Miriello, E. (2005). Compared analysis of metro networks supported by graph theory. *Networks and Spatial Economics*, 5(4), 395-414.
- [16]. Guihaire, V., & Hao, J. (2008). Transit Network Design and Scheduling: A Global review. *Transportation Research Part A: Policy and Practice*, 42(10), 1251-1273.
- [17]. Gutiérrez-Jarpa, G., Laporte, G., Marianov, V., & Moccia, L. (2017). Multi-objective rapid transit network design with modal competition: the case of Concepción, Chile. *Computers & Operations Research*, 78, 27-43.

- [18]. Informe especial: Movilidad urbana. (2023).
- [19]. Karlaftis, M. G. (2004). A DEA approach for evaluating the efficiency and effectiveness of urban transit systems. *European Journal of Operational Research*, 152(2), 354-364.
- [20]. Laporte, G., Marín, Á. M., Mesa, J. A., & Ortega, F. A. (2007). An Integrated Methodology for the Rapid Transit Network Design Problem. En *Springer eBooks* (pp. 187-199).
- [21]. Laporte, G., Marín, Á., Mesa, J. A., & Perea, F. (2011). Designing robust rapid transit networks with alternative routes. *Journal of Advanced Transportation*, 45(1), 54-65.
- [22]. Laporte, G., Mesa, J. A., & Ortega, F. A. (1997). Assessing the efficiency of rapid transit configurations. *Top*, 5(1), 95-104.
- [23]. Laporte, G., Mesa, J. A., & Perea, F. (2010). A game theoretic framework for the robust railway transit network design problem. *Transportation Research Part B-methodological*, 44(4), 447-459.
- [24]. Laporte, G., Mesa, J. A., Ortega, F. A., & Perea, F. (2011). Planning rapid transit networks. *Socio-economic Planning Sciences*, 45(3), 95-104.

[25]. Laporte, G., Mesa, J. A., Ortega, F. B., & Sevillano, I. (2005). Maximizing Trip Coverage in the Location of a Single Rapid Transit Alignment. *Annals of Operations Research*, 136(1), 49-63.

[26]. Lee, K., Jung, W., Park, J. S., & Choi, M. Y. (2008). Statistical analysis of the Metropolitan Seoul Subway System: network structure and passenger flows. *Physica D: Nonlinear Phenomena*, 387(24), 6231-6234.

[27]. Manzanares Abásolo, M. A. (2022). Localización de una línea de metro: efectos de la sobreocupación [TRABAJO FIN DE MÁSTER]. Universidad de Sevilla.

[28]. Marín, Á. M., & García-Ródenas, R. (2009). Location of infrastructure in urban railway networks. *Computers & Operations Research*, 36(5), 1461-1477.

[29]. Marín, A. y Jaramillo, P (2008). Urban rapid transit network design: accelerated Benders decomposition, *Annals of Operations Research* 35-375.

[30]. Marín, Á., Mesa, J. A., & Perea, F. (2009). Integrating robust railway network design and line planning under failures. En *Springer eBooks* (pp. 273-292).

[31]. Mesa, J. (2015). *The design of rapid transit networks*. Us.

[32]. Musso, A., & Vuchic, V. R. (1988). CHARACTERISTICS OF METRO NETWORKS AND METHODOLOGY FOR THEIR EVALUATION. Transportation Research Record, 1162.

[33]. Ramos, F. L. (2014). Conjoint design of railway lines and frequency setting under semi-congested scenarios. En TDX (Tesis Doctorals en Xarxa).

[34] Saidi S, Wirashinghe S, Kattan L (2016) Long-term planning for ring-radial urban rail transit networks. Transp Res B Methodol 86:128–146

[35] Schöbel, A. (2011). Line planning in public transportation: models and methods. OR Spectrum, 34(3), 491-510.

[36]. Scholl, S.: Customer-Oriented Line Planning. Ph.D. Thesis, University of Kaiserslautern (2005).

[37]. Sun, D., & Guan, S. (2016). Measuring vulnerability of urban metro network from line operation perspective. Transportation Research Part A-policy and Practice, 94, 348-359.

[38]. T.A. Feo and M.G.C. Resende. (1995). Greedy randomized adaptive search procedures. Journal of Global Optimization, 109– 133.

[39]. The International Association of Public Transport | UITP. 2023. UITP.

[40]. Transporte Seguro y sostenible | Unión Europea. 2023. European Union.

[41]. UrbanMobilityDays. (2023). Mobility and Transport.

7. ANEXOS.

7.1. Anexo 1: Modelo 1 de Programación Matemática para el Diseño de Líneas de Tránsito rápido.

```
import math
import timeit
start = timeit.default_timer()

with open ('datos_9.txt') as datos:
    new_line = [int(x) for x in next(datos).split()]
    n = new_line[0]
    m = new_line[1]
    next(datos).split()
    matriz = []
    contador_l = 1
    while contador_l <= n:
        new_line = [float(x) for x in next(datos).split()]
        matriz.append(new_line)
        contador_l += 1
```

```

    # print("matriz", matriz)
next(datos).split()
contador_l = 1
while contador_l <= m:
    new_line = [int(x) for x in next(datos).split()]
    matriz.append(new_line)
    contador_l += 1

#print(matriz)

import pyomo.environ as pyo
Metro = pyo.ConcreteModel()
Metro.Nodes = pyo.RangeSet(1,n)
N = Metro.Nodes

def Pairs_rule(Metro):
    return set([(i,j) for i in N for j in N if i != j])
Metro.Pairs = pyo.Set(initialize=Pairs_rule, dimen=2)
W = Metro.Pairs

h = {(i): matriz[i-1][0] for i in N}
#print('El número de habitantes por cada estación es:', h)

# # defino costes de arcos, tengo que añadir un diccionario para calcular la raiz cuadrada: costes
= raiz de (coord de uno menos la de j) al cuadrado más (coord de i menos j) al cuadrado
coordenada_hor = {(i): matriz[i-1][1] for i in N}
coordenada_ver = {(i): matriz[i-1][25] for i in N}
#print('las coordenadas son', coordenada_hor, coordenada_ver)

ejes = set()
for l in range(n,n+m):
    ejes.add((matriz[l][0],matriz[l][1]))
#print("ejes", ejes)

#Crear el conjunto de aristas como aquellos pares de nodo con distancia menor que dist_promedio
# def Edges_rule(Metro):
#     return set([(i,j) for i in N for j in N if i < j and dist[i,j] < dist_promedio])
# #Las aristas (edges en inglés) no están dirigidas
# Metro.Edges = pyo.Set(initialize = Edges_rule, dimen = 2)

Metro.Edges = ejes
E= Metro.Edges

```

```

# matriz_coordenadas = np.array(list(coordenadas.values()))

# # from scipy.spatial.distance import pdist, squareform
# # multiplicamos las distancias por 1000 para que el programa pueda procesarlas bien
dist = {(i,j):round(1000*math.sqrt((coordenada_hor[i] - coordenada_hor[j])**2 + (coordenada_ver[i]
- coordenada_ver[j])**2)) for i in N for j in N if i!=j}
print('la distancia entre cada estación es', dist)

#Los arcos están dirigidos
arcos = set()
for e in ejes:
    arcos.add(e)
    arcos.add((e[1],e[0]))
#print("arcos",arcos)
Metro.Arcs = arcos
A= Metro.Arcs
# print("Número de arcos:", len(A))

# print("E",E)
# print("A",A)

#método alternativo, multiplico las distancias por 2
u= {(i,j): 2*round(dist[i,j]) for i in N for j in N if (i,j) in W}
print("la matriz u es",u)

#defino la demanda W
g= {(i,j): 1000/dist[i,j] for i in N for j in N if (i,j) in W}
print("g",g)
# print((15,21) in W)
# print(g[15,21])
#Coste de estaciones, en este ejemplo las ponemos constantes
b = {(i): 20 for i in N}

#Coste de cada km de vía
c = 50

#Presupuesto máximo
P = 15000

```



```

# definimos parámetros de la restricción 3.12.
# tiempo de parada del metro
t_k = 4
# tiempo de espera:
t_wait = 6

v = 50 # km/h, velocidad promedio

t_a = {(i,j): round(dist[i,j]) for (i,j) in A}
print("t_a",t_a)

# Creo el conjunto de posibles nodos origen de la línea, serán las estaciones 1,2,3 o 4
def I_rule (Metro):
    return set([i for i in N if i <= 2])
Metro.I = pyo.Set(initialize=I_rule, dimen=1)
I = Metro.I
# I.pprint()

# Creo el conjunto de posibles nodos fin de la línea, será el aeropuerto
def J_rule (Metro):
    return set([i for i in N if i == n])
Metro.J = pyo.Set(initialize=J_rule, dimen=1)
J = Metro.J
# J.pprint()

# Creo el conjunto de estaciones intermedias
def L_rule (Metro):
# return (set([(i) for i in Nodes if (i>4) and (i < 28)]))
    return (set([(i) for i in N if (i not in I) and (i not in J)]))
Metro.L = pyo.Set(initialize=L_rule, dimen=1)
L = Metro.L
# L.pprint()

print("DEFINICIÓN DE VARIABLES")
#defino variables
Metro.x = pyo.Var(E, within = pyo.Binary, initialize = 0)
x = Metro.x
Metro.y = pyo.Var(N, within = pyo.Binary, initialize = 0)
y = Metro.y

```

a variable f se define con el índice A, pero su segundo índice es W,A, lo que significa que la variable es un diccionario que tiene como llave una tupla de dos elementos.

```
Metro.f = pyo.Var(W,A, within= pyo.Binary, initialize = 0)
```

```
f = Metro.f
```

```
#variable de captura
```

```
Metro.z = pyo.Var(W, within = pyo.Binary, initialize = 0)
```

```
z = Metro.z
```

```
# #fijar que la solución sea la obtenida con el modelo 2 para 6 estaciones
```

```
# solucion_mod2 = {(2,3), (3,4), (4,5), (5,6)}
```

```
# for e in E:
```

```
#   if e in solucion_mod2:
```

```
#       x[e].fix(1)
```

```
#   else:
```

```
#       x[e].fix(0)
```

```
# # fijar que la solución sea la obtenida con el modelo 2 para 9 estaciones
```

```
# solucion_mod2 = {(4, 3), (3, 5), (5, 6), (6, 8), (8, 7), (7, 9)}
```

```
# for e in E:
```

```
#   if e in solucion_mod2:
```

```
#       x[e].fix(1)
```

```
#   else:
```

```
#       x[e].fix(0)
```

```
print("DEFINICIÓN OBJETIVO")
```

```
#función objetivo
```

```
Metro.obj = pyo.Objective(expr = sum(g[w]*z[w] for w in W), sense = pyo.maximize)
```

```
print("DEFINICIÓN RESTRICCIÓN Presupuesto")
```

```
#Restricción 3.2 de presupuesto, sigue igual
```

```
coste_red = sum(b[i]*y[i] for i in N) + sum(c*dist[e]*x[e] for e in E)
```

```
Metro.Presupuesto = pyo.Constraint(expr = coste_red <= P)
```

```
#restricción 3.4
```

sigue igual: para que si se construye el arco (i,j), entonces la estación j debe existir, y no la estación i. Si construyo una vía he de construir las estaciones adyacentes

```
print("DEFINICIÓN RESTRICCIÓN Estaciones")
```

```
Metro.Estaciones = pyo.ConstraintList()
```

```
for (i,j) in E:
```

```
    Metro.Estaciones.add(x[i,j] <= y[i])
```

```
    Metro.Estaciones.add(x[i,j] <= y[j])
```

```

# restricción 3.5
print("DEFINICIÓN RESTRICCIÓN Estacionesi")
# #tiene que haber una estación en el grupo de estaciones iniciales
Metro.Estacionesi = pyo.Constraint(expr = sum(y[i] for i in I)==1)

print("DEFINICIÓN RESTRICCIÓN Estacionesj")
# # tiene que haber una estacion en el grupo de estaciones finales, realmente como es una no es
necesaria, pero si cambiamos el modelo ahí está
Metro.Estacionesj = pyo.Constraint(expr = sum(y[j] for j in J)==1)

# restricción 3.6
print("DEFINICIÓN RESTRICCIÓN ViasEstaciones")
# El número de estaciones es igual al número de vías + 1
Metro.ViasEstaciones = pyo.Constraint(expr = sum(y[i] for i in N) == sum(x[e] for (e) in E) + 1)
#Metro.ViasEstaciones.pprint()

# restricción 3.7 - condición de que la línea sea un camino
print("DEFINICIÓN RESTRICCIÓN Camino")
Metro.Camino = pyo.ConstraintList()
for k in L:
    Metro.Camino.add(sum(x[e] for e in E if k in e) <= 2*y[k])

# restricción 3.8 - de todas las estaciones origen sol se construye una, idem estaciones finaleS
print("DEFINICIÓN RESTRICCIÓN Estación Inicial")
Metro.Estacion_I = pyo.Constraint(expr = sum(x[e] for o in I for e in E if o in e) == 1)
#
# sum(x[i,o] for o in I for i in N if (i,o) in E) == 1)
# print("DEFINICIÓN RESTRICCIÓN Estación Final")
Metro.Estacion_J = pyo.Constraint(expr = sum(x[e] for d in J for e in E if d in e) == 1)
# Metro.Estacion_I.pprint()
# Metro.Estacion_J.pprint()

# flujo 3.9
print("DEFINICIÓN RESTRICCIÓN FlujoF")
Metro.FlujoF = pyo.ConstraintList()
for w in W:
    for k in N:
        if k == w[0]:
            Metro.FlujoF.add(sum(f[w,k,j] for j in N if (k,j) in A) - sum(f[w,i,k] for i in N if (i,k) in A) ==
z[w])

        elif k == w[1]:
            Metro.FlujoF.add(sum(f[w,k,j] for j in N if (k,j) in A) - sum(f[w,i,k] for i in N if (i,k) in A) == -
z[w])

```

else:

```
Metro.FlujoF.add(sum(f[w,k,j] for j in N if (k,j) in A) - sum(f[w,i,k] for i in N if (i,k) in A) == 0)
```

```
# restriccion 3.10, si hay flujo la variable fwa, vale 1
```

```
print("DEFINICIÓN RESTRICCIÓN Capacidad")
```

```
Metro.Capacidad = pyo.ConstraintList()
```

```
for w in W:
```

```
    for e in E:
```

```
        Metro.Capacidad.add(f[w, e] + f[w, e[1], e[0]] <= x[e])
```

```
# restriccion 3.11
```

```
print("DEFINICIÓN RESTRICCIÓN Localización1")
```

```
Metro.Loc1 = pyo.ConstraintList()
```

```
for (ws,wt) in W:
```

```
    Metro.Loc1.add(sum(f[ws,wt,ws,j] for j in N if (ws,j) in A) <= y[ws])
```

```
Metro.Loc2 = pyo.ConstraintList()
```

```
for (ws,wt) in W:
```

```
    Metro.Loc2.add(sum(f[ws,wt,i,wt] for i in N if (i,wt) in A) <= y[wt])
```

```
# restriccion 3.12
```

```
print("DEFINICIÓN RESTRICCIÓN ConstruccionInicioFinal")
```

```
Metro.ConstruccionInicioFinal = pyo.ConstraintList()
```

```
for (ws,wt) in W:
```

```
    Metro.ConstruccionInicioFinal.add(z[ws,wt] <= y[ws])
```

```
    Metro.ConstruccionInicioFinal.add(z[ws,wt] <= y[wt])
```

```
# elección del modo; restriccion 3.13
```

```
print("DEFINICIÓN RESTRICCIÓN Eleccion")
```

```
Metro.Eleccion = pyo.ConstraintList()
```

```
for w in W:
```

```
    Metro.Eleccion.add(sum(t_a[a]*f[w,a] for a in A)
```

```
        + sum(t_k*f[w,k,j] for k in L for j in N if (k,j) in A)
```

```
        + t_wait*z[w] <= u[w])
```

```
# #Ahora resuelvo el modelo con el solve glpk
```

```
print("RESOLUCIÓN DEL MODELO")
```

```
opt = pyo.SolverFactory('glpk')
```

```

#opt = pyo.SolverFactory('gurobi')
opt.solve(Metro, tee = True)
stop=timeit.default_timer()
#muestro el resultado por pantalla
#Metro.display()

print('***** SOLUTION *****')
print('Objetivo óptimo:',pyo.value(Metro.obj))
pares_captados = sum(pyo.value(z[w]) for w in W)
stop = timeit.default_timer()
print('Time: ', stop - start)
print('Número de pares captados', pares_captados, 100*pares_captados/len(W),'% del total')
print('Coste red', pyo.value(coste_red))

print('Se construyen las estaciones:')
for i in N:
    if pyo.value(y[i])==1:
        print(i)
print('Se construyen las vías:')
for e in E:
    if pyo.value(x[e])==1:
        print(e)
print('Se captan los pares:')
for w in W:
    if pyo.value(z[w])==1:
        print('par',w)
        tiempo_red = sum(t_a[a]*pyo.value(f[w,a]) for a in A) + sum(t_k*pyo.value(f[w,k,j]) for k in
L for j in N if (k,j) in A)+ t_wait*pyo.value(z[w])
        print("\t Captado. Tiempo en red", tiempo_red, "u",u[w],"usa los arcos")
        for a in A:
            if pyo.value(f[w,a])==1:
                print("\t",a)
        # else:
        #   print("\t no captado")

```

7.2. Anexo 2: Modelo 2 de Programación Matemática para el Diseño de Líneas de Tránsito Rápido.

```
import math
import timeit
start = timeit.default_timer()

#Leo los datos desde un fichero .txt
with open('datos_9.txt') as datos:
    new_line = [int(x) for x in next(datos).split()]
    n = new_line[0]
    m = new_line[1]
    next(datos).split()
    matriz = []
    contador_l = 1
    while contador_l <= n:
        new_line = [float(x) for x in next(datos).split()]
        matriz.append(new_line)
        contador_l += 1
        # print("matriz", matriz)
    next(datos).split()
    contador_l = 1
    while contador_l <= m:
        new_line = [int(x) for x in next(datos).split()]
        matriz.append(new_line)
        contador_l += 1

#print(matriz)

import pyomo.environ as pyo
Metro = pyo.ConcreteModel()
Metro.Nodes = pyo.RangeSet(1,n)
N = Metro.Nodes

def Pairs_rule(Metro):
    return set([(i,j) for i in N for j in N if i != j])
Metro.Pairs = pyo.Set(initialize=Pairs_rule, dimen=2)
W = Metro.Pairs
```

```

h = {(i): matriz[i-1][0] for i in N}
#print('El número de habitantes por cada estación es:', h)

coordenada_hor = {(i): matriz[i-1][1] for i in N}
coordenada_ver = {(i): matriz[i-1][25] for i in N}
#print('las coordenadas son', coordenada_hor, coordenada_ver)

ejes = set()
for l in range(n,n+m):
    ejes.add((matriz[l][0],matriz[l][1]))
#print("ejes", ejes)
Metro.Edges = ejes
E= Metro.Edges

dist = {(i,j):round(1000*math.sqrt((coordenada_hor[i] - coordenada_hor[j])**2 + (coordenada_ver[i]
- coordenada_ver[j])**2)) for i in N for j in N if i!=j}
arcos = set()
for e in ejes:
    arcos.add(e)
    arcos.add((e[1],e[0]))
#print("arcos", arcos)
Metro.Arcs = arcos
A= Metro.Arcs

# defino costes de arcos
dist = {(i,j):round(1000*math.sqrt((coordenada_hor[i] - coordenada_hor[j])**2 + (coordenada_ver[i]
- coordenada_ver[j])**2)) for i in N for j in N if i!=j}
print('la distancia entre cada estación es', dist)
# D = {(i,j): matriz2[i-1][j-1]
#       for i in N for j in N}
# print('Las distancias entre vías son', D)

h = {(i): matriz[i-1][0] for i in N}
print('El número de habitantes por cada estación es:', h)

#Coste de estaciones, en este ejemplo las ponemos constantes
c_i = {(i): 20 for i in N}

#Coste de cada km de vía
c_ij = 50

```

```

#Presupuesto máximo
C_max = 15000

# Creo el conjunto de posibles nodos origen de la línea, serán las estaciones 1,2,3 o 4
def I_rule (Metro):
    return set([(i) for i in N if i <= 2])
Metro.I = pyo.Set(initialize=I_rule, dimen=1)
I = Metro.I
I.pprint()

# Creo el conjunto de posibles nodos fin de la línea, será el aeropuerto
def J_rule (Metro):
    return set([(i) for i in N if i == n])
Metro.J = pyo.Set(initialize=J_rule, dimen=1)
J = Metro.J

# Creo el conjunto de estaciones intermedias
def L_rule (Metro):
    return (set([(i) for i in N if (i not in I) and (i not in J)]))
Metro.L = pyo.Set(initialize=L_rule, dimen=1)
L = Metro.L
L.pprint()

print("DEFINICIÓN DE VARIABLES")
#defino variables
Metro.x = pyo.Var(A, within = pyo.Binary, initialize = 0)
x = Metro.x
Metro.y = pyo.Var(N, within = pyo.Binary, initialize = 0)
y = Metro.y
Metro.u = pyo.Var(N)
u = Metro.u

# #fijar que la solución sea la obtenida con el modelo 1 para 6 estaciones
# solucion_mod1 = {(1,3), (3,4), (4,6)}
# for e in E:
#     if e in solucion_mod1:
#         x[e].fix(1)
#     else:
#         x[e].fix(0)

# #fijar que la solución sea la obtenida con el modelo 1 para 6 estaciones

```



```

# solucion_mod1 = {(1, 3), (3, 4), (4, 6), (6, 5), (5, 7), (7, 8), (8, 9)}
# for e in E:
#   if e in solucion_mod1:
#       x[e].fix(1)
#   else:
#       x[e].fix(0)

print("DEFINICIÓN OBJETIVO")
#función objetivo
Metro.obj = pyo.Objective(expr = sum(h[i]*y[i] for i in N), sense = pyo.maximize)

#Restricción de presupuesto
Metro.Presupuesto = pyo.Constraint(expr = sum(c_ij[i]*y[i] for i in N) +
                                         sum(c_ij*dist[i,j]*x[i,j] for (i,j) in A) <= C_max)
# Metro.Presupuesto.pprint()

print("DEFINICIÓN RESTRICCIÓN Estaciones")
# Si construyo una vía he de construir las estaciones adyacentes
Metro.E_origen = pyo.ConstraintList()
for (i,j) in A:
    Metro.E_origen.add(x[i,j] <= y[i])
Metro.E_destino = pyo.ConstraintList()
for (i,j) in A:
    Metro.E_origen.add(x[i,j] <= y[j])

print("DEFINICIÓN RESTRICCIÓN ViasEstaciones")
# El número de estaciones es igual al número de vías + 1
Metro.ViasEstaciones = pyo.Constraint(expr = sum(y[i] for i in N) == sum(x[i,j] for (i,j) in A) + 1)
#Metro.ViasEstaciones.pprint()

# defino las vías que salen de la estación de origen
Metro.V_origen = pyo.ConstraintList()
for i in I:
    Metro.V_origen.add(sum(x[i,j] for j in N if (i,j) in A) == y[i])

print("DEFINICIÓN RESTRICCIÓN Estacionesi")
#tiene que haber una estación en el grupo de estaciones iniciales
Metro.Estacionesi = pyo.Constraint(expr = sum(y[i] for i in I)==1)

# defino las vias que salen de la estacion final
Metro.V_destino = pyo.ConstraintList()
for j in J:

```

```

Metro.V_destino.add(sum(x[i,j] for i in N if (i,j) in A) == y[j])

print("DEFINICIÓN RESTRICCIÓN Estacionesj")
# tiene que haber una estacion en el grupo de estaciones finales, realmente como es una no es
necesaria, pero si cambiamos el modelo ahí está
Metro.Estacionesj = pyo.Constraint(expr = sum(y[j] for j in J)==1)

print("DEFINICIÓN RESTRICCIÓN V_intermedias")
Metro.V_intermedias_i = pyo.ConstraintList()
Metro.V_intermedias_j = pyo.ConstraintList()
for l in L:
    Metro.V_intermedias_i.add(sum(x[i,l] for i in N if (i,l) in A) == y[l])
    Metro.V_intermedias_j.add(sum(x[l,j] for j in N if (l,j) in A) == y[l])

# Restricciones de ruptura de ciclos MTZ. Esta vez la defino como una regla, otra opción que nos
da Python/Pyomo, se puede definir tambien como una lista
def MTZ_rule(Metro,i,j):
    if i in l:
        return pyo.Constraint.Skip
    elif j in J:
        return pyo.Constraint.Skip
    else:
        return u[i] - u[j] + n*x[i,j] <= n - 1
Metro.MTZ = pyo.Constraint(A, rule=MTZ_rule)

print("RESOLUCIÓN DEL MODELO")

#Ahora resuelvo el modelo con el solve glpk
# opt = pyo.SolverFactory('glpk')
# opt.solve(Metro)
# stop=timeit.default_timer()

opt = pyo.SolverFactory('glpk')
opt.solve(Metro, tee = True)
stop=timeit.default_timer()

#muestro el resultado por pantalla
Metro.display()
#Como la salida anterior puede ser difícil de interpretar por tener muchas variables, creo una rutina
para sacar resultados por pantalla de forma más "amigable"
print('***** SOLUTION *****')
```

```

print('Objetivo óptimo:',pyo.value(Metro.obj))
print('Se construyen las estaciones:')
for i in N:
    if pyo.value(y[i])==1:
        print(i)
print('Se construyen las vías:')
for a in A:
    if pyo.value(x[a])==1:
        print(a)

stop = timeit.default_timer()
print('Time: ', stop - start)
print('Numero estaciones', sum(pyo.value(y[i]) for i in N))
print('Numero vias', sum(pyo.value(x[i,j]) for (i,j) in A))

```

7.3. Anexo 3: Bases de datos.

- Base de datos para 6 estaciones:

```

6 9
#data nodes: population, horizontal coordinate, vertical coordinate
7500      37.343136 -5.973038
7500      37.349947 -5.976327
3095      37.357223 -5.977435
1750      37.356453 -5.979593
38215     37.356217 -5.986310
34736     37.363559 -5.986364

# edge set
1 2
1 3
2 3
2 4
3 4
3 5
4 5

```

4 6

5 6

- Base de datos para 9 estaciones:

9 16

#data nodes: population, horizontal coordinate, vertical coordinate

7500 37.343136 -5.973038

7500 37.349947 -5.976327

3095 37.357223 -5.977435

1750 37.356453 -5.979593

38215 37.356217 -5.986310

34736 37.363559 -5.986364

36081 37.363448 -5.981496

9888 37.364133 -5.977262

0 37.425425 -5.895632

edge set

1 2

1 3

1 9

2 3

2 4

3 4

3 5

4 5

4 6

5 6

5 7

6 7

6 8

7 8

7 9

- Base de datos para 28 estaciones:

28 116

#data nodes: population, horizontal coordinate, vertical coordinate

7500	37.343136	-5.973038
7500	37.349947	-5.976327
3095	37.357223	-5.977435
1750	37.356453	-5.979593
38215	37.356217	-5.986310
34736	37.363559	-5.986364
36081	37.363448	-5.981496
9888	37.364133	-5.977262
6079	37.368838	-5.979254
8346	37.368569	-6.003830
8346	37.371822	-6.004254
8346	37.380078	-5.997317
9325	37.373199	-5.965312
20281	37.387902	-5.958306
35207	37.395931	-5.964301
1179	37.391012	-5.998558
13457	37.390577	-6.007557
11790	37.391561	-6.001653
11790	37.392930	-5.992164
13457	37.398902	-5.994772
1667	37.404799	-5.998462
1667	37.403810	-5.995804
18658	37.402412	-5.970498
18658	37.400502	-5.975131
17368	37.404505	-5.935539
17368	37.409824	-5.914817

29284 37.388704 -5.920020

0 37.425425 -5.895632

edge set

1 3

1 28

1 4

1 5

1 6

1 7

1 8

1 9

1 10

1 11

2 3

2 4

2 5

2 6

2 7

2 8

2 9

2 10

2 11

2 28

3 4

3 5

3 6

3 7

3 8

3 9

3 10

3 11

3 12

3 28

4 5

4 6

4 7

4 8

4 9

4 10

4 11

4 12

4 13

4 28

5 6

5 7

5 8

5 9

5 10

5 11

5 12

5 13

6 7

6 8

6 9

6 10

6 11

6 12

6 13

6 14

7 8

7 9

7 10

7 11

7 12

7 13

7 14

8 9

8 10

8 11

8 12

8 13

8 14

9 10

9 11

9 12

9 13

9 14

10 11

10 12

10 16

10 18

10 19

11 16

11 17

11 18

11 19

12 14

12 16

12 18

12 19

12 20

12 21

13 14

13 15

14 15

14 19

14 24
14 25
15 24
15 25
15 19
16 17
16 18
16 19
16 20
16 21
16 22
16 23
16 24
17 18
17 19
17 20
17 21
17 22
17 23
17 24
18 19
18 20
18 21
18 22
18 23
18 24
18 15
19 20
19 21
19 22
19 23
19 24

19 15
19 25
19 28
20 21
20 22
20 23
20 24
20 25
20 15
21 22
21 23
21 24
21 25
21 15
22 23
22 24
22 25
22 26
22 15
23 25
23 15
23 26
15 27
23 27
23 28
24 23
24 25
24 26
24 27
24 28
25 26
25 28

26 28

27 26

27 28