Universidad Autónoma Metropolitana

Unidad Iztapalapa

División de Ciencias Básicas e Ingeniería
Departamento de Matemáticas

Optimización convexa y métodos variacionales para estimar matrices origen destino

T E S I S

que para obtener el grado de
Doctora en Ciencias Matemáticas

Presenta:
María Victoria Chávez Hernández

Asesor: Dr. Lorenzo Héctor Juárez Valencia

Sinodales: Dr. David Guillermo Romero Vargas
Dr. Joaquín Delgado Fernández
Dr. Marcos Aurelio Capistrán Ocampo
Dr. Miguel Ángel Gutiérrez Andrade

Ciudad de México a 3 de julio de 2019

Dedicatoria

A mis padres,
Alicia Hernández Cortés
y Joel Chávez Martínez.

A mis hermanas y hermanos,
(en orden de aparición)
Miguel, Alicia, Lupita, Joel, Sarita y Fede.

Al amor de mi vida,
Sergio.

A mis gathijos,
Gaussiana, Hessiana y Heisenberg.

Agradecimientos

Al pensar en esta etapa de mi vida, recuerdo aquel famosos juego de los 90's donde el plomero de bigotes simpáticos que comía hongos y atrapaba estrellas se ponía trajes ad hoc para cada ocasión y superaba cada inconveniente que el mundo le ponía. En este momento siento que he terminado el juego y, aunque sé que la princesa está en otro castillo, veo cada una de las etapas por las que pasé como pequeños mundos y quiero agradecer a las personas que me ayudaron a confeccionar el traje y me proporcionaron las herramientas para poder hacer uso de algunos superpoderes.

Quiero agradecer profundamente a mi padre académico Héctor Juárez, quien sin conocer a esa chica que un día se apareció en su cubículo para decirle que quería trabajar con él, tuvo la confianza de aceptarme como su alumna de maestría y posteriormente de doctorado. Por sus observaciones, comentarios y sugerencias para llevar a cabo este trabajo. Por su iniciativa de enviarme a congresos, talleres de alto nivel y estancias de investigación que ayudaron a complementar mi formación y a vincularme con investigadores reconocidos internacionalmente. Por sus recomendaciones y su confianza para involucrarme con el taller que impartí en el metro a profesionales y expertos en transporte. Por su calidad humana y ayudarme a ver las cosas en perspectiva cuando más lo necesitaba. Definitivamente no puede tener mejor asesor.

A mis sinodales por sus valiosas observaciones, su paciencia y disposición para concluir esta etapa.

A Yasmín Ríos por jugar conmigo a los camioncitos y enseñarme el lado divertido de la construcción de modelos lineales e inspirarme a seguir adelante.

A Michael Florian y Yolanda Noriega que siempre han estado pendientes de este trabajo, nos han facilitado las licencias del software y han aportado su perspectiva y experiencia en los problemas de transporte.

A Roberto Cominetti, por su disposición para recibirme y contarme las anécdotas que surgen al intentar ofrecer un proyecto. Además, por su interés en orientarme para la siguiente etapa.

A María Luisa Sandoval por darme la oportunidad de involucrarme en la organización de un evento que fue muy significativo para mí y por sus consejos.

A mis profesores Raúl Montes de Oca, Roberto Quezada y Ernesto Barrios por sus provechosas

enseñanzas.

Al ingeniero Pablo Torres por sus observaciones complementarias y por facilitarnos la base de datos que manejan en el metro.

A Carolina Moreno por haberme dado la oportunidad de entrar a la industria privada y por su enorme vocación de cambiar el mundo.

A mis amigas y amigos que me acompañaron en cada etapa escuchándome, prestándome sus hombros, dándome mi espacio e inyectando sonrisas y alegría siempre: Male, Jessica, Pablo, Isma, Lorelei, Victor, Tere, Estela, Jorge, Daniel, Héctor, Laura, Alfredo, Erwin, Giovanni, Juan Luis, Fernando, Omar, Ivonne, Miguel, Karla y Melina.

A Sergio, por estar conmigo siempre e impulsarme a seguir creciendo. Por llevarme de la mano cuando se me olvidó cómo camiar y por inculcarme hábitos sanos.

A mis suegros y mi cuñada, quienes se han convertido en parte de mi familia sin papeles ni rituales de por medio, simplemente por el cariño que les tengo y que se han ganado a pulso.

A mis alumnos que me han ayudado a llenar esa necesidad que tengo de transmitir mi gusto por las matemáticas y a la vez me han marcado con sus invaluables ejemplos de vida: Benjamín, Eunice, Isabel, Aura y Alejandro.

Por último, quiero agradecer al CONACYT por otorgarme una beca, sin la cual no me hubiese sido posible salir de mi ciudad natal para llevar a cabo mis estudios de posgrado ni tener los intercambios académicos y culturales que he experimentado.

Es así que, cuando algo se complica, ustedes me ayudan a activar mi *superP* y soy capaz de volar hasta el castillo más lejano para derrotar al malvado rey de los koopas.

"Si he visto más lejos es porque voy a hombros de Gigantes"
Isaac Newton,1675.

Índice general

Índice de figuras

Resumen

En este trabajo se incluye una revisión de los modelos y algoritmos más utilizados para estimar matrices de demanda o matrices origen destino (O-D) en redes de transporte público a partir de una pequeña cantidad de datos observados. Además, se proponen nuevos modelos y algoritmos para resolver este problema, los cuales mejoran los resultados y son más eficientes que los comúnmente utilizados hasta el momento por los ingenieros e investigadores del transporte.

Considerando que no hay cambios bruscos de la demanda en el área geográfica de estudio, el objetivo de los modelos que aquí estudiamos es encontrar una nueva matriz O-D que sea lo más cercana posible a otra matriz O-D de referencia, la cual pudo haberse obtenido a partir de encuestas en los hogares o algún otro método. Esta nueva matriz O-D debe satisfacer dos condiciones más

- sus entradas deben ser no negativas y
- al llevar a cabo una asignación de tránsito con ella, se deben reproducir los datos observados.

En la literatura (ver capítulo 1) se pueden encontrar modelos que de manera general consisten en minimizar la suma de un par de funciones que miden la distancia entre los valores de referencia y los estimados para la demanda y el flujo de pasajeros en algunos segmentos de la red. Una de las metodologías más estudiadas en la literatura es el método de (Spiess, 1990), en el cual se busca minimizar la distancia entre los volúmenes de tránsito (tráfico) observados y los que se obtienen después del método de asignación. Para resolver el problema, Spiess propuso un método de máximo descenso multiplicativo que preserva la estructura de una matriz conocida *a priori* y la no negatividad de las soluciones.

En este trabajo se proponen dos nuevos modelos para resolver el problema, con base en un enfoque de control óptimo para resolver el problema inverso: un modelo de penalización (sección 3.2) y un modelo de Lagrangiano aumentado (sección 3.3). Estos modelos son inéditos dentro del ámbito de la investigación en transporte, y se construyeron tomando en cuenta la teoría de problemas inversos y de control en modelos descritos por ecuaciones diferenciales parciales. Ambos son modelos de optimización cuadrática, en donde se busca la matriz más cercana a una matriz obsoleta y que ajusta mediciones de flujo de pasajeros en un porcentaje pequeño de arcos de la red.

Se utilizan algoritmos iterativos para resolver los problemas, los cuales demuestran ser más eficientes que los comúnmente utilizados en la investigación del transporte: un método de gradiente conjugado multiplicativo (para el modelo de penalización en la sección 3.2.2) y un

método de ascenso dual y multiplicadores (para el modelo de Lagrangiano aumentado en la sección 3.3.1).

Además, se demuestra que las soluciones del modelo penalizado convergen a la solución del modelo de Spiess, cuando el parámetro de penalización tiende a infinito (sección 3.2.1). Los resultados numéricos corroboran esta propiedad (sección 4.2). Cabe aclarar que en la literatura, el modelo de Spiess se considera uno de los más simples y eficientes, de hecho es utilizado ampliamente en el software comercial canadiense EMME, el cual es uno de los más exitosos y populares en el ambiente de transporte (capítulo 1, ecuaciones (1.2) y (1.3)).

El modelo de penalización no solo generaliza el modelo de Spiess sino también es equivalente a los modelos de promedios pesados que tienen su base en la optimización cuadrática, pero con la ventaja de que los resultados teóricos permiten orientar el valor adecuado de los pesos, con base en su equivalencia con el parámetro de penalización (3.9).

Al tener términos de penalización en ambos modelos y sus correspondiente algoritmos, es posible encontrar soluciones de manera estable ante perturbaciones o errores en los datos. Es decir son modelos de regularización que permiten resolver los problemas subdeterminados, asociados al problema, sin amplificar el ruido o error en los datos o mediciones (ver el capítulo 4).

Los dos modelos y sus correspondientes algoritmos son varias veces más rápidos que el modelo de Spiess y su algoritmo de descenso máximo multiplicativo; en particular, los resultados son más precisos cuando se utiliza el Lagrangiano aumentado para forzar la no negatividad de los coeficientes en la matriz O-D (tablas 4.2-4.8).

Las propiedades anteriores permiten aplicar estas herramientas a la estimación de demanda en redes de transporte de gran tamaño. El tiempo de ejecución es del orden de segundos en una computadora personal de tamaño normal (Laptop); como se demuestra con los resultados para la red de la Zona Metropolitana del Valle de México, la cual incluye la Ciudad de México, los municipios conurbados del estado de México y un municipio del estado de Hidalgo (tabla 4.7). La utilización de supercómputo permitirá mejorar aun más los tiempos de ejecución.

En la tabla 5.2, se puede ver que el modelo de Lagrangiano aumentado es el más completo, ya que aparte de ser más preciso y eficiente en redes de gran tamaño; puede ser generalizado para incluir otros aspectos a medir, como los volúmenes de transporte en segmentos de la red, los costos de viaje las producciones y atracciones en la demanda dentro de la red de transporte (ecuaciones (3.31) y (3.32), tabla 4.8).

Los métodos aquí propuestos se probaron con dos redes: la red de tránsito de la ciudad de Winnipeg, que cuenta con 23716 pares O-D; y la red de tránsito de la Zona Metropolitana del Valle de México con más de 2 millones de pares O-D. En estos dos casos, consideramos una reducción del tamaño del problema (sección 3.4) extrayendo los coeficientes nulos en la matriz de referencia para reducir aun más el tiempo de cómputo.

Capítulo 1

Introducción

El flujo de pasajeros entre cada par de zonas en una red de tránsito es un arreglo bidimensional conocido como matriz de demanda origen-destino (matriz O-D) y es uno de los elementos más importantes, pero a la vez más desafiante y costoso de obtener para llevar a cabo cualquier proceso de planificación en una red de transporte. Si bien la mayor parte de la investigación se ha orientado a redes de carreteras, aquí nos enfocamos en redes de transporte público, como los autobuses, el metro, el metrobús, etc.

Las matrices O-D generalmente se obtienen a partir de encuestas cada cierto tiempo (10 años o más), dependiendo de la dinámica de la población (Bera y Rao, 2011). Sin embargo; llevar a cabo cualquiera de estos estudios y procesar la información requiere de un gran esfuerzo, además de una gran inversión monetaria y también de tiempo, de tal manera que cuando se libera la construcción de la matriz O-D; frecuentemente ésta ya no está estrictamente vigente y se requiere de un proceso de actualización o afinación de la misma para reflejar mejor el flujo de viajeros en la red, lo cual es de fundamental importancia en cualquier modelo de tránsito (Juárez et al., 2013). Por ejemplo, en la Zona Metropolitana del Valle de México (ZMVM) se llevaron a cabo encuestas para estimar la matriz O-D en los años 2007 y 2017; sin embargo la información fue liberada casi un año después INEGI (2007, 2018), respectivamente; tiempo suficiente para que haya cambios en la demanda.

Debido a que en el proceso de planificación del transporte se debe actualizar frecuentemente la matriz O-D, es necesario tomar la información disponible y complementarla con información adicional de la red, que se pueda obtener de manera relativamente fácil y sin mucho costo, para obtener una mejor matriz O-D. A lo largo de los años, los expertos del transporte han propuesto agregar información adicional a la ya disponible para actualizar la demanda, como por ejemplo: los flujos observados en algunos lugares estratégicos de la red, información por zonas, datos de teléfonos móviles o datos de tarjetas de tarifas inteligentes (Alsger et al., 2016; Bierlaire, 1995; Doblas y Benitez, 2005; Heidari et al., 2017; Kumar et al., 2016; Nuzzolo y Comi, 2016).

Una vez que se actualiza la matriz O-D, es posible aplicar un método de asignación de demanda adecuado para obtener patrones de flujo (número de personas) que, al ser examinados, servirán para identificar problemas y se podrán tomar las medidas necesarias para mejorar el transporte en la red de estudio. Por supuesto, con una buena estimación de la matriz O-D, es

más probable asimilar los datos observados, mejorar los modelos de tránsito y los pronósticos correspondientes.

Desde los años 70's se han desarrollado varios métodos para estimar matrices O-D, los cuales se pueden dividir en estáticos y dinámicos. En el caso estático; se estudia un cierto período del día, como la hora pico de la mañana, los volúmenes de tránsito se consideran independientes del tiempo y se utilizan para estimar matrices a largo plazo con la finalidad de diseñar los cambios necesarios en la infraestructura de la red. En el problema dinámico (Antoniou et al., 2016; Ashok y Ben-Akiva, 2002; Cascetta et al., 2013; Cipriani et al., 2011; Frederix et al., 2013; Shafiei et al., 2016; Verbas et al., 2011; Hu et al., 2017), se consideran varios períodos del día y debe modelarse la tasa de cambio en el flujo a lo largo del periodo de estudio; se utilizan para estimar la demanda a corto plazo con la finalidad de controlar el flujo de vehículos o pasajeros sobre la red, o para sugerir rutas a los usuarios. En este trabajo; nos concentramos en los enfoques estáticos, ya que estos modelos son la base de los enfoques dinámicos (Etemadnia y Abdelghany, 2009).

Entre las formulaciones y métodos estudiados en la literatura, varios enfoques buscan minimizar la distancia de Mahalanobis entre los datos observados y los valores predichos (Bell, 1991; Cascetta y Nguyen, 1988). Otros; como Noriega y Florian (2009), Verbas et al. (2011) y Shafiei et al. (2016); consideran un promedio ponderado del cuadrado de las distancias Euclidianas entre los volúmenes observados y los estimados y el cuadrado de la diferencia entre una matriz de referencia (usualmente obtenida a partir de encuestas) y la nueva estimada.

También se ha estudiado el problema de estimar la matriz O-D desde enfoques estadísticos como: máxima verosimilitud, mínimos cuadrados generalizados y estimadores de Bayes. En Cascetta (1984) y Cascetta y Nguyen (1988) se afirma que el estimador de mínimos cuadrados con un procedimiento de asignación lineal es el mejor estimador sin sesgo, si tanto la matriz de referencia como los volúmenes en los segmentos observados son exactos. La solidez del método de mínimos cuadrados generalizados se ha explotado para estimar matrices O-D estáticas o dinámicas, ya que permite la combinación de datos topográficos y datos de conteo de flujo, permitiendo incorporar la precisión relativa de las dos fuentes de datos (Bell, 1991; Cascetta et al., 2013; Fujita et al., 2016; Malapert y Kuusinen, 2017).

En este trabajo de tesis, el problema de actualizar una matriz O-D a partir de conteos de flujo en algunos segmentos de tránsito, se formula como un problema de optimización convexa restringido. Denotemos por $\mathcal{P}$ al conjunto de nodos origen y por $\mathcal{Q}$ al conjunto de nodos destino; así, $pq \in \mathcal{PQ}$ denota un par O-D con $p \in \mathcal{P}$ y $q \in \mathcal{Q}$. Además, representemos por $\mathcal{A}$ al conjunto de todos los segmentos dirigidos en la red de tránsito y $\hat{\mathcal{A}} \subset \mathcal{A}$ al subconjunto de segmentos donde se tienen flujos observados. La cantidad de segmentos en $\mathcal{A}$ suele ser mucho mayor que la cantidad de segmentos en $\hat{\mathcal{A}}$; además, pensando que cada nodo origen (destino) puede se un destino (origen), los conjuntos $\mathcal{P}$ y $\mathcal{Q}$ son iguales.

La función objetivo corresponde a una función de distancia entre una matriz *a priori* (de referencia) $\hat{\mathbf{g}} = \{\hat{g}_{pq}\}_{pq \in \mathcal{PQ}}$ y la demanda resultante $\mathbf{g} = \{g_{pq}\}_{pq \in \mathcal{PQ}}$. Las restricciones hacen que los volúmenes obtenidos mediante una asignación $\mathbf{v} = \{v_a\}_{a \in \hat{A}}$ correspondan a los volúmenes

observados $\hat{\mathbf{v}} = \{\hat{v}_a\}_{a\in\hat{A}}$ en los segmentos donde se realizaron los conteos $a \in \hat{\mathcal{A}}$. En particular, el modelo penalizado (introducido en Juárez y Chávez, 2014; Chávez y Juárez, 2014, 2016) pertenece al problema de programación general

$$\min_{\mathbf{g}} F_1(\mathbf{g}; \hat{\mathbf{g}}) + F_2(\mathbf{v}; \hat{\mathbf{v}}), \tag{1.1}$$

donde F_1 y F_2 son funciones de distancia y $\mathbf{v} = Assign(\mathbf{g})$ se entiende como una asignación de tránsito de equilibrio; por ejemplo, el modelo lineal con base en estrategias óptimas introducido por Spiess y Florian (1989), el cual se usa en este trabajo.

Después de una asignación de tránsito lineal, obtenemos las proporciones de ruta π^a_{pq}; cada una representa la probabilidad de que una persona use el segmento a para ir de p a q. Estas probabilidades se pueden organizar en una matriz $P = \{\pi_{a,i}\}$ en donde, para simplificar la notación, el índice i representa el par O-D pq; esto significa que P tiene dimensiones $m \times n$, con m el número de segmentos de tránsito y n el número de pares de O-D. De este modo; una asignación de tránsito se puede representar como el producto de una matriz P y un vector de demanda $\mathbf{g} = \{g_i\}$, obtenido al ordenar adecuadamente la matriz O-D, para obtener el flujo de tránsito $\mathbf{v} = P\mathbf{g}$ sobre los segmentos en la red.

El problema (1.1) se puede ver como un problema inverso (Michau et al., 2017); donde el problema directo consiste en calcular los flujos del segmento cuando se conoce la matriz O-D, es decir $\mathbf{v} = Assign(\mathbf{g})$, y el problema inverso consiste en estimar la matriz O-D cuando se conocen los flujos de personas en algunos segmentos de tránsito, es decir dados los volúmenes $\hat{\mathbf{v}}$ y la matriz *a priori* $\hat{\mathbf{g}}$ resolver (1.1). En este trabajo nos enfocamos en el problema inverso para redes de gran tamaño.

En el pasado, los modelos del tipo (1.1) tenían poca relevancia práctica debido a "*la gran cantidad de tiempo de cómputo y requerimientos de almacenamiento que se presenta en su implementación, lo cual limita estos enfoques solamente a redes pequeñas*", según Spiess (1990). Entonces, tratando de superar esta deficiencia, Spiess introdujo el siguiente modelo:

$$\min_{\mathbf{g}} Z(\mathbf{g}) = \frac{1}{2}\sum_{a\in\hat{A}} (v_a - \hat{v}_a)^2 = \frac{1}{2}\|P\mathbf{g} - \hat{\mathbf{v}}\|^2, \tag{1.2a}$$

$$\text{sujeto a: } g_i \geq 0 \text{ para todo } i = pq \in \mathcal{PQ}, \tag{1.2b}$$

el cual es altamente indeterminado y tiene un número infinito de soluciones, esto significa que hay infinitas matrices O-D que recuperan igualmente bien los volúmenes observados $\hat{\mathbf{v}}$. Para superar esta degeneración, Spiess sugirió elegir la matriz estimada más cercana a la matriz *a priori*, que mantiene la misma estructura. Para lograr este objetivo, él introdujo un método multiplicativo de descenso máximo (MDM):

$$g_i^{\ell+1} = \begin{cases} \hat{g}_i, & \text{para} \quad \ell = 0, \\ g_i^{\ell}\left(1 - \gamma_\ell \frac{\partial Z(\mathbf{g}^\ell)}{\partial g_i}\right), & \text{para} \quad \ell = 1, 2, ... \end{cases} \quad \text{sujeto a: } \gamma_\ell \frac{\partial Z(\mathbf{g}^\ell)}{\partial g_i} \leq 1, \tag{1.3}$$

para cada $i = pq \in \mathcal{PQ}$; donde ℓ es el contador de iteraciones y γ_ℓ es la longitud del tamaño de paso en la iteración ℓ. Este algoritmo multiplicativo mantiene la estructura de la matriz a

priori $\hat{\mathbf{g}}$, y su simplicidad hace que sea aplicable a redes de gran escala.

En Chávez (2014); Chávez y Juárez (2014), se extiende la idea de Spiess por medio de la introducción de un algoritmo de gradiente conjugado multiplicativo, el cual mejora la eficiencia del cómputo (menos iteraciones con la misma precisión) para obtener (actualizar) la matriz O-D. Los métodos multiplicativos están inspirados para mantener la estructura de la matriz O-D *a priori*; sin embargo, esta condición se puede relajar y de esta manera permitir formular otros modelos con algoritmos de solución adecuados (ver Codina y Barceló, 2000; Bierlaire y Toint, 1995; Codina y Barceló, 2000; Codina et al., 2006; Cipriani et al., 2011; Florian y Chen, 1995; Lundgren y Peterson, 2008; Shafiei et al., 2016; Shen y Winter, 2012; Xie et al., 2011).

Otros autores han formulado el problema como uno de programación lineal (entera o no entera) con la intención o esperanza de aplicar algoritmos que permitan resolver problemas para redes de gran escala (Hu et al., 2017; Michau et al., 2017; Pitombeira-Neto et al., 2016; Sherali et al., 1994). De hecho, en Hu et al. (2017) se afirma que "*debido a la estructura lineal, su modelo es más efectivo computacionalmente y resoluble en redes reales grandes a diferencia de la formulación de mínimos cuadrados comúnmente utilizada, la cual es computacionalmente difícil e ineficiente para redes grandes*". Sin embargo, estos autores aplican su metodología solo a una red pequeña, con 80 zonas de tráfico, 830 arcos y 395 nodos.

Los modelos cuadráticos, con algoritmos de solución adecuados, han demostrado ser eficientes y muy competitivos para problemas grandes en muchas aplicaciones diferentes. Sin embargo, a pesar del aumento del poder de cómputo a lo largo de los años y el éxito de los modelos cuadráticos en las redes de tránsito (Verbas et al., 2011), todavía hay mucho trabajo por hacer para desarrollar algoritmos efectivos que ayuden a estimar o actualizar matrices de demanda de O-D para redes de tránsito y tráfico a gran escala con este tipo de modelos convexos.

En esta tesis abordaremos el problema de actualizar demanda en transporte con modelos de mínimos cuadrados de la forma (1.1) y algunas de sus variantes; además, aplicaremos algoritmos de optimización convexa para obtener las soluciones. El objetivo es extender los modelos cuadráticos, comúnmente utilizados en este tipo de problemas, e introducir algoritmos que permitan mejorar los resultados para reducir el costo computacional de las soluciones numéricas.

Los modelos y algoritmos introducidos en esta tesis han sido desarrollados y publicados a largo de la investigación y, hasta donde sabemos, son originales en el contexto de la estimación de matrices O-D. Mostramos la efectividad de los mismos con tres redes: una ficticia, la red de tránsito de Winnipeg y la red de tránsito con base en la Zona Metropolitana del Valle de México (ZMVM), que está constituida por las 16 delegaciones de la Ciudad de México, 59 municipios conurbados del estado de México y uno del estado de Hidalgo.

Primero, reformularemos el problema de estimación de demanda como un problema inverso con un modelo de mínimos cuadrados con restricciones; posteriormente, incluiremos las restricciones, asociadas a la medición de los volúmenes, en la función objetivo mediante un modelo de penalización (introducido en Chávez (2014) y publicado en Chávez y Juárez (2014); Juárez

y Chávez (2014); Chávez y Juárez (2016)) y demostraremos matemáticamente que cuando el parámetro de penalización tiende a su valor límite, el conjunto de soluciones de los modelos penalizados convergen a la solución del modelo de Spiess, lo cual además es validado por los resultados numéricos obtenidos.

El modelo penalizado no solo generaliza el modelo de Spiess sino también a otros modelos de promedios ponderados comúnmente utilizados en la literatura (Noriega y Florian, 2009). Después utilizaremos un algoritmo de gradiente conjugado multiplicativo para resolver este modelo con el objeto de obtener matrices actualizadas con la misma estructura que la matriz de referencia (o dada *a priori*), pero con menos costo computacional que el método de descenso máximo de Spiess. Posteriormente utilizaremos un enfoque de Lagrangiano aumentado combinado con el método de solución de ascenso dual y multiplicadores (Nocedal y Wright, 2006; Boyd et al., 2010) para forzar la no negatividad de los coeficientes de la matriz O-D (Chávez et al., 2019). Este enfoque nos permite evitar algoritmos iterativos con estructura multiplicativa y obtener mejores resultados con esencialmente el mismo costo de cómputo. Además, emplearemos una reducción directa del problema (reducción de orden del modelo computacional) para mejorar aun más el tiempo de cómputo de las soluciones numéricas.

Finalmente, queremos mencionar que algunos artículos relacionados donde se emplea el enfoque de Lagrangiano de una manera o contexto diferente, en una red de transporte, incluyen por ejemplo los de Balakrishnan et al. (1989); Bierlaire y Toint (1995); Doblas y Benitez (2005).

El resto de esta tesis está organizado de la siguiente manera. En el capítulo 2, describimos de manera breve el método de asignación que se empleó en este trabajo. En el capítulo 3, formulamos el problema general para estimar matrices O-D, introducimos el modelo penalizado, mostramos su convergencia al problema de Spiess cuando el parámetro de penalización tiende a infinito y describimos el algoritmo de GCM para resolver el problema. En particular, en la sección 3.3, presentamos un modelo de Lagrangiano aumentado, discutimos sus propiedades así como la relación entre los dos parámetros de penalización involucrados y describimos el algoritmo de solución, ascenso dual y la método de multiplicadores (ADMM); en la sección 3.4, consideramos una reducción del problema para facilitar la implementación de los algoritmos en redes grandes; además, incorporamos al modelo información acerca de la demanda en las zonas agregadas de tránsito. Los resultados correspondientes para los dos casos de estudio se muestran en el capítulo 4. Finalmente, en el capítulo 5 se incluyen algunas conclusiones.

Capítulo 2

El problema de asignación

En 1989 Spiess y Florian introdujeron un modelo de asignación de tránsito con base en el concepto de *estrategia óptima*; donde una estrategia es un conjunto de reglas que cuando se aplican, le permiten al viajero llegar a su destino. En este contexto una estrategia óptima se refiere a la estrategia que minimiza el tiempo esperado de viaje de todo el sistema.

En este capítulo; en la sección 2.1, se introduce la notación y el modelo de asignación lineal con base en estrategias óptimas de Spiess y Florian (1989); en la sección 2.2, se formula el problema dual y el algoritmo de solución. Finalmente, en la sección 2.3 se muestra un ejemplo de una red ficticia en el que se aplica el algoritmo de asignación.

2.1. Formulación del problema

Para describir matemáticamente el modelo de asignación lineal, introduzcamos la siguiente notación:

N	Conjunto de nodos de la red de tránsito.
$\mathcal{P} \subset N$	Conjunto de nodos origen.
$\mathcal{Q} \subset N$	Conjunto de nodos destino.
$\mathcal{A}$	Conjunto de segmentos dirigidos en la red de tránsito.
$(i,j) \in \mathcal{A}$	Segmento dirigido de tránsito que va del nodo $i \in N$ al nodo $j \in N$.
π_{pq}^{ij}	Probabilidad de que un pasajero que va de $p \in \mathcal{P}$ a $q \in \mathcal{Q}$ utilice el segmento $(i,j) \in \mathcal{A}$.
v_{ij}	Cantidad total (también llamado volumen) de pasajeros que utilizan el segmento $(i,j) \in \mathcal{A}$.
t_{ij}	Tiempo de viaje sobre el segmento $(i,j) \in \mathcal{A}$.
v_i	Cantidad de pasajeros esperando en el nodo $i \in N$.
χ_{ij}	Indicadora para determina si el segmento $(i,j) \in \mathcal{A}$ se encuentra o no dentro del conjunto de estrategias para ir de cada origen a cada destino.
f_{ij}	Frecuencia de servicio en el segmento $(i,j) \in \mathcal{A}$.
$\mathbf{g}$	Matriz de demanda de pasajeros cuyas entradas $\{g_{pq}\}$ representan la cantidad de pasajeros que van de $p \in \mathcal{P}$ a $q \in \mathcal{Q}$.

Así, el problema de asignación consiste en la búsqueda de los volúmenes de flujo que resuelven

el siguiente problema lineal de optimización convexa para cada destino $q \in \mathcal{Q}$ (Spiess y Florian, 1989):

$$\text{mín} \sum_{(i,j)\in\mathcal{A}} t_{ij}v_{ij} + \sum_{i\in N} w_i \tag{2.1a}$$

$$s.\ a. \sum_{j\in N} v_{kj} - \sum_{i\in N} v_{ik} = g_{kq}, \qquad \forall\ k \in N, \tag{2.1b}$$

$$v_{ij} \le f_{ij}w_i, \qquad \forall (i,j) \in \mathcal{A}, \quad i \in N, \tag{2.1c}$$

$$v_{ij} \ge 0, \qquad \forall (i,j) \in \mathcal{A}. \tag{2.1d}$$

con $w_i = v_i / \sum_{i\in N} \chi_{i,j} f_{ij}$.

Este modelo propone minimizar el tiempo de viaje más el tiempo de espera de todos los pasajeros en la red de tránsito, sujeto a que en cada nodo $k \in N$, la cantidad de pasajeros que se van menos la cantidad de pasajeros que llegan es igual a la demanda de viajes que se originan en el nodo k. Además; la cantidad de pasajeros que utilizan el segmento $(i,j) \in \mathcal{A}$ está acotada por la cantidad de pasajeros que esperan en el nodo $i \in N$.

2.2. Formulación dual del problema y algoritmo de solución

En la práctica, el problema (2.1) no se resuelve directamente, pues es computacionalmente mejor resolver su formulación dual, permitiendo su aplicación a redes de gran tamaño. El problema dual se obtiene después de formular el Lagrangiano asociado al problema de minimización. Introduciendo los multiplicadores $\mathbf{u} = \{u_k\}$ asociados a las restricciones de balanceo de flujo (2.1b) y $\mu = \{\mu_{ij} \ge 0\}$ asociados a la restricción (2.1c), el Lagrangiano correspondiente queda de la siguiente manera:

$$\mathcal{L}(\mathbf{v},\mathbf{w},\mathbf{u},\mu) = \sum_{(i,j)\in\mathcal{A}} t_{ij}v_{ij} + \sum_{i\in N} w_i + \sum_{k\in N} u_k \left(g_{kq} - \sum_{j\in N} v_{kj} + \sum_{i\in N} v_{ik} \right) + \sum_{(i,j)\in\mathcal{A}} \mu_{ij} \left(v_{ij} - f_{ij}w_i \right), \qquad \text{con} \quad v_{ij},\ \mu_{ij}\ \ge 0, \quad \forall (i,j) \in \mathcal{A}.$$

Separando las sumas dobles y reacomodando términos se obtiene

$$\mathcal{L}(\mathbf{v},\mathbf{w},\mathbf{u},\mu) = \sum_{i\in N}\sum_{j\in N} (t_{ij}v_{ij} + \mu_{ij}v_{ij}) + \sum_{i\in N} \left(w_i - \sum_{j\in N} \mu_{ij}f_{ij}w_i \right) + \sum_{k\in N} u_k g_{kq} - \sum_{k\in N} u_k \sum_{j\in N} v_{kj} + \sum_{k\in N} u_k \sum_{i\in N} v_{ik}.$$

Factorizando w_i y reindexando las sumas de los últimos dos términos, se obtiene

$$\mathcal{L}(\mathbf{v},\mathbf{w},\mathbf{u},\mu) = \sum_{i\in N}\sum_{j\in N}(t_{ij}v_{ij}+\mu_{ij}v_{ij}) + \sum_{i\in N} w_i\left(1-\sum_{j\in N}\mu_{ij}f_{ij}\right) + \sum_{k\in N}u_k g_{kq} - \sum_{i\in N}u_i\sum_{j\in N}v_{ij} + \sum_{j\in N}u_j\sum_{i\in N}v_{ij}.$$

Factorizando v_{ij}, finalmente queda

$$\mathcal{L}(\mathbf{v},\mathbf{w},\mathbf{u},\mu) = \sum_{i\in N}\sum_{j\in N} v_{ij}\left(t_{ij}+\mu_{ij}-u_i+u_j\right) + \sum_{i\in N} w_i\left(1-\sum_{j\in N}\mu_{ij}f_{ij}\right) + \sum_{k\in N}u_k g_{kq}.$$

Recordando que el Lagrangiano se maximiza respecto a las variables duales y considerando ahora a $v_{ij}\geq 0$ y w_i como multiplicadores, se puede ver que el problema dual asociado a (2.2) es

$$\begin{aligned}
\text{máx} \quad & \sum_{k\in N} g_{kq}u_k && \text{(2.2a)}\\
\text{sujeto a} \quad & u_i - u_j - \mu_{ij} \leq t_{ij}, \quad (i,j)\in\mathcal{A} && \text{(2.2b)}\\
& \sum_{j\in N} f_{ij}\mu_{ij} = 1, \quad i\in N && \text{(2.2c)}\\
& \mu_{ij}\geq 0, \quad (i,j)\in\mathcal{A}. && \text{(2.2d)}
\end{aligned}$$

Así, el problema dual consiste en maximizar tiempo total de viaje esperado u_k. El algoritmo para resolver este problema consiste en dos etapas. En la primera etapa, desde cada nodo destino q se determinan los segmentos que componen a la estrategia óptima y los tiempos totales esperados de viaje u_k desde cada uno de sus correspondientes nodos origen $k\in N$. En la segunda etapa, desde todos los nodos origen hasta el nodo destino, se distribuye la demanda sobre la red de tránsito, de acuerdo con la estrategia óptima calculada en la primera etapa. El algoritmo de solución es el siguiente:

Algoritmo 1 Asignación lineal de tránsito.

Entrada: Nodo origen $p \in \mathcal{P}$, nodo destino $q \in \mathcal{Q}$, demanda g_{pq}, $\mathcal{A}$, t_{ij} y f_{ij} para todo $(i,j) \in \mathcal{A}$.

Salida: Los volúmenes en los segmentos de la red de tránsito $v_{ij},\ \forall (i,j) \in \mathcal{A}$.

Primera etapa: *Encontrar la estrategia óptima.*

1: Para cada destino $q \in \mathcal{Q}$ hacer: ▷ **Inicialización**

$$u_i = \infty,\ \forall i \in N - \{q\}, \quad u_q = 0, \qquad f_i = 0,\ \forall i \in N, \qquad S = \mathcal{A}, \quad \bar{S} = \emptyset.$$

2: **Si** $S = \emptyset$, ▷ **Seleccionar el siguiente segmento**

3: parar;

4: **si no**,

5: encontrar el segmento $(i,j) \in S$ que satisface:

$$u_j + t_{ij} \leq u_{j'} + t_{i'j'}, \qquad (i',j') \in S,$$

6: $S = S - (i,j)$.

7: **Si** $u_i \geq u_j + t_{ij}$, entonces: ▷ **Actualizar la etiqueta del nodo**

$$u_i = \frac{f_i u_i + f_{ij}(u_j + t_{ij})}{f_i + f_{ij}}, \qquad f_i = f_i + f_{ij}, \qquad \bar{S} = \bar{S} + (i,j),$$

8: **regresar** al paso 2.

Segunda etapa: *Asignar la demanda de acuerdo con la estrategia óptima.*

9: $v_p = g_{pq}$, $v_q = -g_{pq}$.

10: Para cada segmento $(i,j) \in A$, hacer en orden decreciente de $u_j + t_{ij}$:

11: **Si** $(i,j) \in \bar{S}$, entonces:

$$v_{ij} = \frac{f_{ij} v_i}{f_i}, \qquad \pi_{pq}^{ij} = \frac{v_{ij}}{g_{pq}}, \qquad v_j = v_j + v_{ij},$$

12: **si no**,

$$v_{ij} = 0,$$

Observemos que en la inicialización del algoritmo 1, el tiempo esperado u_i para llegar al destino desde el nodo i se hace igual a infinito para todos los nodos, excepto para el nodo destino $q \in \mathcal{Q}$, el cual se toma igual a cero. Las frecuencias combinadas f_i de todos los segmentos seleccionados en el nodo $i \in N$, se hacen iguales a cero. El conjunto de segmentos S se utiliza para identificar los segmentos que no han sido examinados y el conjunto $\bar{S}$ se usa para identificar la estrategia óptima.

En el paso 2, se selecciona el segmento *más cercano* al nodo destino. El tiempo se calcula como el tiempo u_i desde el nodo $i \in N$ del segmento al destino $(u_j + t_{ij})$; si este tiempo es menor que el tiempo actual asociado con el nodo i, se incluye el segmento (i,j) en la estrategia óptima y se actualizan las etiquetas. Observemos que la primera vez que se actualizan las etiquetas, se encuentran casos de la forma $f_i u_i = \infty \cdot 0$, donde se adaptará la convención de que $\infty \cdot 0 = 1$. La primera parte del algoritmo termina cuando se examinan todos los segmentos.

La segunda parte del algoritmo consiste en calcular el volumen de pasajeros en cada segmento de la red y con estos, calcular la probabilidad de que cada segmento (i, j) sea usado para ir del nodo origen p al nodo destino q (π_{pq}^{ij}) de acuerdo con los segmentos de la estrategia óptima. Estas probabilidades, posteriormente serán la clave para representar de manera vectorial el problema de asignación y una vez obtenidas, basta con multiplicarlas por la demanda para obtener el volumen de pasajeros en cada segmento de la red de tránsito.

Observemos también, que el modelo lineal descrito no toma en cuenta los efectos de la congestión ni la capacidad limitada de los vehículos de transporte. Sin embargo; este modelo es muy útil cuando se utiliza como un elemento básico en la construcción de la solución de modelos más generales como el modelo congestionado, Juárez et al. (2013).

2.3. Ejemplo de aplicación en una red pequeña

Consideremos el siguiente ejemplo de una red ficticia de tránsito. Esta red, cuenta con 5 centroides, 6 nodos regulares y cuatro líneas de tránsito cuya velocidad es de 30 km/h. De aquí en adelante nos referiremos a esta red como "red ejemplo".

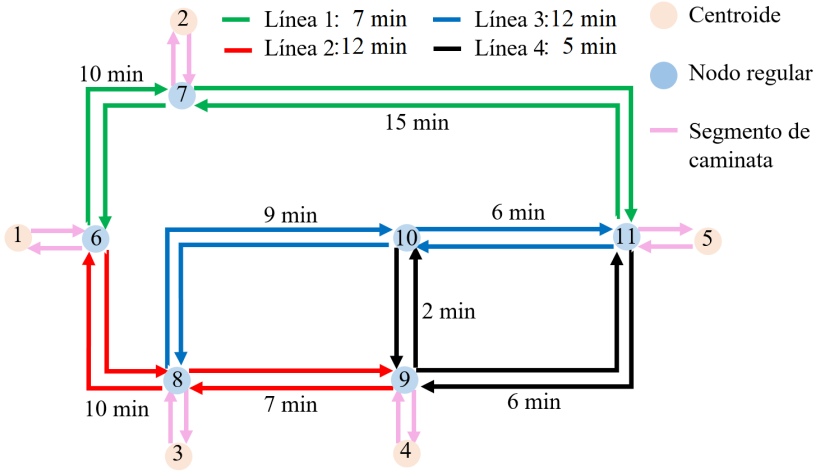


Figura 2.1: Red ejemplo con 5 centroides y 4 líneas.

En la figura 2.1 se muestra el tiempo de viaje t sobre cada uno de los segmentos de tránsito de cada una de las líneas, tomando en cuenta que el tiempo de viaje sobre un segmento en una dirección es igual al tiempo de viaje sobre el segmento que va en dirección contraria para cada una de las líneas de tránsito. El tiempo de cabecera de la línea 1 es de 7 minutos, el de las líneas 2 y 3 es de 12 minutos y el de la línea 4 es de 5 minutos.

Para resolver este problema de asignación, es necesario representar la red de tránsito mediante nodos y segmentos que nos permitan distinguir las bajadas, las subidas y los transbordos entre líneas, para lo cual se construye una red generalizada de tránsito (Fernández, 2013):

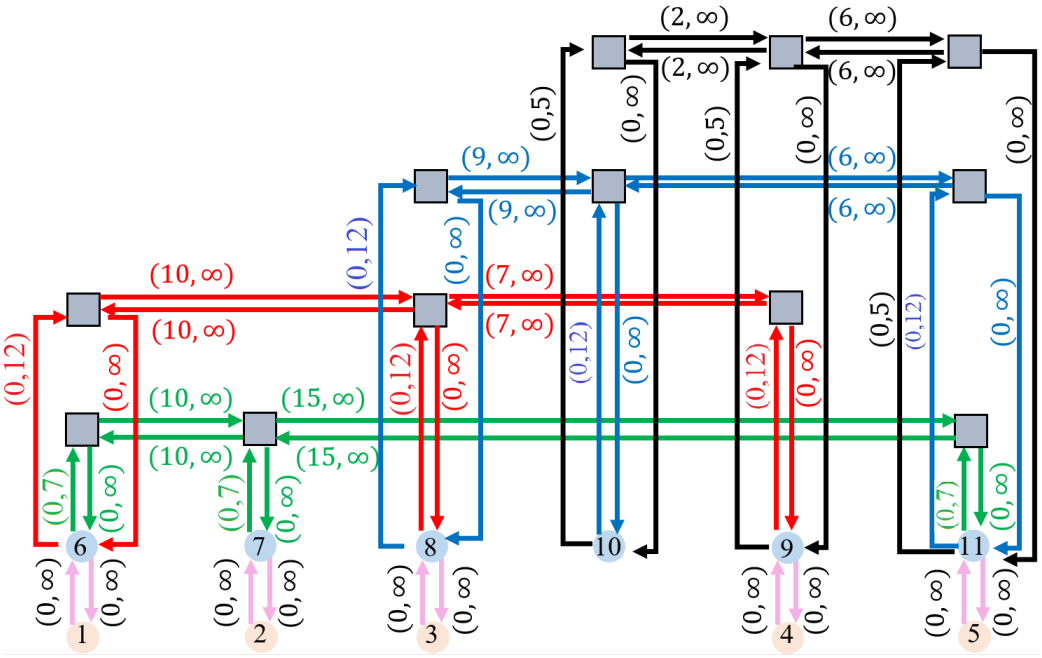


Figura 2.2: Red ejemplo generalizada de tránsito.

En la red generalizada 2.2 se tiene un primer nivel de nodos (centroides) que se conectan mediante segmentos de caminada a alguno de los nodos regulares que se encuentran en un segundo nivel. Los segmentos de caminata para este ejemplo, tienen tiempo de viaje $t_{ij} = 0$ y en general se les asigna una frecuencia de servicio $f_{ij} = \infty$. Además, en cada línea de tránsito se introduce un nodo por cada parada. Los segmentos que conectan a los nodos regulares y las respectivas líneas de tránsito tienen una frecuencia de viaje $f_{ij} = \infty$, si el segmento representa un viaje a bordo del vehículo de tránsito o si es un segmento de bajada. En cambio; si el segmento es de abordaje, la frecuencia es la correspondiente a la frecuencia de servicio de la línea que se va a abordar. Además, los tiempos de viaje sobre los segmentos de bajada o subida son $t_{ij} = 0$ y sobre los segmentos a bordo de un vehículo de tránsito es el tiempo correspondiente de recorrido entre cada parada. Estas dos características se pueden ver en el par (t_{ij}, f_{ij}) asociado a cada segmento.

Es posible simplificar esta red generalizada sumando el segmentos de abordaje con el primer segmento del itinerario de cada línea y el segmento de bajada con el último segmento del itinerario de cada línea; para lo cual se consideran las siguientes convenciones:

$$(0, f_{ij}) + (t_{ij}, \infty) = (t_{ij}, f_{ij}), \qquad (t_{ij}, \infty) + (0, \infty) = (t_{ij}, \infty).$$

Además; dado que los centroides, en este caso, están conectados únicamente con un nodo regular, entonces podemos quitarlos y renombrar los nodos de conexión. Así, la red generalizada simplificada queda de la siguiente manera

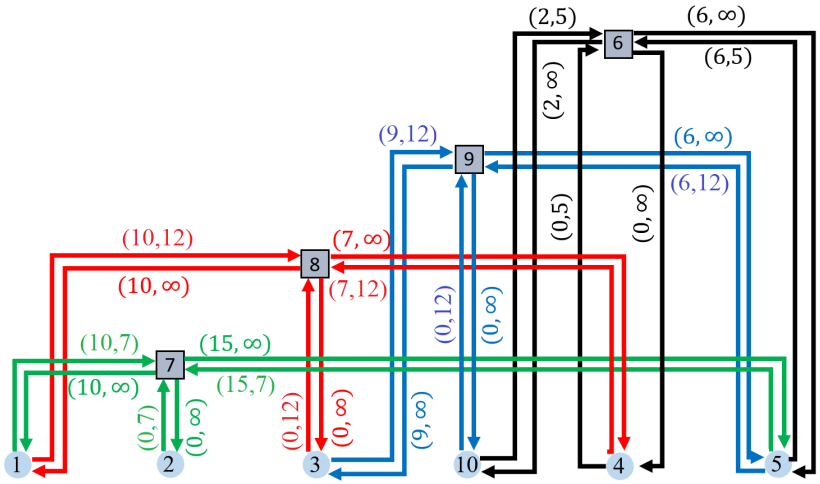


Figura 2.3: Red ejemplo generalizada de tránsito simplificada.

Reacomodando la red para visualizarla más fácilmente se tiene

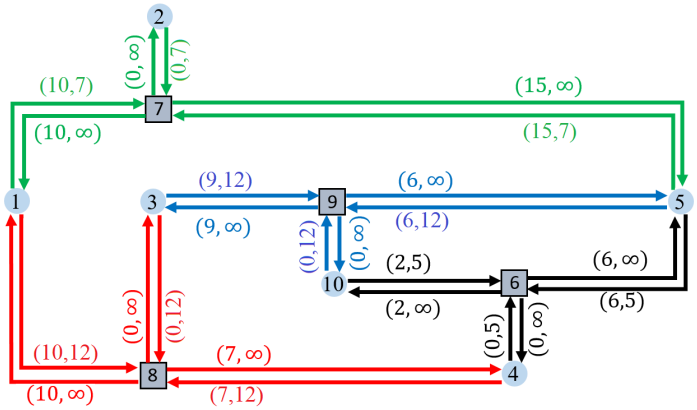


Figura 2.4: Reacomodo de la red ejemplo generalizada de tránsito.

Una vez obtenida la red generalizada de tránsito; se aplica el algoritmo 1 para encontrar el volumen de pasajeros en cada segmento para cada par O-D v_{ij}^{pq} y con ellos las probabilidades de ruta π_{pq}^{ij}. En este ejemplo, se obtienen las siguientes proporciones de demanda

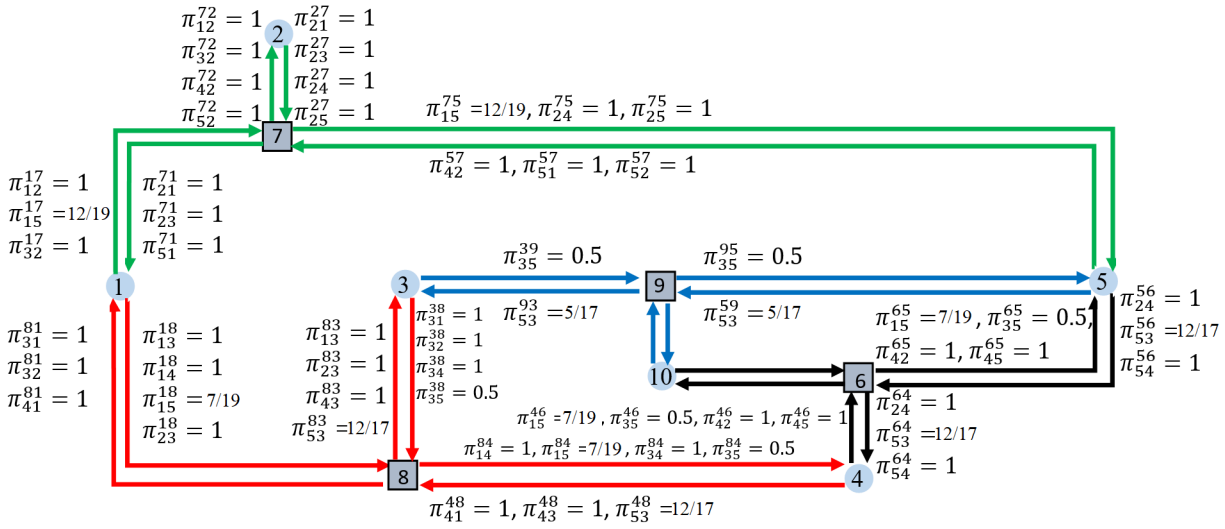


Figura 2.5: Probabilidad de usar cada segmento de la red para cada par O-D.

Ahora, considerando que la demanda de pasajeros entre los centroides, de forma matricial, está dada por

$$\mathbf{g} = \begin{pmatrix} 0 & 10 & 5 & 20 & 76 \\ 5 & 0 & 40 & 30 & 10 \\ 10 & 10 & 0 & 5 & 20 \\ 30 & 40 & 20 & 0 & 30 \\ 50 & 30 & 34 & 40 & 0 \end{pmatrix};$$

donde la entrada $\mathbf{g}_{pq}$ representa la cantidad de viajes que se realizan del nodo $p \in \mathcal{P}$ al nodo $q \in \mathcal{Q}$. Después de asignar la demanda, se obtiene la cantidad de viajes que se realizan utilizando cada segmento (i, j) para cada par O-D (p, q), estos viajes se pueden representar con el producto de la demanda y las probabilidades

$$v_{ij}^{pq} = \pi_{pq}^{ij} g_{pq}.$$

Por ejemplo, la cantidad de pasajeros que van del nodo 5 al nodo 3 y que utilizan el segmento $(4, 8)$ se calcula: $v_{48}^{53} = (12/17)(34) = 24$. Para calcular el volumen total sobre cada segmento, se hace

$$v_{ij} = \sum_{pq \in \mathcal{PQ}} v_{ij}^{pq},$$

entonces para el segmento $(4, 8)$ se tiene

$$v_{48} = \pi_{41}^{48} g_{41} + \pi_{43}^{48} g_{43} + \pi_{53}^{48} g_{53} = (1)(30) + (1)(20) + (12/17)(34) = 74.$$

Observemos que si la matriz de demanda cambia pero la red de tránsito no y se desea saber nuevamente cómo se distribuyen los pasajeros en la red, no es necesario resolver nuevamente el problema de asignación puesto que ya se conocen las proporciones de ruta y sólo es necesario

hacer el producto $\mathbf{v} = P\mathbf{g}$, con

$$\mathbf{v}^T = (v_{17}, v_{71}, v_{72}, v_{27}, v_{75}, ..., v_{65}, v_{56})_{1\times 24}\,, \tag{2.3a}$$

$$P = \begin{pmatrix} \pi_{11}^{17} & \pi_{12}^{17} & \dots & \pi_{15}^{17} & \pi_{21}^{17} & \dots & \pi_{25}^{17} & \pi_{31}^{17} & \dots & \pi_{35}^{17} & \pi_{41}^{17} & \dots & \pi_{45}^{17} & \pi_{51}^{17} & \dots & \pi_{55}^{17} \\ \pi_{11}^{71} & \pi_{12}^{71} & \dots & \pi_{15}^{71} & \pi_{21}^{71} & \dots & \pi_{25}^{71} & \pi_{31}^{71} & \dots & \pi_{35}^{71} & \pi_{41}^{71} & \dots & \pi_{45}^{71} & \pi_{51}^{71} & \dots & \pi_{55}^{71} \\ \pi_{11}^{72} & \pi_{12}^{72} & \dots & \pi_{15}^{72} & \pi_{21}^{72} & \dots & \pi_{25}^{72} & \pi_{31}^{72} & \dots & \pi_{35}^{72} & \pi_{41}^{72} & \dots & \pi_{45}^{72} & \pi_{51}^{72} & \dots & \pi_{55}^{72} \\ \pi_{11}^{27} & \pi_{12}^{27} & \dots & \pi_{15}^{27} & \pi_{21}^{27} & \dots & \pi_{25}^{27} & \pi_{31}^{27} & \dots & \pi_{35}^{27} & \pi_{41}^{27} & \dots & \pi_{45}^{27} & \pi_{51}^{27} & \dots & \pi_{55}^{27} \\ \pi_{11}^{75} & \pi_{12}^{75} & \dots & \pi_{15}^{75} & \pi_{21}^{75} & \dots & \pi_{25}^{75} & \pi_{31}^{75} & \dots & \pi_{35}^{75} & \pi_{41}^{75} & \dots & \pi_{45}^{75} & \pi_{51}^{75} & \dots & \pi_{55}^{75} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \pi_{11}^{65} & \pi_{12}^{65} & \dots & \pi_{15}^{65} & \pi_{21}^{65} & \dots & \pi_{25}^{65} & \pi_{31}^{65} & \dots & \pi_{35}^{65} & \pi_{41}^{65} & \dots & \pi_{45}^{65} & \pi_{51}^{65} & \dots & \pi_{55}^{65} \\ \pi_{11}^{56} & \pi_{12}^{56} & \dots & \pi_{15}^{56} & \pi_{21}^{56} & \dots & \pi_{25}^{56} & \pi_{31}^{56} & \dots & \pi_{35}^{56} & \pi_{41}^{56} & \dots & \pi_{45}^{56} & \pi_{51}^{56} & \dots & \pi_{55}^{56} \end{pmatrix}_{24\times 25}, \tag{2.3b}$$

$$\mathbf{g}^T = (g_{11}, g_{12}, ..., g_{15}, g_{21}, ..., g_{25}, g_{31}, ..., g_{35}, g_{41}, ..., g_{45}, g_{51}, ..., g_{55})_{1\times 25}\,, \tag{2.3c}$$

en donde P es una matriz que contiene las probabilidades de cada segmento para cada par O-D y es de tamaño $no.segmentos \times no.pares\ O-D$, en este caso 24×25; $\mathbf{g}$ es el vector (matriz) de demanda O-D y $\mathbf{v}$ es un vector que contiene el volumen de pasajeros resultante de la asignación en cada segmento. En el anexo B, se puede ver el código de Matlab para resolver este problema.

Capítulo 3

El problema de estimación de matrices O-D

En la sección 2, se describió un modelo de asignación para calcular el volumen de pasajeros en los segmentos de una red de tránsito. Resolver este tipo de problemas ayuda a las personas encargadas de la planificación de redes a determinar las frecuencias de los vehículos, determinar horarios o planificar nuevas rutas. Sin embargo; uno de los elementos más importantes en este proceso de planificación, que también resulta ser uno de los más costosos y difíciles de obtener, es la matriz de demanda O-D. En este capítulo abordaremos diferentes metodologías para estimar dicha demanda a partir de datos relativamente fáciles de obtener.

En la sección 3.1 se plantea el problema general de estimación de matrices O-D. En la sección 3.2 se introduce el modelo penalizado y se demuestra su convergencia al modelo de Spiess cuando el parámetro de penalización tiende a infinito; además, se introduce a detalle el método de gradiente conjugado multiplicativo. En la sección 3.3 se introduce la metodología de Lagrangiano aumentado con el algoritmo de ascenso dual y multiplicadores y se muestra la relación que tienen entre sí los parámetros de penalización. En la sección 3.4 se discuten algunos aspectos para reducir el tamaño del problema y en la sección 3.5 se muestra cómo extender el modelo cuando se cuenta con la información de producciones y atracciones de viajes en algunos centroides. En cada una de estas secciones se aplica la metodología correspondiente en la red ejemplo.

3.1. Problema general

Considérese una red de tránsito que cuenta con n' centroides y m' segmentos dirigidos. Sea $\mathcal{M}$ el conjunto de todas las matrices de demanda origen destino de la red de tránsito. Por simplicidad; podemos acomodar cada una de las matrices de este conjunto de tal forma que cada una de ellas quede representada como un vector $\mathbf{g}$, al cual llamaremos vector origen destino (O-D). Al espacio de vectores O-D le llamaremos $\mathcal{U} = \overline{\mathbb{R}^n_+}$; con $n = (n')^2$, $\mathbb{R}^n_+$ indica el subconjunto de vectores de $\mathbb{R}^n$ con entradas positivas y $\overline{\mathbb{R}^n_+}$ es la cerradura de $\mathbb{R}^n_+$ (el conjunto de vectores con entradas positivas o nulas).

Análogamente, se pueden acomodar los volúmenes de los segmentos de tránsito identificando con un índice a cada par de nodos que definen un segmento; es decir, $(i,j) = 1, 2, ..., m'$. Así, el conjunto de vectores $\mathbf{V}' = \{\mathbf{v} \in \overline{\mathbb{R}_+^{m'}} \,|\mathbf{v} = \{v_a\},\ \text{con } a = 1, 2, ..., m'\}$ representan el volumen de pasajeros que viajan en todos los segmentos de la red de tránsito. Considérese un subconjunto de $\mathbf{V}'$, al cual llamaremos $\mathbf{V} = \{\mathbf{v} \in \overline{\mathbb{R}_+^{m}} \mid \mathbf{v} = \{v_a\},\ \text{con } a = 1, 2, ..., m\}$, con $m < m'$, que representa el flujo de pasajeros en m segmentos de la red de tránsito donde se cuenta con observaciones del volumen promedio de pasajeros.

Considerando que se conoce un vector O-D obsoleto $\hat{\mathbf{g}} \in \mathcal{U}$, que pudo haberse obtenido con encuestas, y que se tienen el flujo promedio de pasajeros $\hat{\mathbf{v}} \in \mathbf{V}$ sobre algunos segmentos de la red de tránsito y las probabilidades resultantes de un proceso de asignación lineal P. Uno de los modelos más usados en la literatura para estimar la demanda "*real*", es el de Spiess (1990), cuya idea original consiste en:

Encontrar un vector de demanda $\mathbf{g}$ *que ajusta los datos de volúmenes* $\hat{\mathbf{v}}$ *y es el más cercano al vector obsoleto* $\hat{\mathbf{g}}$.

El modelo propuesto por Spiess (2.1) se puede formular usando notación vectorial como la minimización de

$$\|P\mathbf{g} - \hat{\mathbf{v}}\|_m^2, \quad \text{sobre el conjunto} \quad \mathcal{U}, \tag{3.1}$$

donde, $\|\cdot\|_m$ indica la norma 2 en $\mathbb{R}^m$. Se supone que $m \ll n$; es decir, el número de segmentos en donde se miden los volúmenes es mucho menor que el número de entradas de la matriz O-D. Observemos que el modelo (3.1) es un problema mal planteado, puesto que permite una infinidad de soluciones. Sin embargo; debido a la estructura algorítmica del método de máximo descenso multiplicativo MDM (1.3), se puede encontrar una aproximación a la solución que es la más cercana a $\hat{\mathbf{g}}$, siempre y cuando se tome como punto inicial $\mathbf{g}^0 = \hat{\mathbf{g}}$ para resolver (3.1). Además; por tratarse de un método multiplicativo, los ceros en la matriz $\hat{\mathbf{g}}$ se preservan en el proceso permitiendo que la nueva estimación $\mathbf{g}$ conserve la estructura.

3.2. Modelo penalizado y su convergencia

En Chávez y Juárez (2014), se introdujo una variante del modelo (3.1), donde la condición de cercanía entre el vector O-D obsoleto $\hat{\mathbf{g}}$ y el estimado $\mathbf{g}$ se incluye directamente en la función objetivo de la siguiente manera:

Dados $\hat{\mathbf{g}} \in \mathcal{U}$ y $\hat{\mathbf{v}} \in \overline{\mathbb{R}_+^m}$, encontrar $\mathbf{g} \in \overline{\mathbb{R}_+^n}$ que minimiza

$$J(\mathbf{g}) = \frac{1}{2}\|\mathbf{g} - \hat{\mathbf{g}}\|_n^2 \quad \text{sobre el conjunto} \quad \mathcal{V} = \{\mathbf{g} \in \mathcal{U} \ :\ P\mathbf{g} = \hat{\mathbf{v}}\}\,. \tag{3.2}$$

Esta formulación del problema permite incorporar la restricción $P\mathbf{g} = \hat{\mathbf{v}}$ en la función objetivo, obteniendo el siguiente modelo penalizado:

Dados $\hat{\mathbf{g}} \in \mathcal{U}$ y $\hat{\mathbf{v}} \in \overline{\mathbb{R}_+^m}$, encontrar $\mathbf{g} \in \mathcal{U}$ que minimiza

$$J_k(\mathbf{g}) = \frac{1}{2}\|\mathbf{g} - \hat{\mathbf{g}}\|_n^2 + \frac{k}{2}\|P\mathbf{g} - \hat{\mathbf{v}}\|_m^2, \tag{3.3}$$

donde $k > 0$ es un parámetro de penalización. Entre más grande sea el parámetro k; más peso tendrá la diferencia de volúmenes en la minimización, lo cual forzará a que volúmenes asignados sean más cercanos a los volúmenes asignados.

3.2.1. Convergencia del modelo penalizado

Dada la matriz P, las probabilidades previamente obtenidas al resolver el problema de asignación, consideremos la solución $\bar{\mathbf{g}}$ del problema (3.2). Para cada $k > 0$, sea $\mathbf{g}_k$ la solución única del problema de minimización:

$$\begin{cases} \mathbf{g}_k \in \mathcal{U}, \\ J_k(\mathbf{g}_k) \leq J_k(\mathbf{g}) \quad \text{para todo} \quad \mathbf{g} \in \mathcal{U}. \end{cases} \tag{3.4}$$

Entonces, por (3.2) y (3.4) se obtiene la siguiente desigualdad:

$$\frac{1}{2}\|\mathbf{g}_k - \hat{\mathbf{g}}\|_n^2 + \frac{k}{2}\|P\mathbf{g}_k - \hat{\mathbf{v}}\|_m^2 \leq \frac{1}{2}\|\bar{\mathbf{g}} - \hat{\mathbf{g}}\|_n^2, \quad \text{para todo} \quad k > 0;$$

la cual, a su vez implica las siguientes dos desigualdades:

$$\|\mathbf{g}_k - \hat{\mathbf{g}}\|_n^2 \leq \|\bar{\mathbf{g}} - \hat{\mathbf{g}}\|_n^2 \qquad \forall\ k > 0, \tag{3.5}$$

$$\|P\mathbf{g}_k - \hat{\mathbf{v}}\|_m^2 \leq \frac{1}{k}\|\bar{\mathbf{g}} - \hat{\mathbf{g}}\|_n^2 \quad \forall\ k > 0. \tag{3.6}$$

Tomando el límite cuando k tiende a infinito en (3.6), se obtiene

$$\lim_{k\to\infty} \|P\mathbf{g}_k - \hat{\mathbf{v}}\|_m^2 \leq \lim_{k\to\infty} \frac{1}{k}\|\bar{\mathbf{g}} - \hat{\mathbf{g}}\|_n^2 \quad \Rightarrow \quad \lim_{k\to\infty} \|P\mathbf{g}_k - \hat{\mathbf{v}}\|_m^2 = 0 \quad \Rightarrow \quad \lim_{k\to\infty} P\mathbf{g}_k = \hat{\mathbf{v}} = P\,\bar{\mathbf{g}}.$$

De acuerdo con (3.5), la sucesión $\{\mathbf{g}_k\}_{k>0}$ está acotada por arriba. Esto es, existe una subsucesión convergente, también llamada $\{\mathbf{g}_k\}$, cuyo límite $\mathbf{g}'$ satisface

$$\|\mathbf{g}' - \hat{\mathbf{g}}\|_n^2 \leq \|\bar{\mathbf{g}} - \hat{\mathbf{g}}\|_n^2. \tag{3.7}$$

Además, como la aplicación $\mathbf{g} \to \|\mathbf{g} - \hat{\mathbf{g}}\|_n^2$ es semi-continua por abajo, se tiene

$$\|\mathbf{g}' - \hat{\mathbf{g}}\|_n^2 \leq \liminf_{k\to\infty} \|\mathbf{g}_k - \hat{\mathbf{g}}\|_n^2 \leq \limsup_{k\to\infty} \|\mathbf{g}_k - \hat{\mathbf{g}}\|_n^2 \leq \|\bar{\mathbf{g}} - \hat{\mathbf{g}}\|_n^2.$$

Dado que $\bar{\mathbf{g}}$ es la solución de norma mínima del problema (3.2), entonces se cumple la igualdad en (3.7), por lo cual

$$\lim_{k\to\infty} \|\mathbf{g}_k - \hat{\mathbf{g}}\|_n^2 = \|\bar{\mathbf{g}} - \hat{\mathbf{g}}, \|_n^2.$$

Por lo tanto,

$$\lim_{k\to\infty} \mathbf{g}_k = \bar{\mathbf{g}}.$$

El análisis previo muestra que cuando k tiende a infinito, la solución $\mathbf{g}_k$ del problema penalizado (3.4) converge a la solución del problema (3.2).

Una observación importante es que el problema (3.4) se puede reformular como un problema de regularización de Tijonov (3.1); así, el problema equivalente es

$$\begin{cases} \mathbf{g}_\alpha \in \mathcal{U}, \\ J_\alpha(\mathbf{g}_\alpha) \leq J_\alpha(\mathbf{g}) \quad \text{para todo} \quad \mathbf{g} \in \mathcal{U}, \end{cases}$$

donde la función de minimización J_α se define como

$$J_\alpha(\mathbf{g}) = \frac{\alpha}{2}\|\mathbf{g} - \hat{\mathbf{g}}\|_n^2 + \frac{1}{2}\|P\mathbf{g} - \hat{\mathbf{v}}\|_m^2, \tag{3.8}$$

con el parámetro de regularización $\alpha > 0$ pequeño. Este problema se relaciona con (3.4) tomando $\alpha = 1/k$ y se puede ver inmediatamente que $J_\alpha(\mathbf{g}) \to (1/2)\|P\mathbf{g} - \hat{\mathbf{v}}\|_m^2$ cuando $\alpha \to 0$. Así, del análisis previo se concluye que

$$\lim_{\alpha\to 0} \mathbf{g}_\alpha = \lim_{k\to\infty} \mathbf{g}_k = \bar{\mathbf{g}}$$

y $\bar{\mathbf{g}}$ es la solución de los problemas (3.1) y (3.2). Concluimos que el modelo de Spiess se puede ver como el límite de los modelos de penalización (regularización). Finalmente queremos mencionar que algunos autores, como Noriega y Florian (2009) y Verbas et al. (2011), prefieren usar modelos cuadráticos con promedios ponderados de la siguiente forma

$$J_\beta(\mathbf{g}) = \frac{\beta}{2}\|P\mathbf{g} - \hat{\mathbf{v}}\|_m^2 + \frac{1-\beta}{2}\|\mathbf{g} - \hat{\mathbf{g}}\|_n^2,$$

donde $0 \leq \beta \leq 1$. Estos modelos también son equivalentes a nuestro modelo penalizado (3.3), y su relación se obtiene eligiendo

$$k = \frac{\beta}{1-\beta} \quad \Longleftrightarrow \quad \beta = \frac{k}{k+1}. \tag{3.9}$$

Así, podemos afirmar que

$$\lim_{\beta\to 1} \mathbf{g}_\beta = \lim_{k\to\infty} \mathbf{g}_k = \bar{\mathbf{g}} \quad \text{and} \quad \lim_{\beta\to 1} P\,\mathbf{g}_\beta = \lim_{k\to\infty} P\,\mathbf{g}_k = \hat{\mathbf{v}}.$$

3.2.2. El algoritmo GCM

En Chávez y Juárez (2014) y Juárez y Chávez (2014), introdujimos el método de gradiente conjugado multiplicativo (GCM), el cual consiste en encontrar iterativamente la matriz de demanda O-D $\mathbf{g} = \{g_i\}$, con $i = 1, 2, ..., n$, usando la siguiente fórmula iterativa

$$g_i^{\ell+1} = \begin{cases} \hat{g}_i & \text{para } \ell = 0, \\ g_i^\ell \left(1 + \gamma_\ell\, d_i^\ell\right) & \text{para } \ell \geq 1 \end{cases} \quad \forall\ i = 1, 2, ..., n,$$

donde d_i^ℓ es la dirección de descenso y γ_ℓ es el tamaño de paso. Conociendo el vector de dirección de descenso $\mathbf{d}^\ell = \{d_i^\ell\}_{i=1,...,n}$ en la iteración ℓ, el tamaño de paso γ_ℓ se calcula como el mínimo de la función escalar

$$\phi(\gamma) = J_k(\mathbf{g}^\ell + \gamma\, \mathbf{d}^\ell). \tag{3.10}$$

La primera dirección de descenso se elige como el negativo del gradiente evaluado en el punto inicial, $\mathbf{d}^0 = -\nabla J_k(\hat{\mathbf{g}})$ y se actualiza en cada iteración con la siguiente fórmula

$$d_i^{\ell+1} = -g_i^{\ell+1}\frac{\partial J_k(\mathbf{g}^{\ell+1})}{\partial g_i} + \beta_\ell\, d_i^\ell, \quad \forall\ i = 1, ..., n,$$

donde la constante β_ℓ se calcula de tal forma que dos direcciones consecutivas $\mathbf{d}^\ell$ y $\mathbf{d}^{\ell+1}$ sean conjugadas entre sí.

Observemos que el gradiente de la función de costo cuadrática del problema (3.3) es

$$\nabla J_k(\mathbf{g}) = \mathbf{g} - \hat{\mathbf{g}} + k\, P^T(P\mathbf{g} - \hat{\mathbf{v}}) = Q_k\mathbf{g} - \mathbf{b}_k,$$

donde $Q_k = I_n + k\, P^T P$ con I_n la matriz identidad de tamaño $n \times n$, y $\mathbf{b}_k = \hat{\mathbf{g}} + k\, P^T\hat{\mathbf{v}}$. La matriz Q_k es definida positiva para cada constante $k > 0$, por lo tanto existe su inversa. De aquí en adelante, usaremos la notación $\mathbf{x} \odot \mathbf{y}$ para indicar la multiplicación elemento a elemento de vectores en $\mathbb{R}^n$, el cual se conoce como el producto de Hadamard, esto significa que el vector $\mathbf{z} = \mathbf{x} \odot \mathbf{y}$ tiene componentes $z_i = x_i\, y_i$. El algoritmo GCM, paso a paso, es el siguiente:

Algoritmo 2 Gradiente conjugado multiplicativo (GCM).

Entrada: $\hat{\mathbf{g}}$, k, Q_k, $\mathbf{b}_k$ y $\varepsilon > 0$ (una tolerancia pequeña).
Salida: Una matriz O-D $\mathbf{g}$ actualizada.

1: Elegir el punto de inicio $\mathbf{g}^0 = \hat{\mathbf{g}}$. ▷ **Inicialización**
2: Evaluar el primer gradiente: $\mathbf{r}^0 \doteq \nabla J_k(\mathbf{g}^0) \equiv Q_k(\mathbf{g}) - \mathbf{b}_k$.
3: Calcular el primer gradiente multiplicativo: $\mathbf{r}_M^0 = \mathbf{g}^0 \odot \mathbf{r}^0$.
4: Establecer la primera dirección de descenso multiplicativa: $\mathbf{d}_M^0 = -\mathbf{r}_M^0$.
5: **Para** $\ell \geq 0$, **hacer** ▷ **Descenso**.
6: Calcular la solución γ_ℓ de (3.10) tal que: $g_i^\ell + \gamma\,(d_M^\ell)_i \geq 0$ para todo $i = 1, ..., n$.
7: Actualizar la demanda: $\mathbf{g}^{\ell+1} = \mathbf{g}^\ell + \gamma_\ell\, \mathbf{d}_M^\ell$.
8: Actualizar el gradiente: $\mathbf{r}^{\ell+1} = \mathbf{r}^\ell + \gamma_\ell\, Q_k\mathbf{d}_M^\ell$.
9: Calcular el gradiente multiplicativo: $\mathbf{r}_M^{\ell+1} = \mathbf{g}^{\ell+1} \odot \mathbf{r}^{\ell+1}$.
10: **Si** $\|\mathbf{r}^{\ell+1}\|_n \leq \varepsilon\, \|\mathbf{r}^0\|_n$ **entonces** ▷ **Prueba de convergencia**
11: tomar $\mathbf{g}_k = \mathbf{g}^{\ell+1}$ y **parar**,
12: **si no**
13: Calcular: $\beta_\ell = \dfrac{\mathbf{r}_M^{\ell+1} Q_k \mathbf{d}_M^\ell}{(\mathbf{d}_M^\ell)^T Q_k \mathbf{d}_M^\ell}$.
14: $\mathbf{d}_M^{\ell+1} = -\mathbf{r}_M^{\ell+1} + \beta_\ell\, \mathbf{d}_M^\ell$. ▷ **Nueva dirección de descenso**
15: Hacer: $\ell = \ell + 1$ y regresar al paso 5.

En el algoritmo anterior empleamos la notación $\mathbf{r}^\ell = \nabla J_k(\mathbf{g}^\ell)$. Queremos hacer énfasis en que la única diferencia entre el método GCM y el método de gradiente conjugado (GC) estándar es el cálculo del gradiente multiplicativo con el producto de Hadamard en los pasos 3 y 9 en cada iteración, así que el costo computacional adicional es marginal. Sin embargo, debemos tener cuidado en el paso de inicio 1, la estructura multiplicativa del algorithm requiere que el punto inicial $\mathbf{g}^0 = \hat{\mathbf{g}}$ sea diferente a $\mathbf{0} \in \mathbb{R}^n$, de otro modo $\mathbf{g}^\ell$ permanecerá como $\mathbf{0}$ en cada iteración. Observemos que la fórmula en el paso 8 se obtiene de la fórmula en el paso 7 multiplicando por Q_k y restando $\mathbf{b}_k$ en ambos lados. Para calcular γ_ℓ en el paso 6, primero calculamos el valor de γ tal que $\phi'_\ell(\gamma) = \nabla J_k(\mathbf{g}^\ell + \gamma\, \mathbf{d}^\ell_{M})^T \mathbf{d}^\ell_{M} = (\mathbf{r}^{\ell+1})^T \mathbf{d}^\ell_{M} = 0$; este valor es

$$\gamma = -(\mathbf{r}^\ell)^T \mathbf{d}^\ell_{M} / (\mathbf{d}^\ell)^T_{M} Q_k \mathbf{d}^\ell_{M}, \tag{3.11}$$

el cual frecuentemente satisface la restricción dada $g^\ell_i + \lambda\, (d^\ell_{M})_i \geq 0$ para todo $i = 1, ..., n$; pero si no se satisface, tomamos $g^{\ell+1}_i = 0$ (see Vollebregt, 2014).

El valor para β_ℓ en el paso 13 es tal que $\mathbf{d}^{\ell+1}_{M}$ y $\mathbf{d}^\ell_{M}$ son Q_k–conjugadas. Así, el algoritmo se puede reescribir para calcular $Q_k \mathbf{d}^\ell_{M}$ y $\left(\mathbf{d}^\ell_{M}\right)^T Q_k \mathbf{d}^\ell_{M}$ una vez en cada iteración. La igualdad en el paso 8 implica que β_ℓ también es igual a

$$\beta_\ell = \frac{\left(\mathbf{r}^{\ell+1}_{M}\right)^T \left(\mathbf{r}^{\ell+1} - \mathbf{r}^\ell\right)}{(\mathbf{d}^\ell_{M})^T \left(\mathbf{r}^{\ell+1} - \mathbf{r}^\ell\right)}, \tag{3.12}$$

la cual es similar a la fórmula de Hestenes–Stiefel para el algoritmo GC (Nocedal y Wright, 2006).

El algoritmo GCM garantiza que dos direcciones consecutivas $\mathbf{d}^\ell_{M}$ y $\mathbf{d}^{\ell+1}_{M}$ sean Q_k–conjugadas; sin embargo no hay garantía de que todas las direcciones de descenso generadas en el proceso iterativo sean conjugadas entre sí. De hecho, los experimentos numéricos muestran que a lo más tres iteraciones sucesivas producen vectores Q_k-conjugados.

3.2.3. Ejemplo de aplicación del algoritmo GCM

Consideremos nuevamente la red ejemplo de la figura 2.1 y el vector de demanda O-D exacta

$$\bar{\mathbf{g}}^T = (0, 10, 5, 20, 76, 5, 0, 40, 30, 10, 10, 10, 0, 5, 20, 30, 40, 20, 0, 30, 50, 30, 34, 40, 0)\,.$$

La matriz de probabilidades de ruta P_c que se se obtuvo de resolver el problema de asignación

en la sección 2.3 es la siguiente:

$$P_c = \begin{pmatrix}
0 & 1 & 0 & 0 & 12/19 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 12/19 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 7/19 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 12/17 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 7/19 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 12/17 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5/17 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5/17 & 0 & 0 \\
0 & 0 & 0 & 0 & 7/19 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12/17 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 7/19 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12/17 & 1 & 0
\end{pmatrix}$$

cuyo rango es 14. Al realizar una asignación de tránsito se obtienen los volúmenes

$$\hat{\mathbf{v}}_c = P_c\bar{\mathbf{g}} = (68, 95, 90, 85, 88, 120, 93, 50, 89, 35, 63, 74, 10, 10, 0, 0, 10, 10, 78, 94, 0, 0, 78, 94),$$

de los cuales, seleccionamos los segmentos $(5, 7)$, $(1, 8)$ y $(5, 6)$ para obtener el vector que jugará el papel de volúmenes observados $\hat{\mathbf{v}} = (v_{5,7}, v_{1,8}, v_{5,6})^T = (120, 93, 94)^T$. Así, la matriz de probabilidades reducida, que corresponde únicamente a los segmentos donde se tienen datos, nos queda

$$P = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 7/19 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12/17 & 1 & 0
\end{pmatrix}.$$

Para utilizar el algoritmo GCM se genera un ejemplo sintético; en donde se perturba el vector $\bar{\mathbf{g}}$ uniformemente alrededor del 20 %, usando la función *rand* en Matlab, de la forma $\hat{\mathbf{g}} = 0.8\bar{\mathbf{g}} + 0.4\bar{\mathbf{g}} \odot rand(25, 1)$. En este ejemplo se redondea a enteros el resultado de la perturbación obteniendo:

$$\hat{\mathbf{g}}^T = (0, 12, 4, 23, 84, 4, 0, 41, 35, 12, 9, 12, 0, 5, 22, 26, 39, 23, 0, 36, 53, 24, 34, 47, 0),$$

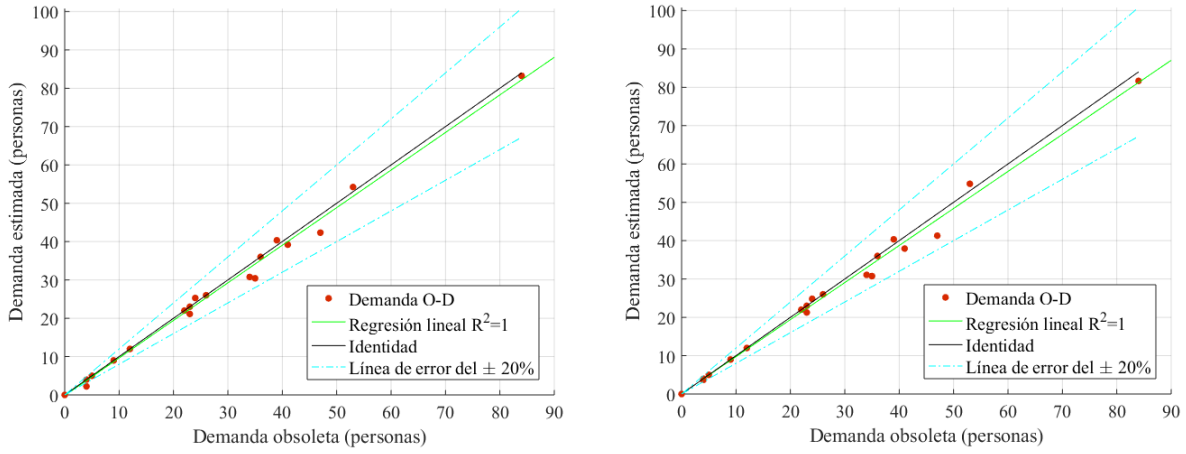
con $\|\bar{\mathbf{g}} - \hat{\mathbf{g}}\|/\|\bar{\mathbf{g}}\| = 0.12$.

Así, el problema consiste en encontrar el vector $\mathbf{g}$ más cercano a $\hat{\mathbf{g}}$, tal que $P\mathbf{g} = \hat{\mathbf{v}}$ con $g_i \geq 0,\ i = 1, 2, ..., 25$. Aplicando el algoritmo 2, con $\varepsilon = 10^{-3}$ y diferentes valores del coeficiente de penalización k, se obtiene

Tabla 3.1: Comparación de los algoritmos MDM y GCM en la red ejemplo.

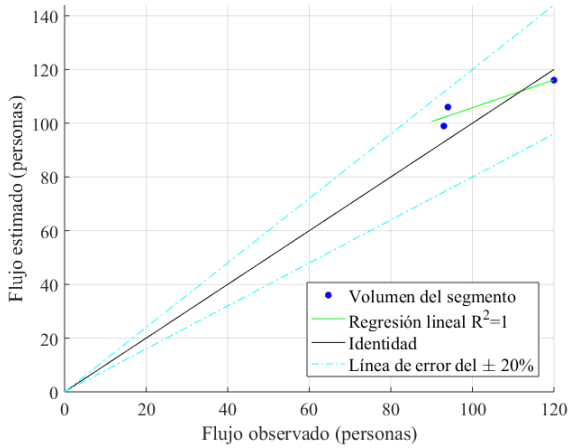
k	Método	Iters	$RMSE_v$	$\|\|P\mathbf{g} - \hat{\mathbf{v}}\|\|_m$	$RMSE_g$	$\|\|\mathbf{g} - \hat{\mathbf{g}}\|\|_n$
Inicial	$(l = 0)$		8.07	14.0		
10	MDM	339	0.30	0.5	1.66	8.3
	GCM	14	0.30	0.5	1.66	8.3
100	MDM	173	0.03	0.1	1.75	8.8
	GCM	7	0.03	0.1	1.76	8.8
1000	MDM	5	0.00	0.0	1.82	9.1
	GCM	3	0.00	0.0	1.82	9.1
∞	MDM	5	0.00	0.0	1.82	9.1
	GCM	3	0.01	0.0	1.82	9.1

La figura 3.1 muestra la demanda *a priori* (eje x) contra la demanda estimada (eje y) con el método GCM para $k = 10$, parte (a), y $k = \infty$, parte (b). Por otra parte, la figura 3.2, muestra los volúmenes observados (eje x) contra los volúmenes calculados con la demanda *a priori* (eje y), parte (a); los volúmenes observados contra los volúmenes estimados con $k = 10$, parte (b), y los volúmenes observados contra los volúmenes estimados con $k = \infty$, parte (c). Todos los puntos que están sobre la identidad representan un ajuste perfecto entre los valores estimados correspondientes y los datos, mientras que los puntos que están entre las dos líneas punteadas corresponden a un ajuste con un error relativo menor al 20 %.

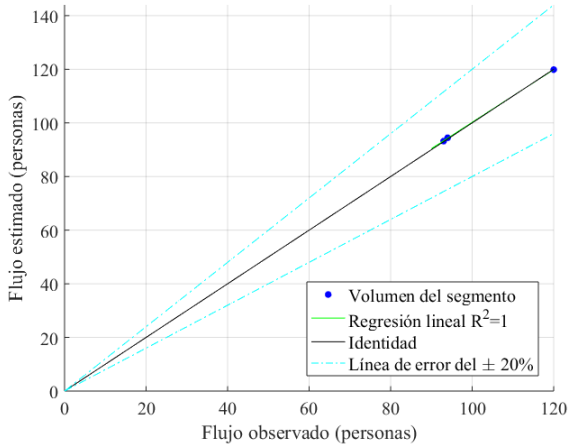


(a) $k = 10$. (b) $k = \infty$.

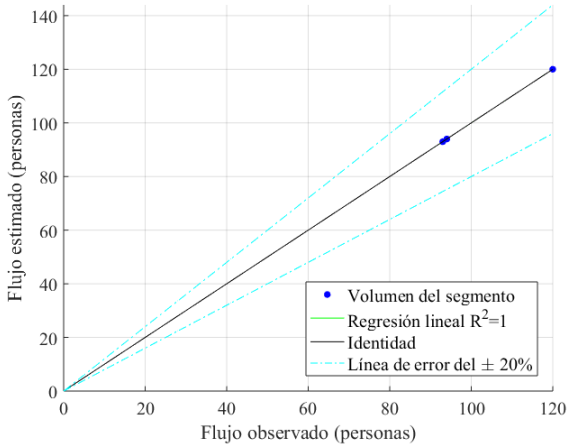
Figura 3.1: Demanda *a priori* vs demanda estimada con el método GCM para la red ejemplo.



(a) Inicial.



(b) $k = 10$.



(c) $k = \infty$.

Figura 3.2: Volúmenes observados vs volúmenes estimados con el método GCM para la red ejemplo.

Las líneas verdes representan la regresión lineal. Los valores correspondientes para la pendiente m y la ordenada al origen b se muestran en la tabla 3.2, dichos valores muestran existe una fuerte correlación entre los parámetros estimados y los datos.

Tabla 3.2: Parámetros correspondientes a las rectas de regresión resultantes.

k	Método	m_v	b_v	m_g	b_g
Inicial	$(l=0)$	0.52	54.03	1.00	0.0
10	MDM	0.98	1.89	0.98	-0.17
	GCM	0.98	1.90	0.98	-0.17
100	MDM	1.00	0.20	0.97	-0.05
	GCM	1.00	0.20	0.98	-0.11
1000	MDM	1.00	0.02	0.97	0.09
	GCM	1.00	-0.02	0.97	0.09
∞	MDM	1.00	0.00	0.97	0.09
	GCM	1.00	-0.05	0.97	0.09

Finalmente, la matriz (vector) de demanda O-D actualizada y el vector de volúmenes estimados para $k = \infty$ nos queda:

$$\mathbf{g} = \begin{pmatrix} 0 & 12 & 3.7 & 21.3 & 81.7 \\ 4 & 0 & 37.9 & 30.8 & 12 \\ 9 & 12 & 0 & 5 & 22 \\ 26 & 40.3 & 23 & 0 & 36 \\ 54.8 & 24.8 & 31.1 & 41.3 & 0 \end{pmatrix}, \qquad \mathbf{v} = \begin{pmatrix} 120 \\ 93 \\ 94 \end{pmatrix}.$$

Observemos que la matriz O-D resultante $\mathbf{g}$ no es necesariamente igual a la matriz de demanda exacta; puesto que en el modelo se busca la matriz más cercana a la matriz de referencia $\hat{\mathbf{g}}$ y en este caso $\|\hat{\mathbf{g}} - \mathbf{g}\| = 9.1$, mientras que $\|\hat{\mathbf{g}} - \bar{\mathbf{g}}\| = 16.6$.

3.3. Enfoque de Lagrangiano aumentado

Una de las características principales de los algoritmos multiplicativos (MDM y GCM) es que no permiten que las entradas nulas en el vector O-D obsoleto evolucionen a valores positivos en el vector actualizado (o viceversa) y esto podría ser una condición no requerida para algunas instancias. Además, desde el punto de vista computacional, en el algoritmo GCM no todas las direcciones de descenso $\left\{\mathbf{d}_M^\ell\right\}_\ell$ son Q_k–conjugadas entre sí; lo cual puede degradar el desempeño del método y la precisión de la solución numérica. Una alternativa es incluir de forma explícita las condiciones de no negatividad, $g_i \geq 0,\ \forall i = 1, ..., n$, en la función objetivo, evitando la estructura multiplicativa de los algoritmos iterativos vistos en la sección anterior y permitiendo así el uso del algoritmo de gradiente conjugado (CG) estándar.

Partiendo del modelo penalizado (3.3), podemos usar un enfoque Lagrangiano para lidiar con las restricciones de no–negatividad para los coeficientes del vector O-D. Introduciendo un nuevo vector de variables $\mathbf{y} \in \mathbb{R}^n$ que satisface

$$y_i^2 = g_i \quad \text{para todo} \quad i = 1, ..., n. \tag{3.13}$$

Con (3.13) transformamos las restricciones de desigualdad $g_i \geq 0$ a restricciones de igualdad. Entonces el problema de minimización (3.3)–(3.4) es equivalente a minimizar el funcional cuadrático de costo (3.3) sujeto a

$$\mathbf{g} = \mathbf{y} \odot \mathbf{y} \quad \text{and} \quad \mathbf{y} \in \mathbb{R}^n.$$

Una forma muy común de tratar con restricciones de igualdad (Nocedal y Wright, 2006) es introduciendo una nueva variable, llamada multiplicador de Lagrange, por cada restricción; a las cuales denotaremos como el vector $\boldsymbol{\mu} \in \mathbb{R}^n$ formando así el siguiente Lagrangiano

$$\mathcal{L}_k(\mathbf{g}, \mathbf{y}, \boldsymbol{\mu}) = J_k(\mathbf{g}) + \boldsymbol{\mu}^T(\mathbf{y} \odot \mathbf{y} - \mathbf{g}). \tag{3.14}$$

Así, un punto crítico $(\mathbf{g}, \mathbf{y}, \boldsymbol{\mu})$ satisface las condiciones de Karush–Kuhn–Tucker (KKT):

$$\nabla_{\mathbf{g}}\mathcal{L}_k(\mathbf{g}, \mathbf{y}, \boldsymbol{\mu}) = (\mathbf{g} - \hat{\mathbf{g}}) + k\, P^T(P\mathbf{g} - \hat{\mathbf{v}}) - \boldsymbol{\mu} = Q_k\mathbf{g} - \mathbf{b}_k - \boldsymbol{\mu} = \mathbf{0}, \tag{3.15}$$

$$\nabla_{\mathbf{y}}\mathcal{L}_k(\mathbf{g}, \mathbf{y}, \boldsymbol{\mu}) = 2\boldsymbol{\mu} \odot \mathbf{y} = \mathbf{0}, \tag{3.16}$$

$$\nabla_{\boldsymbol{\mu}}\mathcal{L}_k(\mathbf{g}, \mathbf{y}, \boldsymbol{\mu}) = \mathbf{y} \odot \mathbf{y} - \mathbf{g} = \mathbf{0}, \tag{3.17}$$

donde

$$Q_k = I + k\, P^T P \quad \text{and} \quad \mathbf{b}_k = \hat{\mathbf{g}} + k\, P^T\hat{\mathbf{v}}. \tag{3.18}$$

El sistema no lineal (3.15)–(3.17) se puede resolver mediante un proceso iterativo; sin embargo, el punto de inicio dado por $\mathbf{g}^0 = \hat{\mathbf{g}}$, $\mathbf{y}^0 = \sqrt{\hat{\mathbf{g}}}$ (raíz cuadrada componente a componente) podría llevar a obtener valores muy grandes para los multiplicadores en la primera iteración (y algunas de las siguientes), puesto que $\boldsymbol{\mu}^j = Q_k\mathbf{g}^j - \mathbf{b}_k = \nabla J_k(\mathbf{g}^j)$, especialmente para valores muy grandes del parámetro de penalización k. Para evitar la posible inestabilidad, convexificamos el Lagrangiano (3.14) introduciendo el siguiente Lagrangiano aumentado:

$$\mathcal{L}_{k,\rho}(\mathbf{g}, \mathbf{y}, \boldsymbol{\mu}) = J_k(\mathbf{g}) + \boldsymbol{\mu}^T(\mathbf{y} \odot \mathbf{y} - \mathbf{g}) + \frac{\rho}{2}\|\mathbf{y} \odot \mathbf{y} - \mathbf{g}\|_n^2, \tag{3.19}$$

donde k y ρ son constantes positivas. Ahora las condiciones KKT son:

$$\begin{aligned}\nabla_{\mathbf{g}}\mathcal{L}_{k,\rho}(\mathbf{g}, \mathbf{y}, \boldsymbol{\mu}) &= \mathbf{g} - \hat{\mathbf{g}} + k\, P^T(P\mathbf{g} - \hat{\mathbf{v}}) - \boldsymbol{\mu} + \rho(\mathbf{g} - \mathbf{y} \odot \mathbf{y}) \\ &= Q_{k,\rho}\mathbf{g} - \mathbf{b}_k - \boldsymbol{\mu} - \rho\, \mathbf{y} \odot \mathbf{y} = \mathbf{0}, \end{aligned} \tag{3.20}$$

$$\nabla_{\mathbf{y}}\mathcal{L}_{k,\rho}(\mathbf{g}, \mathbf{y}, \boldsymbol{\mu}) = 2\left[\boldsymbol{\mu} + \rho(\mathbf{y} \odot \mathbf{y} - \mathbf{g})\right] \odot \mathbf{y} = \mathbf{0}, \tag{3.21}$$

$$\nabla_{\boldsymbol{\mu}}\mathcal{L}_{k,\rho}(\mathbf{g}, \mathbf{y}, \boldsymbol{\mu}) = \mathbf{y} \odot \mathbf{y} - \mathbf{g} = \mathbf{0}, \tag{3.22}$$

con $\mathbf{b}_k$ el vector definido en (3.18) y

$$Q_{k,\rho} = (1 + \rho)\, I + k\, P^T P.$$

Observemos que si se sustituye (3.22) en (3.20) y (3.21), se recupera el sistema (3.15)–(3.17), por lo tanto ambos sistemas son equivalentes. Sin embargo, algunas propiedades algorítmicas y computacionales del sistema (3.20)-(3.22) que podrían ser ventajosas sobre el sistema (3.15)-(3.17) son:

1. $Q_{k,\rho}$ es una matriz simétrica y definida positiva con mejor número de condición que Q_k para el mismo valor de k, puesto que $\rho > 0$. Por ejemplo, la matriz Q_{100} asociada al problema resuelto en la sección 3.2.3, tiene un número de condición de 315, mientras que la matriz $Q_{100,2}$ para el mismo problema tiene un número de 106 (con $\rho = 2$).

2. El valor de ρ en (3.19) no necesariamente tiene que ser grande, puesto que la restricción $\mathbf{g} = \mathbf{y} \odot \mathbf{y}$ se relaja con el multiplicador de Lagrange. Este parámetro se puede escoger con base en el valor de k cuando se asigna un valor constante a la relación $k/(1+\rho)$, como se muestra en los resultados numéricos. Ver subsección 3.3.2 para más detalles.

3. Cuando se aplica un proceso iterativo para obtener una aproximación a la solución del problema, tenemos la opción de actualizar el multiplicador $\boldsymbol{\mu}$ usando la ecuación (3.21) como se muestra en 3.25 del siguiente algoritmo; mientras que en el sistema (3.15)-(3.17) la única opción es usando (3.20).

3.3.1. El algoritmo ADMM

De cuerdo con las observaciones en la sección anterior, proponemos el algoritmo de ascenso dual y método de multiplicadores (ADMM) (Boyd et al., 2010):

Algoritmo 3 Ascenso dual y método de multiplicadores (ADMM).

Entrada: $\hat{\mathbf{g}}$ y $\varepsilon > 0$ (tolerancia pequeña).
Salida: Vector O-D $\mathbf{g}$ actualizado.

1: Tomar como punto de inicio $\boldsymbol{\mu}^0 = 0$ y $\mathbf{y}^0 \odot \mathbf{y}^0 = \hat{\mathbf{g}}$. ▷ **Inicialización**
2: **Para** $\jmath \geq 1$, dado $\boldsymbol{\mu}^{\jmath-1}$ y $\mathbf{y}^{\jmath-1}$, para calcular $\mathbf{g}^\jmath$ y $\mathbf{y}^\jmath$ **hacer** ▷ **Iteraciones**.
3: $\quad \mathbf{g}^\jmath = \arg\min_{\mathbf{g}} \mathcal{L}_{k,\rho}(\mathbf{g}, \mathbf{y}^{\jmath-1}, \boldsymbol{\mu}^{\jmath-1})$, (3.23)
4: $\quad \mathbf{y}^\jmath = \arg\min_{\mathbf{y}} \mathcal{L}_{k,\rho}(\mathbf{g}^\jmath, \mathbf{y}, \boldsymbol{\mu}^{\jmath-1})$. (3.24)
5: $\quad$ **Si** $\|\mathbf{y}^\jmath \odot \mathbf{y}^\jmath - \mathbf{g}^\jmath\|_n \leq \varepsilon \|\hat{\mathbf{g}}\|_n$, **entonces** ▷ **Prueba de convergencia**
6: $\quad\quad$ Hacer $\mathbf{g} = \mathbf{g}^\jmath$ y **parar**,
7: $\quad$ **si no** ▷ **Actualizar**
8: $\quad\quad \boldsymbol{\mu}^\jmath = \boldsymbol{\mu}^{\jmath-1} + \rho\,(\mathbf{y}^\jmath \odot \mathbf{y}^\jmath - \mathbf{g}^\jmath)$. (3.25)
9: $\quad$ Hacer $\jmath = \jmath + 1$ y regresar al paso 2.

El problema (3.23) involucra la minimización de una función estrictamente cuadrática sobre $\mathbb{R}^n$ con matriz Hessiana constante igual a $Q_{k,\rho}$; así, el único mínimo $\mathbf{g}^\jmath$ satisface el sistema lineal

$$Q_{k,\rho}\mathbf{g} = \mathbf{b}_k + \boldsymbol{\mu}^{\jmath-1} + \rho\,\mathbf{y}^{\jmath-1} \odot \mathbf{y}^{\jmath-1}, \tag{3.26}$$

el cual se puede aproximar con el algoritmo GC con punto inicial $\mathbf{g}^0 = \mathbf{g}^{\jmath-1}$. Observemos que el algoritmo GC nos puede dar soluciones con algunas entradas de $\mathbf{g}^\jmath$ menores que cero, por lo cual se propone parar las iteraciones en GC si para algún ℓ se obtiene que el mín$(\mathbf{g}^\ell) \leq -0.25$ y, en este caso, continuar con el paso 4 del algoritmo ADMM. Así, el algoritmo ADMM puede ser más eficiente si en el paso 3 se realizan sólo algunas iteraciones en el algoritmo de gradiente conjugado sin alcanzar necesariamente la convergencia y continuar con el paso 4

para actualizar el multiplicador; especialmente cuando el algoritmo GC nos lleva a vectores de demanda (matrices) con coeficientes negativos. Así, el algoritmo GC termina en la iteración ℓ si la norma del gradiente $\|Q\mathbf{g}^{\jmath,\ell} - \mathbf{b}_k - \boldsymbol{\mu}^{\jmath-1} - \rho\,\mathbf{y}^{\jmath-1} \odot \mathbf{y}^{\jmath-1}|| \leq \varepsilon \|Q\mathbf{g}^{\jmath,0} - \mathbf{b}_k - \boldsymbol{\mu}^{\jmath-1} - \rho\,\mathbf{y}^{\jmath-1} \odot \mathbf{y}^{\jmath-1}||$ (como en el algoritmo GCM), o bien si mín$(\mathbf{g}^{\ell}) \leq -0.25$.

El problema (3.24) involucra minimizar

$$f(\mathbf{y}) = \frac{\rho}{2}\|\mathbf{y} \odot \mathbf{y} - \mathbf{g}^{\jmath}\|_n^2 + (\boldsymbol{\mu}^{\jmath-1})^T(\mathbf{y} \odot \mathbf{y} - \mathbf{g}^{\jmath}) + J_k(\mathbf{g}^{\jmath}) \tag{3.27}$$

con respecto a $\mathbf{y}$. Un punto crítico $\mathbf{y} = \{y_i\}_{i=1}^n$ satisface

$$\frac{\partial f}{\partial y_i}(\mathbf{y}) = 2\left[\rho\,\left(y_i^2 - g_i^{\jmath}\right) + \mu_i^{\jmath-1}\right]\, y_i = 0 \quad \text{for } i = 1, \ldots, n,$$

esto quiere decir que $y_i = 0$ o $y_i^2 = g_i^{\jmath} - \mu_i^{\jmath-1}/\rho$. Así que el punto crítico $\mathbf{y}^{\jmath}$ se construye de la siguiente manera:

Para cada $i = 1, \ldots, n$:

$$\text{si } g_i^{\jmath} - \mu_i^{\jmath-1}/\rho > 0 \text{ entonces } (y_i^{\jmath})^2 = g_i^{\jmath} - \mu_i^{\jmath-1}/\rho, \text{ de lo contrario } y_i^{\jmath} = 0. \tag{3.28}$$

Esta vez, la Hessiana de $f(\mathbf{y})$ es una matriz diagonal D con entradas

$$D_{ii}(\mathbf{y}) = \frac{\partial^2 f}{\partial y_i^2} = 2\,\left[\rho\,\left(y_i^2 - g_i^{\jmath}\right) + \mu_i^{\jmath-1} + 2\rho\,y_i^2\right],$$

y al evaluarla en el punto crítico (3.28) resulta

$$D_{ii}(\mathbf{y}^{\jmath}) = \begin{cases} 4\rho\,\left(y_i^{\jmath}\right)^2, & \text{si} \quad (y_i^{\jmath})^2 = g_i^{\jmath} - \mu_i^{\jmath-1}/\rho > 0, \\ 2\rho\,(\mu_i^{\jmath-1}/\rho - g_i^{\jmath}), & \text{si} \quad y_i^{\jmath} = 0. \end{cases}$$

Así, la Hessiana tiene entradas positivas en la diagonal, excepto cuando $g_i^{\jmath} - \mu_i^{\jmath-1}/\rho = 0$, cuya entrada correspondiente en la diagonal se vuelve cero en la iteración $\jmath$. Por lo tanto, en general la matriz Hessiana es semi definida positiva. Sin embargo, el punto crítico (3.28) aún nos lleva a una buena estimación del mínimo en (3.24). Una forma de ver esto es mediante la función cuadrática para $\mathbf{z} = \mathbf{y} \odot \mathbf{y}$:

$$f(\mathbf{z}) = \frac{\rho}{2}\|\mathbf{z} - \mathbf{g}^{\jmath}\|_n^2 + (\boldsymbol{\mu}^{\jmath-1})^T(\mathbf{z} - \mathbf{g}^{\jmath}) + J_k(\mathbf{g}^{\jmath}).$$

Entonces, la función original de cuarto orden $f(\mathbf{y})$ en (3.27) se convierte en una nueva función cuadrática en la variable $\mathbf{z}$ con coeficientes no negativos. Para esta nueva función, el gradiente y la Hessiana son respectivamente

$$\nabla f(\mathbf{z}) = \rho\,(\mathbf{z} - \mathbf{g}^{\jmath}) + \boldsymbol{\mu}^{\jmath-1} \quad \text{y} \quad H_f(\mathbf{z}) = \rho\,I_n,$$

donde I_n es la matriz identidad de tamaño $n \times n$. Así que esta vez la Hessiana es constante y definida positiva, y el gradiente se anula si

$$\mathbf{z} = \mathbf{g}^{\jmath} - \boldsymbol{\mu}^{\jmath-1}/\rho \quad \text{y} \quad \mathbf{z} \geq \mathbf{0}. \tag{3.29}$$

Observemos que la ecuación (3.29), para la variable $\mathbf{z}$, se relaciona con la ecuación (3.28) para la variable $\mathbf{y}$.

3.3.2. Relación entre los parámetros de penalización k y ρ

Estos dos parámetros tienen una influencia importante en el rendimiento del algoritmo iterativo 3 y en particular, en los subproblemas (3.23)-(3.25); principalmente en la solución del sistema lineal (3.26) y la minimización de la función cuadrática (3.27). Con respecto al subproblema (3.26), entre más grande sea el valor $k/(1+\rho)$ mayor es el número de condición de la matriz $Q_{k,\rho}$. Para el modelo penalizado (3.3) se obtienen buenos resultados con $k \geq 10^3$ y si se quiere tener un desempeño computacional similar en la solución de (3.26) se establece $k/(1+\rho) \geq 10^3$. Se debe tener en cuenta que el paso 3 es la parte más costosa del algoritmo iterativo 3; sin embargo, tiene la ventaja de que puede ser resuelto con el algoritmo GC.

Finalmente, hay que tener en cuenta que si tomamos $k = \infty$ (o $\alpha = 0$ en (3.8)), se obtiene una matriz simétrica y definida positiva $Q_\rho = \rho\, I + P^T P$; lo cual implica que esta variante del Lagrangiano aumentado tiene un efecto de regularización para el problema original y en este caso, basta con tomar cualquier valor positivo para ρ. Los ejemplos numéricos en la sección 4.3 corroboran esta propiedad.

Resumiendo, el algoritmo iterativo 3 es eficiente ya que el trabajo adicional (que incluye resolver el problema (3.24) y actualizar los multiplicadores en (3.25)) es marginal en comparación con la solución del problema (3.23). Observemos que el cálculo numérico de (3.23) con el algoritmo GC es más rápido que el algoritmo GCM para resolver el problema penalizado (3.4), porque la matriz $Q_{k,\rho}$ tiene un número de condición inferior que el de Q_k para el mismo valor de k. Además, se garantiza que todas las direcciones de descenso generadas con el algoritmo GC son $Q_{k,\rho}$ – conjugadas. Este nuevo enfoque es más robusto porque los coeficientes nulos del vector O-D obsoleto no son forzosamente nulos en el vector O-D estimado.

3.3.3. Ejemplo de aplicación del algoritmo ADMM

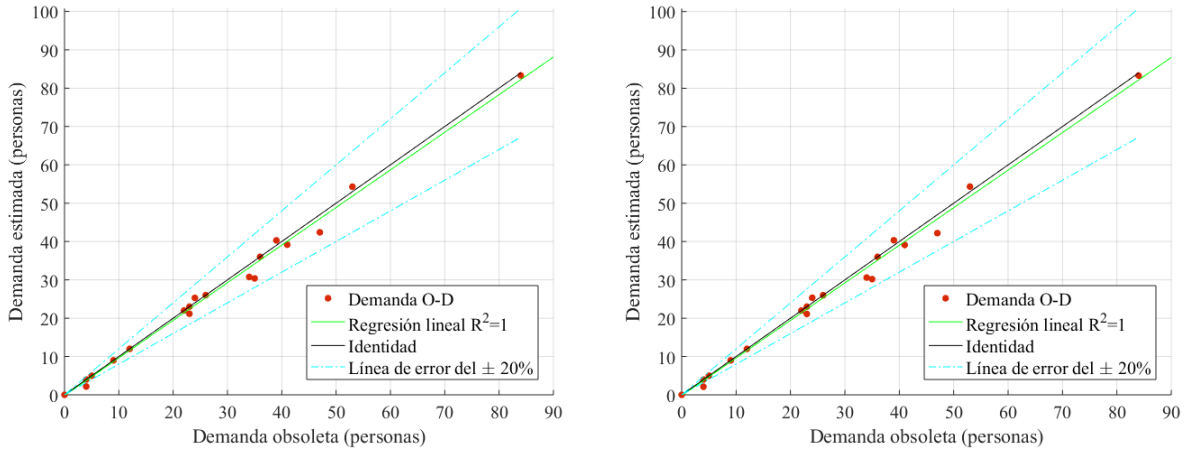
Para fijar ideas, retomemos nuevamente el ejemplo de la sección 3.2.3. Al aplicar el algoritmo 3, tomando $\varepsilon = 0.1$ y diferentes valores para ρ y k se obtienen, para la red ejemplo de la sección 2.3 los resultados mostrados en la tabla 3.3.

ρ	k	$\frac{k}{1+\rho}$	$(J,\ \overline{\ell})$	$RMSE_v$	$\|\|P\mathbf{g}-\hat{\mathbf{v}}\|\|_m$	$RMSE_g$	$\|\|\mathbf{g}-\hat{\mathbf{g}}\|\|_n$
1	2×10^2	10^2	(1,3)	0.03	0.1	1.72	8.6
	2×10^3	10^3	(1,3)	0.00	0.0	1.72	8.6
	2×10^4	10^4	(1,3)	0.00	0.0	1.72	8.6
	2×10^5	10^5	(1,3)	0.00	0.0	1.72	8.6
	∞	–	(1,3)	0.00	0.0	1.72	8.6
19	2×10^2	10	(1,3)	0.30	0.5	1.66	8.3
	2×10^3	10^2	(1,3)	0.03	0.1	1.72	8.6
	2×10^4	10^3	(1,3)	0.00	0.0	1.72	8.6
	2×10^5	10^4	(1,3)	0.00	0.0	1.72	8.6
	∞	–	(1,3)	0.00	0.0	1.72	8.6
199	2×10^2	1	(1,3)	2.22	3.9	1.25	6.2
	2×10^3	10	(1,3)	0.30	0.5	1.66	8.3
	2×10^4	10^2	(1,3)	0.03	0.1	1.72	8.6
	2×10^5	10^3	(1,3)	0.00	0.0	1.72	8.6
	∞	–	(1,3)	0.00	0.0	1.72	8.6
1999	2×10^2	0.1	(1,2)	6.38	11.1	0.36	1.8
	2×10^3	1	(1,3)	2.22	3.9	1.25	6.2
	2×10^4	10	(1,3)	0.30	0.5	1.66	8.3
	2×10^5	10^2	(1,3)	0.03	0.1	1.72	8.6
	∞	–	(1,3)	0.00	0.0	1.72	8.6

Tabla 3.3: Resultados del algoritmo ADMM para la red ejemplo introducida en la sección 2.3.

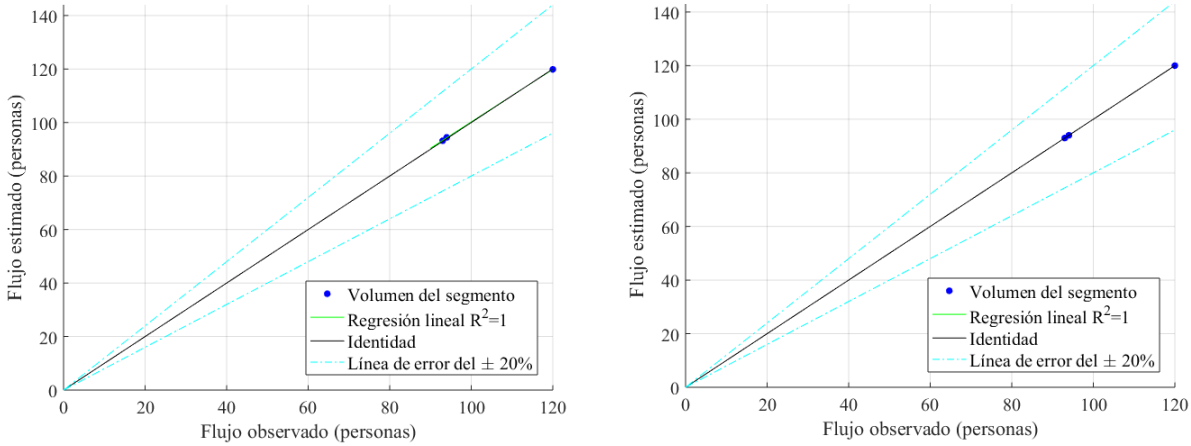
La figura 3.3 muestra la demanda *a priori* (eje x) contra la demanda estimada (eje y) con el algoritmo ADMM para $\rho = 19$ y $k = 200$, parte (a), y $k = 2 \times 10^4$, parte (b). Por otro lado, la figura 3.4, muestra los volúmenes observados (eje x) contra los volúmenes estimados con $\rho = 19$ y $k = 200$ (eje y), parte (a); en la parte (b) de la misma figura se muestran los volúmenes observados contra los volúmenes estimados con $k = 2 \times 10^4$. Nuevamente los puntos que están sobre la identidad representan un ajuste perfecto entre los valores estimados correspondientes y los datos, mientras que los puntos que están entre las dos líneas punteadas corresponden a un ajuste con un error relativo menor al 20 %.

Las líneas verdes representan la regresión lineal, con coeficiente de correlación $R^2 = 1$ (en todas las figuras). Los valores correspondientes para la pendiente m y la ordenada al origen b de la figura 3.3 son $m_g = 0.98$ y $b_g = -0.18$ para (a) y (b); mientras que los parámetros correspondientes en la figura 3.4 son $m_v = 0.98$ y $b_v = 1.88$ para (a) y $m_v = 1.00$ y $b_v = 0.02$ para (b). Nuevamente observamos una correlación fuerte entre los parámetros estimados y los datos, además de obtener un mejor ajuste en los volúmenes para $\rho = 19$ y $k = 2 \times 10^4$.



(a) $\rho = 19$, $k = 200$. (b) $\rho = 19$, $k = 2 \times 10^4$.

Figura 3.3: Demanda *a priori* vs demanda estimada con el algoritmo ADMM para la red ejemplo.



(a) $\rho = 19$, $k = 200$. (b) $\rho = 19$, $k = 2 \times 10^4$.

Figura 3.4: Demanda *a priori* vs demanda estimada con el algoritmo ADMM para la red ejemplo.

Finalmente, la matriz (vector) de demanda O-D actualizada y el vector de volúmenes estimados para $\rho = 19$ y $k = 2 \times 10^4$ nos queda:

$$\mathbf{g} = \begin{pmatrix} 0 & 12 & 2.1 & 21.1 & 82.3 \\ 4 & 0 & 39.1 & 30.2 & 12 \\ 9 & 12 & 0 & 5 & 22 \\ 26 & 40.3 & 23 & 0 & 36 \\ 54.3 & 25.3 & 30.6 & 42.2 & 0 \end{pmatrix}, \qquad \mathbf{v} = \begin{pmatrix} 120 \\ 93 \\ 94 \end{pmatrix}.$$

Observemos que en este caso $\|\hat{\mathbf{g}} - \mathbf{g}\| = 8.6$, por lo cual la demanda de referencia es más cercana a la solución obtenida con el algoritmo ADMM que la solución que se obtuvo con el algoritmo GCM en la sección 3.2.3.

3.4. Reducción del tamaño del problema

En esta sección discutimos cuando es conveniente/posible reducir el tamaño del problema. Cuando se hace alguna reducción automáticamente se obtienen dos beneficios: el ahorro de memoria y una reducción adicional en el costo de cómputo de cada algoritmo.

Una de las propiedades de los algoritmos MDM y GCM es que preservan la estructura de la matriz O-D *a priori* $\hat{\mathbf{g}}$ al calcular la nueva matriz $\mathbf{g}$: cada entrada nula en la primera matriz produce una entrada nula en la actualizada; dado que es muy probable que no todas las entradas nulas evolucionen a un valor positivo, especialmente cuando se actualiza para un periodo corto en una red muy grande. Luego, extrayendo los coeficientes nulos de la matriz O-D *a priori* $\hat{\mathbf{g}}$, obtenemos una matriz O-D *a priori* reducida $\hat{\mathbf{g}}_r$ de tamaño $n_r \leq n$, que tiene solo coeficientes positivos. Así, formulamos el problema (3.2) nuevamente de la siguiente manera:

Dados $\hat{\mathbf{g}}_r \in \mathbb{R}_+^{n_r}$ y $\hat{\mathbf{v}} \in \mathbb{R}_+^m$, encontrar $\mathbf{g}_r \in \mathbb{R}_+^{n_r}$ que minimiza

$$J(\mathbf{g}_r) = \frac{1}{2}\|\mathbf{g}_r - \hat{\mathbf{g}}_r\|_{n_r}^2 \quad \text{sobre el conjunto} \quad \mathcal{V}_r = \left\{\mathbf{g}_r \in \mathbb{R}_+^{n_r} \ : \ P_r\mathbf{g}_r = \hat{\mathbf{v}}\right\}, \tag{3.30}$$

donde la matriz $P_r \in \mathbb{R}^{m\times n_r}$ se obtiene de P al extraer las columnas que corresponden a las entradas nulas de la matriz (vector) O-D *a priori* $\hat{\mathbf{g}}$. Ahora, el modelo penalizado (3.4) y el algoritmo GCM tienen los siguientes beneficios adicionales:

1. El espacio nulo de la matriz P_r tiene una dimensión menor que el espacio nulo de P y comparten los mismos coeficientes positivos. Por lo tanto, hay un ahorro significativo de memoria, especialmente cuando $\hat{\mathbf{g}}$ tiene una gran cantidad de entradas nulas; lo cual es muy común para redes de gran escala.

2. Dado que no hay coeficientes nulos en $\hat{\mathbf{g}}_r$, entonces es más probable que γ_ℓ satisfaga la restricción de no-negatividad en el paso 6 del algoritmo 2.

Análogamente, con esta reducción, el problema de Lagrangiano aumentado es

$$\mathcal{L}_{k,\rho}(\mathbf{g}_r, \mathbf{y}_r, \boldsymbol{\mu}_r) = J_k(\mathbf{g}_r) - \boldsymbol{\mu}_r^T(\mathbf{g}_r - \mathbf{y}_r \odot \mathbf{y}_r) + \frac{\rho}{2}\|\mathbf{g}_r - \mathbf{y}_r \odot \mathbf{y}_r\|_n^2,$$

donde $\mathbf{y}_r, \boldsymbol{\mu}_r \in \mathbb{R}^{n_r}$. Por lo tanto las condiciones de KKT nos llevan a

$$\begin{aligned} ((1+\rho)I_{n_r} + kP_r^TP_r)\mathbf{g}_r &= \hat{\mathbf{g}}_r + kP_r^T\hat{\mathbf{v}} + \boldsymbol{\mu}_r + \rho\,\mathbf{y}_r \odot \mathbf{y}_r, \\ 2\left[\boldsymbol{\mu}_r + \rho(\mathbf{y}_r \odot \mathbf{y}_r - \mathbf{g}_r)\right] \odot \mathbf{y}_r &= \mathbf{0}, \\ \mathbf{g}_r - \mathbf{y}_r \odot \mathbf{y}_r &= \mathbf{0}, \end{aligned}$$

donde I_{n_r} es la matriz identidad de tamaño $n_r \times n_r$. El algoritmo ADMM, para este problema reducido, se aplica igual que para el problema completo.

3.4.1. Ejemplo de la reducción del problema

Retomando nuevamente el ejemplo de la sección 3.2.3 y haciendo las reducciones descritas en la sección anterior se tiene que

$$\hat{\mathbf{g}}_r^T = (12, 4, 23, 84, 4, 41, 35, 12, 9, 12, 5, 22, 26, 39, 23, 36, 53, 24, 34, 47)\,.$$
$$\bar{\mathbf{g}}_r^T = (10, 5, 20, 76, 5, 40, 30, 10, 10, 10, 5, 20, 30, 40, 20, 30, 50, 30, 34, 40)\,.$$
$$P_r = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 7/19 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12/17 & 1 \end{pmatrix}.$$

Al aplicar el algoritmo GCM con estos datos, se obtienen exactamente los mismos resultados que los que se mostraron en la sección 3.2.3. De igual manera, al aplicar el algoritmo ADMM se obtienen exactamente los mismos resultados que los mostrados en la sección 3.3.3. Hasta aquí, no se aprecian las ventajas de reducir el problema, puesto que el ejemplo con el que estamos trabajando es muy pequeño y el costo de cómputo para resolverlo es muy poco (menor a una décima de segundo). Sin embargo, en el capítulo 4 estas ventajas se aprecian mejor.

Otra forma de reducir el tamaño del problema, es observando en las columnas de la matriz P_r que son completamente de ceros; así, si la i-ésima columna de P_r es cero, se elimina y se hace la demanda correspondiente $g_i = \hat{g}_i$; esto es:

$$\hat{\mathbf{g}}_r^T = (12, 4, 23, 84, 4, 41, 35, 12, 9, 12, 5, 22, 26, 39, 23, 36, 53, 24, 34, 47)\,.$$
$$\mathbf{g}_r^T = (12, g_2, g_3, g_4, 4, g_6, g_7, 12, 9, 12, 5, 22, 26, g_{14}, 23, 36, g_{17}, g_{18}, g_{19}, g_{20})\,.$$
$$P_{rr} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 7/19 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 12/17 & 1 \end{pmatrix}.$$
$$\hat{\mathbf{g}}_{rr}^T = (4, 23, 84, 41, 35, 39, 53, 24, 34, 47)\,.$$
$$\mathbf{g}_{rr}^T = (g_2, g_3, g_4, g_6, g_7, g_{14}, g_{17}, g_{18}, g_{19}, g_{20})\,.$$

De esta forma, de tener 25 incógnitas inicialmente, nos queda resolver un problema que tiene sólo 10 incógnitas.

3.5. Extensión del modelo

Hasta ahora, hemos obtenido los mejores resultados con el enfoque de Lagrangiano aumentado, logrando un buen ajuste de los volúmenes manteniendo la demanda estimada cerca de la demanda de referencia. Estos resultados, se pueden mejorar aun más para obtener aproximaciones de demanda más realistas añadiendo más información, en caso de que se tenga disponible. Por ejemplo, las producciones y atracciones totales en cada una de las zonas agregadas de tránsito

(nodos centroides). En este caso, se añaden al modelo las siguientes restricciones:

$$\sum_{q \in Q} g_{pq} = O_p, \qquad \forall p \in \mathcal{P}, \tag{3.31}$$

$$\sum_{p \in P} g_{pq} = D_q, \qquad \forall q \in \mathcal{Q}, \tag{3.32}$$

donde O_p es la demanda total que se produce en la zona $p \in \mathcal{P}$ y D_q es la demanda total que tiene como destino la zona $q \in \mathcal{Q}$. Estas restricciones se pueden representar como el producto de una matriz por un vector, digamos $A\mathbf{g} = \mathbf{O}$ y $B\mathbf{g} = \mathbf{D}$. Penalizando la diferencia de estas dos cantidades y añadiéndolas al Lagrangiano aumentado (3.19) se obtiene el nuevo Lagrangiano:

$$\mathcal{L}_{k_1,k_2,k_3,\rho}(\mathbf{g}, \mathbf{y}, \boldsymbol{\mu}) = \mathcal{L}_{k_1,\rho}(\mathbf{g}, \mathbf{y}, \boldsymbol{\mu}) + \frac{k_2}{2}\|A\mathbf{g} - \mathbf{O}\|^2 + \frac{k_3}{2}\|B\mathbf{g} - \mathbf{D}\|^2. \tag{3.33}$$

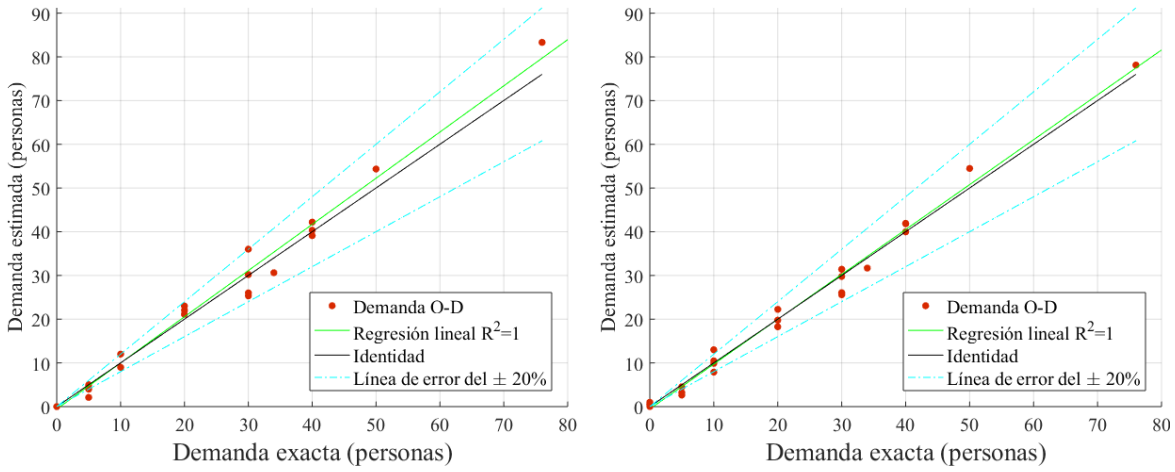
3.5.1. Ejemplo con producciones y atracciones

Para el ejemplo de la sección 3.2.3, al resolver el modelo 3.33 con $k_1 = k_2 = k_3$ se obtienen los resultados que se muestran en la tabla 3.4. Comparando estos resultados con los que se mostraron el la tabla 3.3 par el modelo completo, se puede ver que para valores de $k/(1+\rho)$ mayores o iguales que 10^3 se ajustan igual de bien los volúmenes para ambos modelos; sin embargo, resolver el modelo (3.33) resulta más costoso en cuanto al número de iteraciones. Además, se observa que el resultado con esta nueva aproximación se aleja más de la demanda de referencia.

ρ	k	$\frac{k}{1+\rho}$	$(J,\ \bar{\ell})$	$RMSE_v$	$\|\|P\mathbf{g}-\hat{\mathbf{v}}\|\|_m$	$RMSE_g$	$\|\|\mathbf{g}-\hat{\mathbf{g}}\|\|_n$
1	2×10^2	10^2	(8,8)	0.01	0.0	2.59	13.0
	2×10^3	10^3	(5,7)	0.00	0.0	2.67	13.4
	2×10^4	10^4	(5,7)	0.00	0.0	2.67	13.4
	2×10^5	10^5	(5,7)	0.00	0.0	2.67	13.4
	∞	–	(5,7)	0.00	0.0	2.67	13.4
19	2×10^2	10	(5,7)	0.04	0.1	2.63	13.2
	2×10^3	10^2	(5,7)	0.00	0.0	2.67	13.4
	2×10^4	10^3	(5,7)	0.00	0.0	2.67	13.4
	2×10^5	10^4	(5,7)	0.00	0.0	2.67	13.4
	∞	–	(5,7)	0.00	0.0	2.67	13.4
199	2×10^2	1	(5,6)	0.13	0.2	2.63	13.1
	2×10^3	10	(5,7)	0.02	0.0	2.66	13.3
	2×10^4	10^2	(5,7)	0.00	0.0	2.67	13.4
	2×10^5	10^3	(5,7)	0.00	0.0	2.67	13.4
	∞	–	(5,7)	0.00	0.0	2.67	13.4
1999	2×10^2	0.1	(6,4)	1.16	2.0	2.34	11.7
	2×10^3	1	(5,5)	0.12	0.2	2.63	13.2
	2×10^4	10	(5,7)	0.02	0.0	2.67	13.3
	2×10^5	10^2	(5,7)	0.00	0.0	2.67	13.4
	∞	–	(5,7)	0.00	0.0	2.67	13.4

Tabla 3.4: Resultados del algoritmo ADMM para la red ejemplo.

La dispersión entre la demanda estimada y la demanda "exacta", se se muestra en la figura 3.5, donde se puede observar una mejor aproximación a la demanda exacta cuando se incorpora más información al modelo.



(a) Modelo (3.19). (b) Modelo (3.33).

Figura 3.5: Dispersión para la demanda estimada con $k/(\rho+1) = 10^3$ contra la demanda O-D exacta $\bar{\mathbf{g}}$. Red ejemplo.

Hasta aquí, hemos probado nuestros modelos en una red ficticia pequeña; sin embargo en la realidad, las redes de tránsito suelen ser más grandes. Los algoritmos de solución que hemos usado hasta este momento, tienen la característica de proporcionar buenas soluciones para problemas de gran tamaño, como se muestra en el siguiente capítulo, en donde mostramos los resultados que obtuvimos al aplicar la metodología descrita en este trabajo en dos redes de tránsito reales.

Capítulo 4

Resultados numéricos

En este capítulo se muestran y comparan los resultados numéricos al aplicar cada una de las metodologías descritas. Los algoritmos que presentamos, se implementaron en una computadora HP-Pavilion dm4 que cuenta con un procesador Intel(R) Core(TM) i5 y 8 GB de memoria RAM.

En la sección 4.1, se describen las características las dos redes de tránsito (Winnipeg y Zona Metropolitana del Valle de México) en donde se aplicó cada metodología. En la sección 4.2, se muestran y comparan los resultados al usar el modelo penalizado y resolver con los algoritmos multiplicativos MDM y GCM; además, se muestra numéricamente la convergencia cuando el parámetro de penalización k tiende a infinito. En la sección 4.3, se muestra el desempeño del método ADMM y su convergencia numérica cuando la relación entre los parámetros de penalización $k/1+\rho$ tiende a infinito. En ambas secciones 4.2 y 4.3, se consideran tanto el problema completo como el problema reducido y se comparan las soluciones obtenidas.

4.1. Casos de estudio

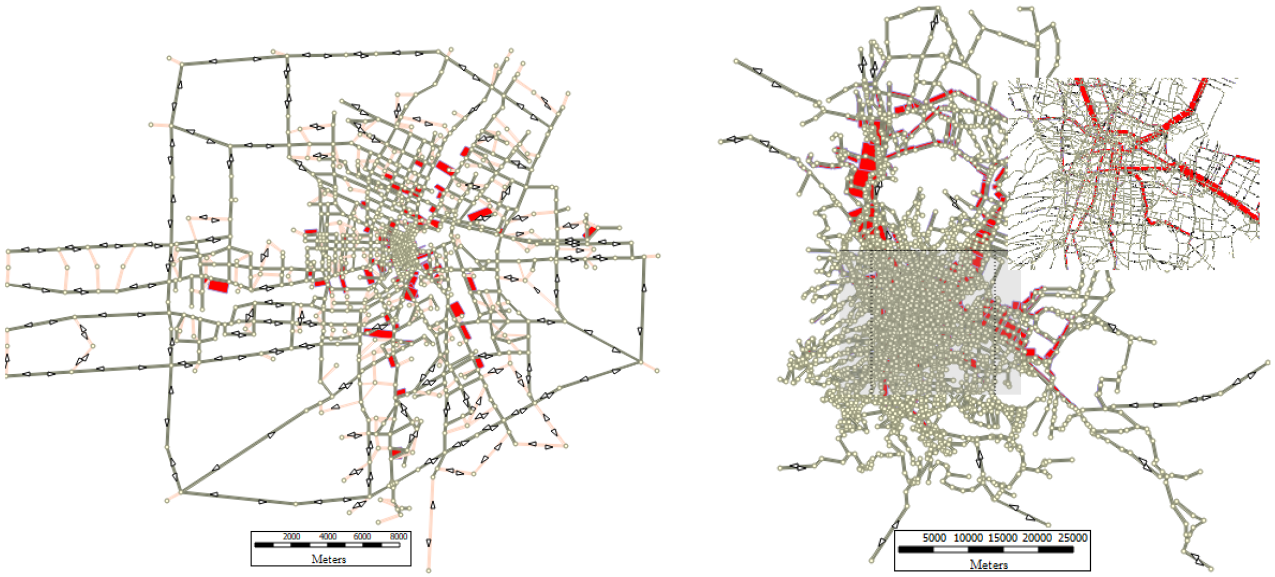
Para los experimentos numéricos, consideramos datos sintéticos (volúmenes de flujo de segmento) de dos redes de tránsito; la red de Winnipeg, Canadá, incluida en la base de datos de demostración de EMME (INRO, 2018), y la red con base en la Zona Metropolitana de el Valle de México (ZMVM), México, proporcionada por Torres (2013).

En cada una de las redes de estudio se construyó el siguiente escenario: a partir de una matriz de demanda O-D, existente en cada una de las bases de datos $\bar{\mathbf{g}}$ (la cual puede representar la demanda de pasajeros en la hora pico de la mañana), realizamos una asignación lineal de tránsito y extrajimos el flujo de pasajeros en algunos segmentos de la red, los cuales juegan el papel de los volúmenes observados $\hat{\mathbf{v}}$. Posteriormente, generamos una matriz de demanda O-D "*a priori*" a partir de la matriz $\bar{\mathbf{g}}$ con una perturbación uniforme al rededor del 20 %, obteniendo una distancia relativa $\|\hat{\mathbf{g}}-\bar{\mathbf{g}}\|/\|\hat{\mathbf{g}}\| = 0.13$, para ambas redes. Luego, con esta información, aplicamos los métodos descritos en este trabajo para evaluar y comparar sus desempeños. La tabla 4.1 muestra las características generales de las dos redes de estudio; donde un nodo regular representa la intersección de dos o más arcos y se consideran todos los modos de tránsito disponibles en cada red (autobús, metro, metrobús, tren ligero, etc.). En este contexto, los arcos

representan las calles, mientras que los segmentos representan las rutas de las líneas de tránsito; esto significa que para cada arco se puede tener más de un segmento de tránsito (o ninguno). La figura 4.1 muestra los segmentos con conteos de flujo, resaltados en rojo para redes de Winnipeg y de la ZMVM, respectivamente.

Tabla 4.1: Características de la red de Winnipeg y de la red de la ZMVM.

Atributo	Winnipeg	ZMVM
Zonas	154	1705
paires O-D	23,716	2,907,025
No. de pares con $\hat{g}_{pq} > 0$	5,394 (22.7 %)	20,278 (0.7 %)
Nodos regulares	906	7241
Arcos	3,005	31,720
Modos	5	18
Tipos de vehículos	4	11
Líneas de tránsito	133	845
Segmentos de tránsito	4,347	46,981
Segmentos con conteos	136 (3.1 %)	1,470 (3.1 %)



(a) Red de tránsito de Winnipeg.

(b) Red de tránsito con base en la Zona Metropolitana del Valle de México (ZMVM).

Figura 4.1: Segmentos con conteos disponibles en rojo para cada caso de estudio.

4.2. Desempeño de los algoritmos MDM y GCM

En esta sección, comparamos el desempeño del algoritmo de máximo descenso multiplicativo (MDM) propuesto por Spiess (parte del software de planificación de transporte EMME/4 (INRO, 2018)) con nuestro algoritmo GCM descrito en la sección 3.2.2. Las tablas 4.2 y 4.3

muestran los resultados numéricos que se obtuvieron con el modelo penalizado (3.3) para la red de tránsito de Winnipeg y la red de tránsito de la ZMVM, respectivamente. En ambas tablas, k es el parámetro de penalización, CPU es el tiempo de cómputo en segundos, Iters. es el número de iteraciones que requirió cada método para alcanzar la convergencia con una tolerancia dada (establecida como $\varepsilon = 10^{-3}$ para estos experimentos), $RMSE_v$ es la raíz del error cuadrático medio para el ajuste de los volúmenes en los segmentos, $\|P\mathbf{g} - \hat{\mathbf{v}}\|_m$ es la distancia entre los volúmenes de segmento estimados y los flujos observados, $RMSE_g$ es la raíz del error cuadrático medio para la demanda O-D y $\|\mathbf{g} - \hat{\mathbf{g}}\|_n$ es la distancia entre la matriz de demanda estimada y la demanda *a priori*. En el apéndice A, se muestra cómo se calcula el $RMSE$ para un conjunto de datos.

El caso cuando $k = \infty$ ($\alpha = 0$ en (3.8)) en ambas tablas, deberá considerarse como el caso cuando la función objetivo es $(1/2)\|P\mathbf{g} - \hat{\mathbf{v}}\|_m^2$ (el modelo de Spiess). Estos resultados los obtuvimos con nuestro propio código en Matlab y muestran lo que esperábamos; cuando el parámetro de penalización se aproxima a infinito, los resultados convergen al modelo de Spiess. De echo, para valores de $k > 100$ obtenemos casi los mismos resultados, incluyendo el número de iteraciones y el tiempo de cómputo, y prácticamente no hay cambios cuando $k \geq 1000$. Esto podría explicar por qué los resultados numéricos que obtuvieron algunos autores (como Noriega y Florian, 2009; Verbas et al., 2011) son mejores, en sus respectivos casos, para valores de β en el intervalo $(0.999, 1)$. Por lo tanto, un valor de $k = 100$ o mayor, es suficiente en la práctica para ambos casos de estudio.

La diferencia más notable es el desempeño (número de iteraciones y tiempo de cómputo) del algoritmo GCM con respecto al algoritmo MDM. Para la red de tránsito de Winnipeg, el $RMSE_v$ inicial se reduce 262 (180) veces con el algoritmo GCM (MDM) y el tiempo de cómputo mejora cerca de tres veces con GCM respecto a MDM. Para el caso de la red de tránsito de la ZMVM, el $RMSE_v$ inicial se reduce 111 (73) veces con el algoritmo GCM (MDM), mientras que el tiempo de cómputo se reduce cerca de 4.3 veces con GCM respecto a MDM. Por lo tanto, el algoritmo GCM no sólo es más rápido, sino que reduce más el $RMSE_v$ inicial en ambas redes, lo cual se traduce en un mejor ajuste de los volúmenes observados.

Tabla 4.2: Comparación de los algoritmos MDM y GCM para la red de Winnipeg.

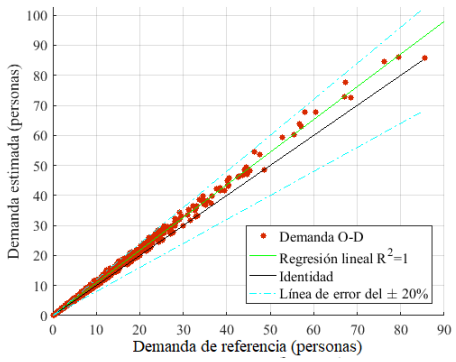
k	Método	CPU	*Iters.*	$RMSE_v$	$\|\|P\mathbf{g} - \hat{\mathbf{v}}\|\|_m$	$RMSE_g$	$\|\|\mathbf{g} - \hat{\mathbf{g}}\|\|_n$
Inicial	$(l = 0)$			28.79	335.7		
100	MDM	1.1 s.	80	0.16	1.9	0.33	50.3
	GCM	0.3 s.	21	0.11	1.3	0.33	50.2
1000	MDM	0.8 s.	78	0.17	1.9	0.33	50.9
	GCM	0.3 s.	21	0.11	1.3	0.33	51.0
10000	MDM	0.9 s.	78	0.17	1.9	0.33	50.9
	GCM	0.3 s.	21	0.11	1.3	0.33	51.1
∞	MDM	1.0 s.	78	0.16	1.9	0.33	50.9
	GCM	0.3 s.	21	0.11	1.3	0.33	51.1

Tabla 4.3: Comparación de los algoritmos MDM y GCM para la red de la ZMVM.

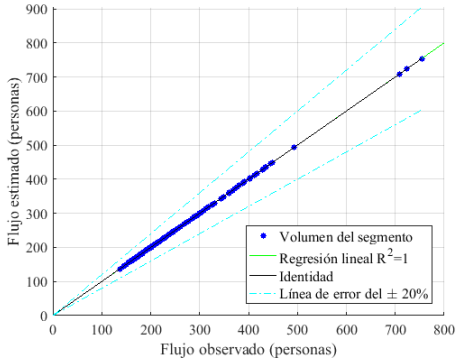
k	Método	CPU	*Iters.*	$RMSE_v$	$\|\|P\mathbf{g} - \hat{\mathbf{v}}\|\|_m$	$RMSE_g$	$\|\|\mathbf{g} - \hat{\mathbf{g}}\|\|_n$
Inicial	$(l = 0)$			5797.04	222262.0		
100	MDM	23.5 s.	147	78.94	3026.8	2.54	4323.0
	GCM	5.4 s.	30	52.34	2006.8	2.64	4495.7
1000	MDM	23.6 s.	147	78.92	3025.9	2.54	4324.9
	GCM	5.6 s.	30	52.34	2006.7	2.64	4499.3
10000	MDM	23.6 s.	147	78.92	3025.9	2.54	4325.1
	GCM	5.5 s.	30	52.34	2006.6	2.64	4499.7
∞	MDM	24.0 s.	147	78.92	3025.8	2.54	4325.1
	GCM	5.7 s.	30	52.34	2006.6	2.64	4499.7

La figura 4.2 muestra la demanda *a priori* (eje x) contra la demanda estimada (eje y), parte (a), y los volúmenes observados (eje x) contra los estimados (eje y), parte (b), para la red de tránsito de Winnipeg. Todos los puntos en la recta de identidad, indican un ajuste perfecto entre los valores estimados correspondientes y los datos; mientras que los puntos que se encuentran entre las dos líneas punteadas, tienen un error relativo menor al 20 %. Las líneas de regresión representadas en verde, con coeficiente de correlación $R^2 = 1$ (para ambas figuras) y respectivas pendiente m y ordenada al origen b, son: $m_g = 1.09$ y $b_g = 0.00$ para la demanda y $m_v = 1.00$ y $b_v = -0.02$ para los flujos en los segmentos. Estos resultados indican que los valores estimados están altamente correlacionados con los datos.

Análogamente, la figura 4.3 muestra los diagramas de dispersión correspondientes para la red con base en la ZMVM, con pendiente $m_g = 1.06$ y ordenada al origen $b_g = 0$ para la demanda, pendiente $m_v = 1.00$ y ordenada al origen $b_v = -0.39$ para los flujos en los segmentos. Estas dos figuras muestran que la mayoría de los puntos caen dentro del área delimitada por las líneas de error $\pm 20\,\%$, lo que indica que el modelo propuesto brinda resultados precisos. Observemos que el ajuste en los volúmenes de segmento es casi perfecta, mientras que la demanda estimada no es necesariamente cercana a la *a priori* en la misma proporción. En el apéndice A, se muestra cómo calcular los parámetros m y b de la recta de ajuste, así como el cálculo de R^2.

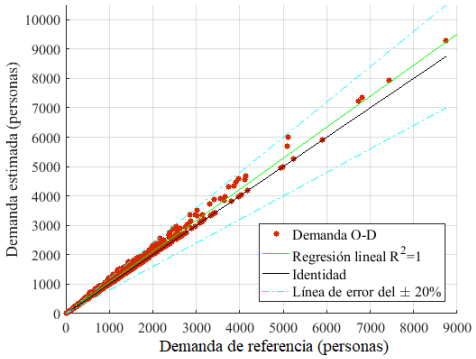


(a) Resultados para la demanda estimada en cada par O-D.

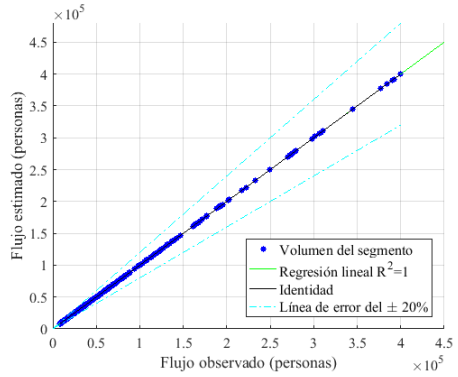


(b) Resultados para los volúmenes estimados en cada segmento con conteos disponibles.

Figura 4.2: Diagramas de dispersión de las estimaciones obtenidas con $k = 1000$ y GCM en la red de tránsito de Winnipeg.



(a) Resultados para la demanda estimada en cada par O-D.



(b) Resultados para los volúmenes estimados en cada segmento con conteos disponibles.

Figura 4.3: Diagramas de dispersión de las estimaciones obtenidas con $k = 1000$ y GCM en la red de tránsito de la ZMVM.

Considerando la reducción del problema, discutido en la sección 3.4, probamos los algoritmos MSD y MCG obteniendo los resultados que se muestran en las tablas 4.4 y 4.5 para cada red. Aquí, también presentamos el error porcentual medio MPE (ver apéndice A) para la matriz de demanda; esta métrica generalmente se utiliza para medir un posible sesgo en la estimación. Comparando estos resultados con los que se muestran en tablas 4.2 y 4.3, la diferencia más significativa es el tiempo de CPU, que se reduce unas tres veces para Winnipeg y unas seis veces para la ZMVM. Dado que el número de iteraciones en el problema reducido es similar al número de iteraciones en el problema completo, la mejora en el tiempo de CPU está esencialmente relacionada con el ahorro de memoria y la comunicación entre procesos, ya que el 77.3 % de los coeficientes en la matriz O-D de Winnipeg son nulos, mientras que en la matriz de la ZMVM lo son el 99.3 %. No incluimos las gráficas de dispersión correspondientes ya que la calidad de la solución es similar a las que se muestran en las figuras 4.2 y 4.3.

Tabla 4.4: Comparación de los algoritmos MDM y GCM en la red de Winnipeg, problema reducido.

k	Método	CPU	*Iters.*	$RMSE_v$	$\|\|P_r\mathbf{g}_r - \hat{\mathbf{v}}\|\|_m$	$RMSE_g$	$\|\|\mathbf{g}_r - \hat{\mathbf{g}}_r\|\|_n$	MPE
100	MDM	0.3 s.	74	0.18	2.1	0.33	50.3	-8.23
	GCM	0.1 s.	21	0.11	1.3	0.33	50.2	-8.30
1000	MDM	0.3 s.	72	0.18	2.1	0.33	50.9	-8.16
	GCM	0.1 s.	21	0.11	1.3	0.33	51.0	-8.20
10000	MDM	0.3 s.	70	0.18	2.2	0.33	50.9	-8.15
	GCM	0.1 s.	21	0.11	1.3	0.33	51.1	-8.19
∞	MDM	0.3 s.	70	0.18	2.1	0.33	50.9	-8.15
	GCM	0.1 s.	21	0.11	1.3	0.33	51.1	-8.19

Tabla 4.5: Comparación de los algoritmos MDM y GCM en la red de la ZMVM, problema reducido.

k	Método	CPU	*Iters.*	$RMSE_v$	$\|\|P_r\mathbf{g}_r - \hat{\mathbf{v}}\|\|_m$	$RMSE_g$	$\|\|\mathbf{g}_r - \hat{\mathbf{g}}_r\|\|_n$	MPE
100	MDM	5.7 s.	147	78.94	3026.8	2.54	4323.0	-5.24
	GCM	1.2 s.	30	52.34	2006.8	2.64	4495.7	-5.40
1000	MDM	5.6 s.	147	78.92	3025.9	2.54	4324.9	-5.24
	GCM	1.2 s.	30	52.34	2006.7	2.64	4499.3	-5.40
10000	MDM	5.6 s.	147	78.92	3025.9	2.54	4325.1	-5.24
	GCM	1.2 s.	30	52.34	2006.6	2.64	4499.7	-5.40
∞	MDM	5.6 s.	147	78.92	3025.8	2.54	4325.1	-5.24
	GCM	1.2 s.	30	52.34	2006.6	2.64	4499.7	-5.40

De acuerdo con los experimentos numéricos que se muestran en esta sección, como máximo tres direcciones consecutivas son Q_k-conjugadas; sin embargo, más direcciones consecutivas son linealmente independientes, y sabemos que cuantas más direcciones son linealmente independientes, más grande es el subespacio donde la función de costo se minimiza, por lo tanto, el mínimo se alcanza más rápido para una tolerancia dada. Por ejemplo, la intuición sobre el método de máximo descenso es que tiene una tasa baja de convergencia, porque se mueve en pasos ortogonales y se produce un fenómeno de "zig-zag", especialmente con problemas mal condicionados. En algoritmo 2, las direcciones de descenso no se mueven en pasos ortogonales y de acuerdo con la fórmula para calcular el paso 3.11, la nueva dirección de descenso multiplicativa es una combinación del nuevo gradiente multiplicativo y la dirección de descenso multiplicativa anterior. Sin embargo, nuestros experimentos numéricos indican que el modelo penalizado combinado con el algoritmo GCM, no solo reproduce los resultados obtenidos con el modelo de Spiess combinado con el algoritmo MDM, sino que también mejora el tiempo de cómputo para el problema completo y el problema reducido.

Calculando el MPE; en las tablas 4.4 y 4.5, así como en las figuras 4.2 (a) y 4.3 (a), se puede ver que los algoritmos multiplicativos tienden a sobrestimar la demanda O-D con respecto a la matriz inicial. Dada la naturaleza multiplicativa del algoritmo, estos aumentos son más evidentes en las zonas con mayor demanda.

4.3. Desempeño del algoritmo ADMM

En esta sección mostramos los resultados obtenidos al aplicar la metodología descrita en la sección 3.3 considerando el problema completo y su reducción (donde se extraen los coeficientes nulos de las matrices O-D) discutidos en la sección 3.4, para las dos redes de estudio.

La tabla 4.6 muestra los resultados numéricos obtenidos con el modelo de Lagrangiano aumentado y el algoritmo 3 (ADMM) para la red de tránsito de Winnipeg. Las primeras tres columnas de la tabla incluyen los valores de los parámetros k, ρ y $k/(1+\rho)$, comunes para ambos problemas. Desde la cuarta columna hasta la séptima, se muestran juntos los resultados numéricos para el problema completo (izquierda) y el problema reducido (derecha), separados por un guión en el centro, la octava columna muestra el MPE para el problema reducido. En esta tabla J denota el número de iteraciones realizadas por el algoritmo ADMM para alcanzar la convergencia hasta la tolerancia deseada ($\varepsilon = 10^{-3}$), mientras que $\overline{\ell}$ es el promedio de iteraciones del algoritmo de GC para encontrar el mínimo en el paso 3 del algoritmo 3; por lo que el número total de iteraciones para resolver el problema es $J \times \overline{\ell}$.

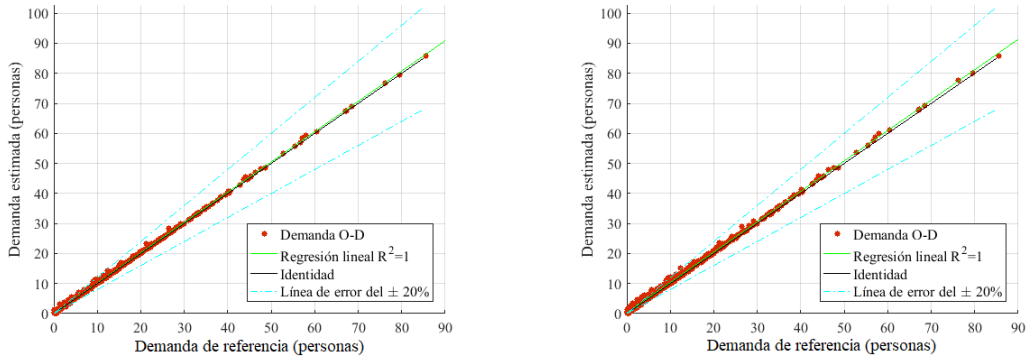
En todos los casos, se obtuvo una raíz del error cuadrático medio en la demanda de $RMSE_g = 0.12$ en el problema completo y $RMSE_g = 0.18$ para el problema reducido. Además; se llegó a que la norma de la distancia entre la matriz obsoleta y la matriz estimada es de $||\mathbf{g} - \hat{\mathbf{g}}||_n = 19.2$ para el problema completo y 27.8 para el problema reducido.

Tabla 4.6: Resultados para la red de Winnipeg, obtenidos con el modelo de Lagrangiano aumentado y el algoritmo ADMM. Desde la cuarta columna hasta la novena, los resultados del problema completo (reducido) se muestran a la izquierda (derecha) del guión.

ρ	k	$\frac{k}{1+\rho}$	CPU (s)	$(J,\ \bar{\ell})$	$RMSE_v$	$\|\|P\mathbf{g}-\hat{\mathbf{v}}\|\|_m$	MPE
1	2×10^2	10^2	2.5 - 0.6	(7,28) - (3,55)	0.09 - 0.01	1.1 - 0.1	-40.83
	2×10^3	10^3	2.3 - 0.4	(7,28) - (3,53)	0.09 - 0.01	1.1 - 0.1	-40.84
	2×10^4	10^4	2.2 - 0.4	(7,28) - (3,53)	0.09 - 0.01	1.1 - 0.1	-40.84
	∞	–	2.1 - 0.4	(7,29) - (3,52)	0.09 - 0.01	1.1 - 0.1	-40.84
19	2×10^2	10	2.3 - 0.3	(7,28) - (2,44)	0.09 - 0.09	1.1 - 1.1	-40.98
	2×10^3	10^2	2.3 - 0.5	(7,28) - (3,53)	0.09 - 0.01	1.1 - 0.1	-40.82
	2×10^4	10^3	2.2 - 0.4	(7,28) - (3,53)	0.09 - 0.01	1.1 - 0.1	-40.84
	∞	–	2.2 - 0.4	(7,28) - (3,53)	0.09 - 0.01	1.1 - 0.1	-40.84
199	2×10^3	10	2.2 - 0.4	(7,29) - (2,44)	0.09 - 0.09	1.1 - 1.1	-40.99
	2×10^4	10^2	2.1 - 0.4	(7,28) - (3,52)	0.09 - 0.01	1.1 - 0.1	-40.82
	2×10^5	10^3	2.2 - 0.4	(7,28) - (3,51)	0.09 - 0.01	1.1 - 0.1	-40.84
	∞	–	2.1 - 0.4	(7,28) - (3,51)	0.09 - 0.01	1.1 - 0.1	-40.84
1999	2×10^3	1	2.4 - 0.1	(7,30) - (1,17)	0.10 - 0.58	1.2 - 6.8	-41.57
	2×10^4	10	2.2 - 0.2	(7,29) - (2,44)	0.09 - 0.09	1.1 - 1.1	-40.99
	2×10^5	10^2	2.2 - 0.4	(7,28) - (3,52)	0.09 - 0.01	1.1 - 0.1	-40.82
	∞	–	2.1 - 0.4	(7,28) - (3,51)	0.09 - 0.01	1.1 - 0.1	-40.84

El comportamiento cualitativo del algoritmo ADMM con respecto a la dispersión entre las matrices O-D para el problema completo y el problema reducido se muestra en la figura 4.4. La pendiente y la ordenada al origen de las líneas de regresión correspondientes que se muestran en esta figura son: $m_g = 1.01$, $b_g = 0.05$ y $m_{g_r} = 1.01$, $b_{g_r} = 0.21$, para el problema completo y el reducido, respectivamente. Estos resultados muestran una mejora significativa para la estimación de la demanda con respecto a los obtenidos anteriormente con el modelo penalizado y el algoritmo MCG (ver tablas 4.2, 4.4 y la figura 4.2). La figura 4.5 muestra el ajuste en los flujos de los segmentos; como podemos ver, se parecen mucho a los resultados obtenidos con el modelo penalizado.

Regresando a la tabla 4.6, observamos que la solución no cambia para valores de $k/(\rho + 1) > 10^2$, independientemente de los valores de ρ y k. Además, la solución del problema reducido se calcula aproximadamente cinco o seis veces más rápido que la solución del problema completo; principalmente, debido al ahorro de memoria. Según las columnas 6-9 de esta tabla, los segmentos son más precisos para el problema reducido, mientras tanto, el ajuste de la demanda es ligeramente mejor para el problema completo. En la figura 4.6, mostramos la evolución de $\|\mathbf{g}^\jmath - \hat{\mathbf{g}}\|_n$ y $\|P\mathbf{g}^\jmath - \hat{\mathbf{v}}\|_m$ a lo largo de las iteraciones $\jmath$ del algoritmo ADMM. Para completar, en la figura 4.7 mostramos la diferencia punto por punto entre la solución del modelo completo y reducido, denotada por $\mathbf{g}_c - \mathbf{g}_r$, para valores de $\rho = 19$ y $k = 20000$.



(a) Modelo completo (b) Modelo reducido

Figura 4.4: Dispersión de la matriz de demanda O-D actualizada obtenida con $k/(\rho+1) = 10^3$. Red de tránsito de Winnipeg.

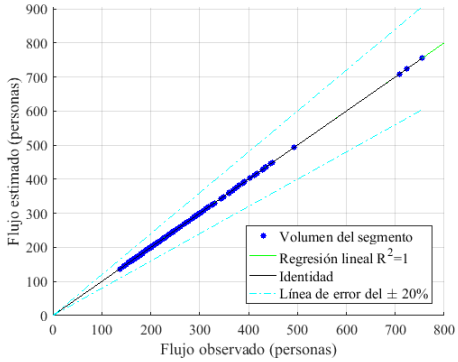
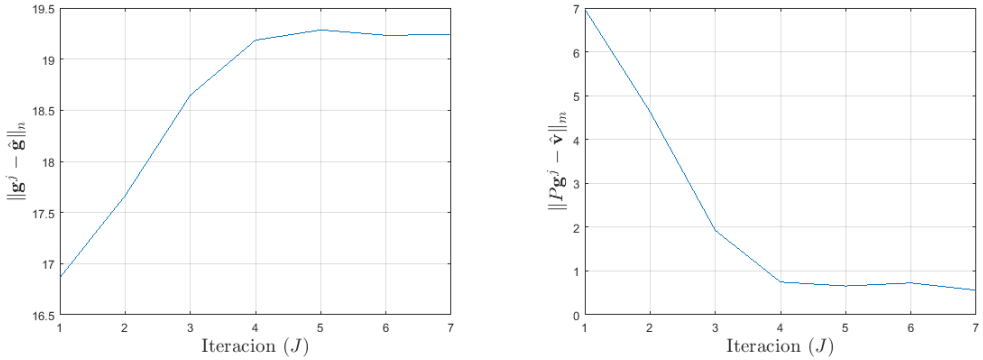


Figura 4.5: Dispersión del flujo en los segmentos obtenida con $k/(\rho+1) = 10^3$. Red de tránsito de Winnipeg.



(a) Evolución de $\|\mathbf{g}^J - \hat{\mathbf{g}}\|_n$. (b) Evolución de $\|P\mathbf{g}^J - \hat{\mathbf{v}}\|_m$.

Figura 4.6: Evolución de las distancias entre los datos y los valores estimados para $j = 1, ..., J$, red de Winnipeg

El último conjunto de experimentos numéricos se realizó para la red con base en la ZMVM.

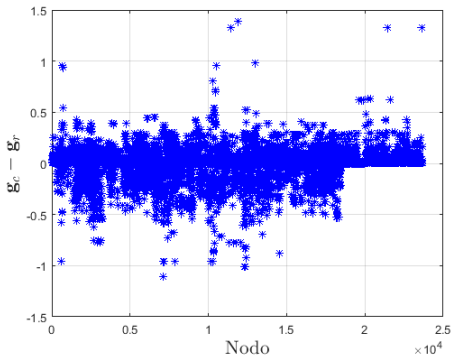


Figura 4.7: Diferencia entre la solución del modelo completo y el modelo reducido para la red de Winnipeg.

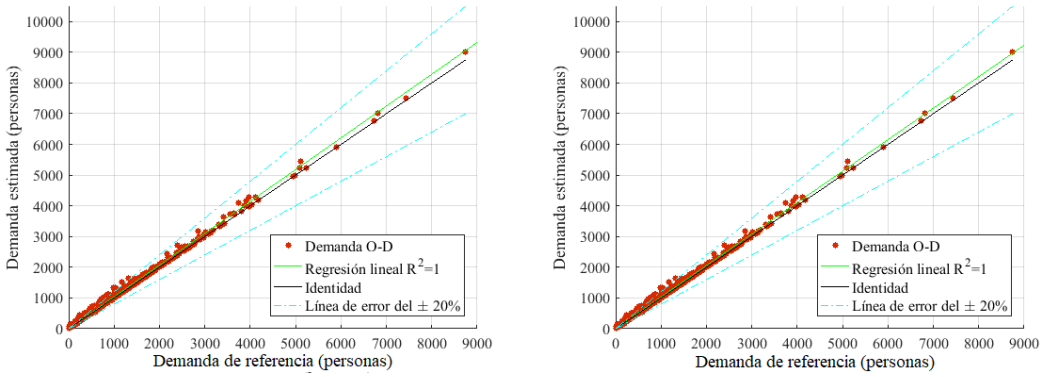
La tabla 4.7 resume los resultados obtenidos con el modelo de Lagrangiano aumentado y el algoritmo ADMM. Una vez más, incluimos juntos los resultados del problema completo y del problema reducido siguiendo el mismo formato que en la tabla 4.6.

Tabla 4.7: Resultados para la red de la ZMVM obtenidos con el modelo de Lagrangiano aumentado y el algoritmo ADMM. De la cuarta columna a la novena, los resultados del problema completo (reducido) se muestran a la izquierda (derecha) del guión.

ρ	k	$\frac{k}{(1+\rho)}$	CPU (s)	$(J,\ \bar{\ell})$	$RMSE_v$	$\|\|P\mathbf{g}-\hat{\mathbf{v}}\|\|_m$	$RMSE_g$	$\|\|\mathbf{g}-\hat{\mathbf{g}}\|\|_n$	MPE
1	2×10^2	10^2	9.5 - 1.1	(1,28) - (1,28)	69.05 - 69.04	2647.2 - 2647.1	2.07 - 2.07	3528.7 - 3528.8	-7.94
	2×10^3	10^3	9.2 - 0.8	(1,28) - (1,28)	69.05 - 69.04	2647.3 - 2647.1	2.07 - 2.07	3529.1 - 3529.1	-7.95
	2×10^4	10^4	9.3 - 0.9	(1,28) - (1,28)	69.04 - 69.04	2647.1 - 2647.1	2.07 - 2.07	3529.2 - 3529.1	-7.95
	∞	–	9.6 - 0.7	(1,28) - (1,28)	69.04 - 69.05	2647.1 - 2647.3	2.07 - 2.07	3529.2 - 3529.2	-7.95
19	2×10^2	10	9.0 - 0.8	(1,27) - (1,27)	73.27 - 73.27	2809.2 - 2809.2	2.06 - 2.06	3508.7 - 3508.7	-7.95
	2×10^3	10^2	9.3 - 0.9	(1,28) - (1,28)	69.05 - 69.04	2647.3 - 2647.1	2.07 - 2.07	3528.7 - 3528.8	-7.95
	2×10^4	10^3	9.2 - 0.8	(1,28) - (1,28)	69.05 - 69.04	2647.5 - 2647.1	2.07 - 2.07	3529.0 - 3529.1	-7.95
	∞	–	10.3 - 0.8	(1,28) - (1,28)	69.04 - 69.04	2647.1 - 2647.2	2.07 - 2.08	3529.2 - 3529.2	-7.95
199	2×10^3	10	9.3 - 0.9	(1,27) - (1,27)	73.27 - 73.27	2809.2 - 2809.2	2.06 - 2.06	3508.7 - 3508.7	-7.95
	2×10^4	10^2	9.7 - 0.8	(1,28) - (1,28)	69.04 - 69.04	2647.1 - 2647.1	2.07 - 2.07	3528.8 - 3528.8	-7.95
	2×10^5	10^3	9.9 - 0.8	(1,28) - (1,28)	69.04 - 69.04	2647.1 - 2647.1	2.07 - 2.07	3529.1 - 3529.1	-7.95
	∞	–	10.0 - 1.0	(1,28) - (1,28)	69.04 - 69.05	2647.1 - 2647.4	2.07 - 2.07	3529.2 - 3529.1	-7.95
1999	2×10^3	1	9.7 - 0.8	(1,27) - (1,27)	73.82 - 73.82	2830.2 - 2830.2	2.04 - 2.04	3477.5 - 3477.5	-7.93
	2×10^4	10	9.9 - 0.8	(1,27) - (1,27)	73.27 - 73.27	2809.2 - 2809.2	2.06 - 2.06	3508.7 - 3508.7	-7.95
	2×10^5	10^2	9.8 - 0.9	(1,28) - (1,28)	69.04 - 69.05	2647.1 - 2647.3	2.07 - 2.07	3528.8 - 3528.7	-7.95
	∞	–	10.2 - 0.8	(1,28) - (1,28)	69.04 - 69.04	2647.1 - 2647.2	2.07 - 2.07	3529.1 - 3529.2	-7.95

Aquí, las soluciones numéricas obtenidas con $k/(\rho+1) > 10^2$ siguen siendo las mismas independientemente de los valores de ρ y k. Esta vez, la solución del problema reducido se calcula aproximadamente nueve o diez veces más rápido que la solución del problema completo. Es notable que, al contrario de lo que sucede con la red de Winnipeg, aquí obtenemos exactamente los mismos resultados para los modelos completo y reducido. El comportamiento cualitativo de la solución para la demanda se muestra en la figura 4.8, donde se ve el diagrama de dispersión de la matriz O-D actualizada para los modelos completo (a) y reducido (b). La pendiente y la ordenada al origen de la recta de regresión correspondiente en (a) son $m_g = 1.04$, $b_g = 0.03$ y

la pendiente y la ordenada al origen de la línea de regresión en (b) son $m_{g_r} = 1.02$, $b_{g_r} = 7.25$. Los diagramas de dispersión para los flujos de segmento se muestran en la figura 4.9.



(a) Modelo completo. (b) Modelo reducido.

Figura 4.8: Diagramas de dispersión de la matriz O-D obtenida con $k/(\rho + 1) = 10^3$. Red de tránsito de la ZMVM.

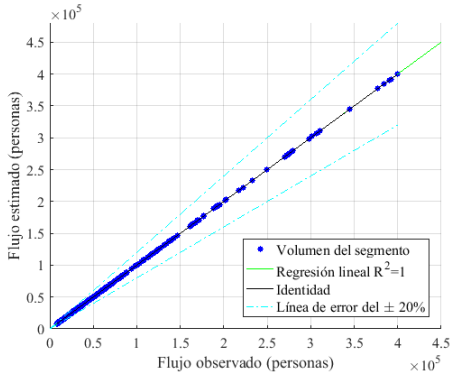


Figura 4.9: Diagrama de dispersión de del flujo en los segmentos obtenidos con $k/(\rho+1) = 10^3$. Red de tránsito de la ZMVM.

La figura 4.10 muestra la diferencia entre la matriz O-D obtenida con el modelo completo y la matriz O-D obtenida con el modelo reducido, denotada por $\mathbf{g}_c - \mathbf{g}_r$; donde es evidente que, la mayor diferencia es muy pequeña y menor que 0.025. Por lo tanto, el modelo de Lagrangiano aumentado con el algoritmo ADMM produce una solución precisa de una manera muy eficiente, al menos para una red grande como la de la ZMVM.

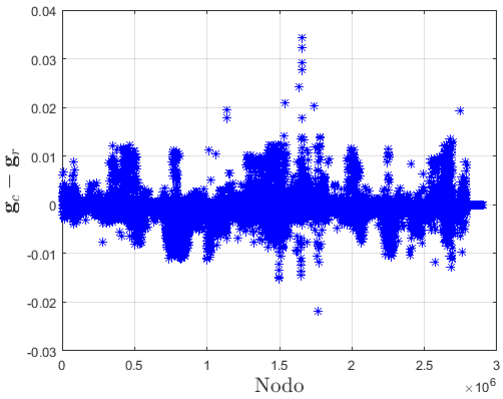


Figura 4.10: Diferencia entre la solución del modelo completo y la solución del modelo reducido para la red de la ZMVM.

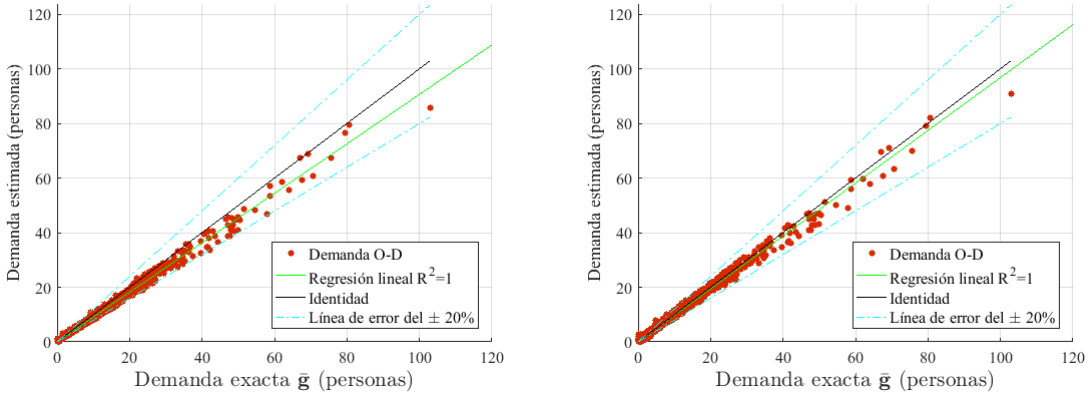
Como se mencionó en la sección 3.5, es posible obtener resultados más realistas si se incorpora más información (datos) en el modelo. Para la red de Winnipeg, implementamos el algoritmo ADMM para resolver el modelo (3.33). La tabla 4.8 muestra estos resultados con $k_1 = k_2 = k_3$, donde se obtuvo $RMSE_g = 0.32$ y $\|\mathbf{g} - \hat{\mathbf{g}}\|_n = 48.9$. Comparando estos resultados con los que se mostraron en la tabla 4.6 para el modelo completo, observamos que hay un incremento en el número de iteraciones y en el tiempo de cómputo. Además, se puede ver que la distancia entre los datos y las estimaciones es mayor en la tabla 4.8 que en la tabla 4.6.

Tabla 4.8: Resultados para la red de Winnipeg obtenidos con el modelo de Lagrangiano aumentado y el algoritmo ADMM para el modelo (3.33).

ρ	k	$\frac{k}{(1+\rho)}$	CPU (s)	$(J,\ \overline{\ell})$	$RMSE_v$	$\|\|P\mathbf{g} - \hat{\mathbf{v}}\|\|_m$
1	2×10^2	10^2	5.8	(26,17)	0.37	4.3
	2×10^3	10^3	5.4	(26,17)	0.37	4.3
	2×10^4	10^4	5.5	(26,17)	0.37	4.3
	∞	–	3.3	(26,17)	0.37	4.3
19	2×10^2	10	5.7	(26,17)	0.36	4.2
	2×10^3	10^2	5.8	(26,17)	0.37	4.3
	2×10^4	10^3	5.5	(26,17)	0.37	4.3
	∞	–	3.2	(26,17)	0.37	4.3
199	2×10^3	10	5.4	(26,17)	0.36	4.1
	2×10^4	10^2	5.4	(26,17)	0.37	4.3
	2×10^5	10^3	5.6	(26,17)	0.37	4.3
	∞	–	3.2	(26,17)	0.37	4.3
1999	2×10^3	1	5.2	(25,17)	0.38	4.5
	2×10^4	10	5.7	(26,17)	0.36	4.2
	2×10^5	10^2	5.5	(26,17)	0.37	4.3
	∞	–	3.1	(26,17)	0.37	4.3

El hecho de que $\|\mathbf{g} - \hat{\mathbf{g}}\|_n$ aumenta cuando se consideran las restricciones (3.31)-(3.32) en el

modelo Lagrangiano, no significa que estos nuevos resultados no sean suficientemente buenos. Para ver si los resultados son mejores o no, deberíamos compararlos con la "solución exacta" $\bar{\mathbf{g}}$; la cual está disponible con nuestros experimentos ya que estamos trabajando con datos sintéticos (claro que la solución exacta no está disponible en un caso real). Obtenemos que: $\|\mathbf{g}-\bar{\mathbf{g}}\|_n = 45.3$ cuando se consideran las restricciones (3.31)-(3.32) en el modelo y $\|\mathbf{g}-\bar{\mathbf{g}}\|_n = 63.3$ cuando esas restricciones no se usan. Por lo tanto, esta información confirma que los nuevos resultados son cercanos a los obtenidos sin las restricciones (3.31)-(3.32), sin embargo, los resultados numéricos mejoran un poco cuando se incorporan los totales marginales de la matriz O-D al modelo, por lo menos para esta red en particular y con los datos correspondientes. La figura 4.11 muestra las dispersiones respectivas para estos resultados con $k/(\rho+1) = 10^3$.



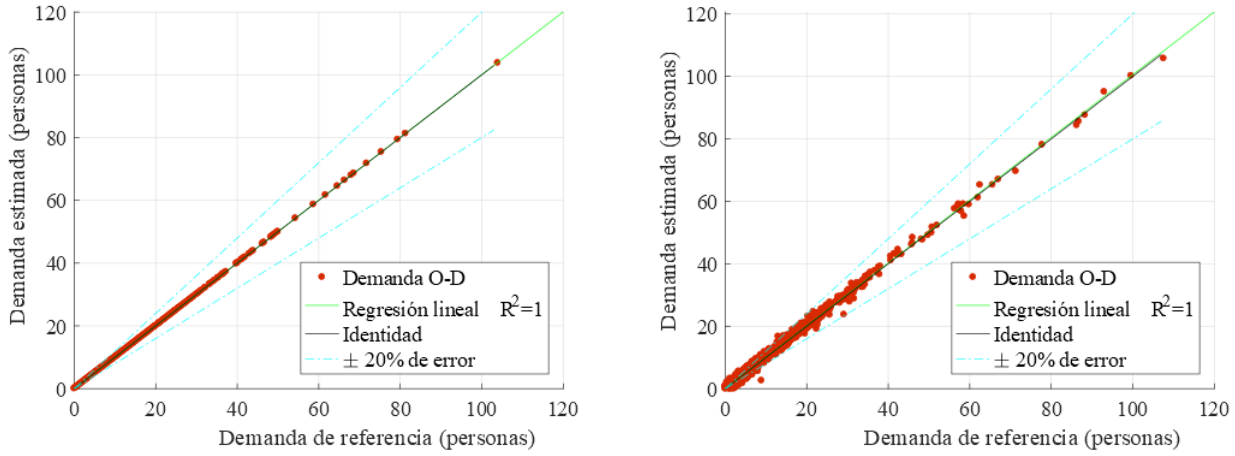
(a) Sin considerar las restricciones (3.31)-(3.32). (b) Considerando las restricciones (3.31)-(3.32).

Figura 4.11: Diagramas de dispersión para la matriz O-D actualizada obtenida contra la demanda O-D "exacta" $\bar{\mathbf{g}}$ con $k/(\rho+1) = 10^3$. Red de tránsito de Winnipeg.

Para evaluar la eficiencia del algoritmo ADMM cuando se esperan diferentes cambios en la demanda, consideremos una demanda *a priori* $\hat{\mathbf{g}} = \bar{\mathbf{g}} + N(0, \delta\bar{\mathbf{g}})$, donde $N(0, \delta\mathbf{g})$ es un vector generado a partir de una distribución normal con media cero y desviación estándar $\delta\bar{\mathbf{g}}$. La tabla 4.9 muestra los resultados numéricos de la red de Winnipeg y los diferentes valores de δ. Estos resultados muestran que la distancia entre la matriz *a priori* y la estimada aumenta en proporción a δ. La figura 4.12 muestra las gráficas de dispersión para la demanda obtenida con diferentes valores de $\hat{\mathbf{g}}$.

Tabla 4.9: Resultados obtenidos para la red de Winnipeg con el modelo de Lagrangiano aumentado y el algoritmo ADMM considerando las restricciones (3.31)-(3.32) y diferentes valores de δ.

δ	$\frac{\|\bar{\mathbf{g}}-\hat{\mathbf{g}}\|}{\|\bar{\mathbf{g}}\|}$	CPU (s)	$(J,\ \bar{\ell})$	$RMSE_v$	$\|\|P\mathbf{g}-\hat{\mathbf{v}}\|\|_m$	$RMSE_g$	$\|\|\mathbf{g}-\hat{\mathbf{g}}\|\|_n$
0.01	0.01	0.7	(1,57)	0.46	5.4	0.01	0.8
0.05	0.05	2.0	(5,31)	0.48	5.5	0.04	5.9
0.10	0.10	3.0	(10,24)	0.29	3.4	0.07	11.1
0.20	0.21	3.8	(19,18)	0.23	2.6	0.15	22.8
0.30	0.31	2.5	(24,14)	0.31	3.6	0.26	39.6



(a) $\hat{\mathbf{g}} = \bar{\mathbf{g}} + N(0, 0.01\bar{\mathbf{g}})$. (b) $\hat{\mathbf{g}} = \bar{\mathbf{g}} + N(0, 0.3\bar{\mathbf{g}})$.

Figura 4.12: Gráficas de dispersión para la demanda obtenida con diferentes valores de $\hat{\mathbf{g}}$, red de tránsito de Winnipeg.

En la Figura 4.13 se muestra la distancia entre la demanda exacta y la matriz de referencia para cada par O-D en azul y la distancia entre la demanda exacta y la estimada para cada par O-D en rojo. Observemos que no hay una variabilidad significativa en estas distancias. Como se mencionó en la introducción de este trabajo, la metodología presentada aquí es para la planificación a corto plazo donde no se esperan cambios drásticos en la demanda.

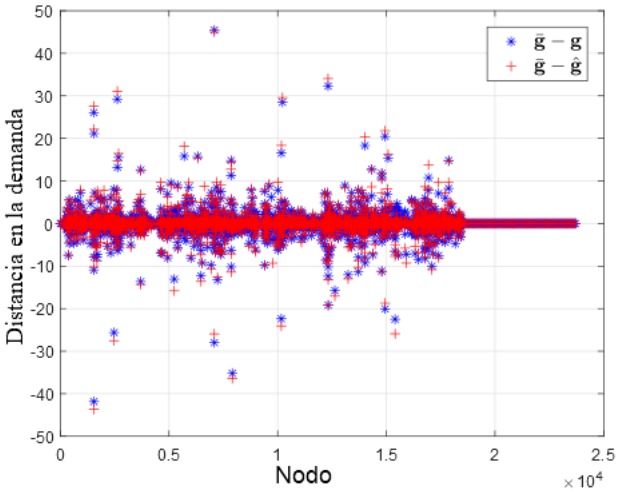


Figura 4.13: Distancias de $\hat{\mathbf{g}}$ y $\mathbf{g}$ a la matriz $\bar{\mathbf{g}}$ para la red de tránsito de Winnipeg.

Capítulo 5

Conclusiones

En este trabajo, estudiamos dos metodologías para estimar matrices O-D en el contexto de redes de tránsito: un modelo penalizado (previamente introducido) combinado con el algoritmo multiplicativo de gradiente conjugado y una nueva metodología, con base en un modelo de Lagrangiano aumentado combinado con el algoritmo de ascenso dual y método de multiplicadores. Mostramos (teórica y numéricamente) que las soluciones del modelo penalizado convergen a la solución del problema de Spiess cuando el parámetro de penalización se aproxima al infinito. El modelo penalizado es equivalente a un modelo de regularización cuadrático y también a algunos modelos con base en promedios ponderados, una propiedad agradable que puede ayudar a obtener estabilidad con respecto a las perturbaciones de los datos de entrada. Estas dos metodologías, muestran que son varias veces más rápidas y más precisas que la metodología de Spiess cuando se emplea un Lagrangiano aumentado para forzar la no negatividad de los coeficientes de la matriz O-D.

Los resultados numéricos, muestran que se obtiene la misma solución, con el mismo tiempo de cómputo, para valores del parámetro de penalización k mayor o igual a 1000. Este comportamiento es consistente con las propiedades de convergencia del modelo penalizado mostradas en la sección 3.2 y con los resultados que se obtienen con modelos de promedios ponderados, como Noriega y Florian (2009) y Verbas et al. (2011); donde se obtuvieron resultados convergentes para valores de $\beta = 0.999$ (equivalente a $k = 1000$, según (3.9)).

Las siguientes dos tablas, muestran una comparación general de las metodologías empleadas en este trabajo para cada red. En la tercera columna de esas tablas, MDM-R indica el método máximo descenso multiplicativo (Spiess) para el modelo reducido. GCM-R y ADMM-R tienen un significado similar.

Tabla 5.1: Comparación general para la red de tránsito de Winnipeg.

k	ρ	Método	CPU(s)	Iters.	$RMSE_v$	$\|P\mathbf{g} - \widehat{\mathbf{v}}\|_m$	$RMSE_g$	$\|\mathbf{g} - \widehat{\mathbf{g}}\|_n$
1000	0	MDM	0.8 s	78	0.17	1.9	0.33	50.9
1000	0	MDM-R	0.3 s	72	0.18	2.1	0.33	50.9
1000	0	GCM	0.3 s	21	0.11	1.3	0.33	51.1
1000	0	GCM-R	0.1 s	21	0.11	1.3	0.33	51.1
20000	19	ADMM	2.2 s	(7,28)	0.09	1.1	0.12	19.2
20000	19	ADMM-R	0.4 s	(3,53)	0.01	0.1	0.18	27.8

Tabla 5.2: Comparación general para la red de tránsito de la ZMVM.

k	ρ	Método	CPU(s)	Iters.	$RMSE_v$	$\|P\mathbf{g} - \widehat{\mathbf{v}}\|$	$RMSE_g$	$\|\mathbf{g} - \widehat{\mathbf{g}}\|$
1000	0	MDM	23.6 s	147	78.92	3025.9	2.54	4324.9
1000	0	MDM-R	5.6 s	147	78.92	3025.9	2.54	4325.1
1000	0	GCM	5.6 s	30	52.34	2006.7	2.64	4499.3
1000	0	GCM-R	1.2 s	30	52.34	2006.7	2.64	4499.7
20000	19	ADMM	9.2 s	(1,28)	69.05	2647.5	2.07	3529.0
20000	19	ADMM-R	0.8 s	(1,28)	69.04	2647.1	2.07	3529.1

El modelo penalizado con GCM brinda la misma solución que el método de Spiess con un menor costo de cómputo, mientras que el modelo de Lagrangiano aumentado con el algoritmo ADMM obtiene soluciones más precisas con un costo de cómputo aún menor para la red de tránsito más grande. El bajo tiempo de CPU de los cálculos muestra que los modelos cuadráticos con algoritmos de solución apropiados siguen siendo una buena opción para las redes de tránsito a gran escala y pueden adaptarse al caso dinámico. Creemos que los resultados obtenidos con estas metodologías pueden mejorarse bajo las siguientes consideraciones:

- Los resultados pueden ser más realistas si se incorpora más información al modelo; por ejemplo, información de la zona, límites superiores e inferiores y costos de viaje. La metodología que aquí presentamos, permite combinar diferentes fuentes de datos de manera relativamente fácil; por ejemplo, encuestas, sensores en los microbuses, torniquetes, tarjetas de tarifas inteligentes y sistemas de localización geográfica.

- La matriz P se puede reducir aún más eliminando aquellas columnas que solo tienen coeficientes nulos (es decir, eliminar la columna j si $P_{i,j} = 0,\ \forall i = 1, 2, ..., m$). En este caso, el valor correspondiente de g_j debe establecerse como $\hat{g}_j$.

- El enfoque de Lagrangiano aumentado se puede implementar para otras restricciones de igualdad y otros algoritmos que han sido efectivos en otros contextos (ver Boyd et al., 2010); por ejemplo, Balakrishnan et al. (1989) utilizaron el método de direcciones alternantes y multiplicadores para el diseño de redes.

- La selección de los segmento donde se tienen conteos, se debe realizar de tal forma que se maximice la cobertura y se minimicen los recursos, como lo mencionan Chootinan et al.

(2005). Además, se pueden desarrollar estrategias para seleccionar conteos adicionales (ver Chen et al., 2007).

Es importante mencionar que al iniciar este trabajo de investigación, usamos el software EMME para llevar a cabo las asignaciones e implementamos nuestros métodos de estimación de matrices O-D en una macro. Los tiempos de cómputo que obtuvimos fueron más pequeños que los tiempos que se obtienen con el algoritmo MDM, parte del software EMME, sin embargo estos tiempos eran del orden de minutos (horas para la red de la ZMVM y el algoritmo MDM). Implementamos rutinas en Python para extraer únicamente los datos que requeríamos de las asignaciones de EMME, guardamos estos datos de tal forma que nos permitieran ahorrar memoria e implementamos los algoritmos en Matlab. Todo esto nos permitió obtener los mismos resultados con tiempos de cómputo mucho menores (segundos).

El siguiente paso en nuestra investigación es ajustar la demanda cuando el modelo de asignación de tránsito considera los límites de capacidad y los efectos de la congestión, el cual es un modelo no lineal. Una forma de resolver este problema, es resolver pequeños subproblemas lineales de forma iterativa, donde se pueden aplicar los algoritmos mostrados en este trabajo para modelos lineales. Debido a la complejidad del problema no lineal, es posible que los algoritmos se ejecuten en un tiempo considerablemente mayor, por lo cual no descartamos la idea de usar técnicas de precondicionamiento, paralelización y supercómputo para poder hacer simulaciones en tiempo real para redes grandes como la de la ZMVM.

Actualmente, contamos con algunos datos obtenidos a partir de la encuesta origen-destino más reciente (INEGI, 2018) para la ZMVM. Un trabajo muy interesante sería aplicar nuestra metodología con este nuevo conjunto de datos y comparar la precisión. Para hacer esto, es importante que primero se actualice nuestra base de datos de la red de tránsito, puesto que en ella no se tienen consideradas a la línea 12 del metro ("la dorada") ni a las rutas más recientes del metrobús.

Otros enfoques que se podrían considerar son:

- Modelos de estadística Bayesiana considerando otro tipo de distribución *a priori* en los datos, por ejemplo Poisson o Gamma, y midiendo la bondad de cada una de éstas en la asimilación de datos.

- Utilizar otro tipo de normas que nos ayuden a identificar los cambios bruscos en la demanda para los centroides que sean cercanos, por ejemplo la norma-1.

- Considerar modelos que nos permitan calcular la demanda de forma dinámica; por ejemplo, dividiendo el periodo de interés en pequeños subintervalos y estimar la demanda en el tiempo t_{i+1} considerando como demanda *a priori* la que se obtuvo en el tiempo t_i. Así, para obtener una buena aproximación en tiempo real, es importante tener algoritmos muy eficientes que resuelvan cada subproblema con un tiempo mucho menor que el tamaño del intervalo a considerar. Es decir, si se desea hacer una simulación con intervalos de 10 minutos, cada subproblema deberá resolverse en un tiempo menor que 10 minutos.

- Considerar además otro tipo de métodos de asignación de tránsito; además del que considera la congestión y los límites de capacidad, otros que consideren diferentes tipos de usuarios (modelos estocásticos) o que dependan del tiempo (modelos dinámicos).

Consideramos que con los algoritmos mostrados en este trabajo, son muy útiles para la planificación de la red de tránsito de la ZMVM; su aplicación permite simular la creación de nuevas líneas de tránsito y modificar de forma adecuada los itinerarios de las líneas de tránsito y así tener una red que funcione de forma más óptima. Además, con los modelos y algoritmos adecuados, este trabajo se puede extender de forma muy eficiente hasta llegar a sugerir rutas de costo mínimo para los usuarios de la red.

Apéndice A

Regresión lineal

La técnica de regresión es una de las herramientas más populares de la estadística. Existen varias formas de regresión como: lineal, no lineal, simple, múltiple, paramétrica, no paramétrica, etc. En este apéndice nos enfocaremos en la regresión lineal. Para ver más detalles acerca de modelos de regresión se puede consultar Draper y Smith (1998). El mayor propósito de la regresión, es explorar la dependencia de una variable con respecto a otras. En una regresión lineal simple, la media de una variable aleatoria Y se modela como una función de otra variable observada x, con la relación

$$E[Y|x] = mx + b \tag{A.1}$$

donde m y b, la pendiente y la ordenada al origen respectivas de la regresión, son parámetros desconocidos. Esto significa que el valor esperado de Y, dado $X = x$, es una función lineal de x.

Para fijar ideas, consideremos el ejemplo de la subsección 3.2.3, en donde ordenaremos los pares O-D como se muestra en la tabla A.1. Además, supongamos que al obtener la matriz *a priori*, por alguna razón se perdieron los datos respectivos al par 4 y al par 15, por lo cual tampoco se obtuvo una estimación de demanda. Supongamos además que la demanda estimada depende linealmente de la demanda *a priori*. Así, el problema consiste en ajustar una recta que resuma la información que se tiene en este conjunto de datos.

Sea n a la cantidad de pares O-D donde se tienen datos (en este caso 23), $\bar{x}$ la media muestral de $\hat{\mathbf{g}}$ y $\bar{y}$ la media muestral de $\mathbf{g}$. Así, se obtiene:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} \hat{g}_i = \frac{1}{23}\sum_{i=1}^{23} \hat{g}_i = 21.7391, \qquad \text{y} \qquad \bar{y} = \frac{1}{23}\sum_{i=1}^{23} g_i = 21.1043.$$

Par O-D	$\hat{\mathbf{g}}$ (*a priori*)	$\mathbf{g}$ (estimada)	Par O-D	$\hat{\mathbf{g}}$ (*a priori*)	$\mathbf{g}$ (estimada)
1	0	0.0	14	5	5.0
2	12	12.0	16	26	26.0
3	4	3.7	17	39	40.3
5	84	81.7	18	23	23.0
6	4	4.0	19	0	0.0
7	0	0.0	20	36	36.0
8	41	37.9	21	53	54.8
9	35	30.8	22	24	24.8
10	12	12.0	23	34	31.1
11	9	9.0	24	47	41.3
12	12	12.0	25	0	0.0
13	0	0.0			

Tabla A.1: Datos de una matriz estimada y una matriz *a priori*.

Definiendo la suma de cuadrados S_{xx} y la suma mixta S_{xy} como

$$\begin{aligned}
S_{xx} &= \sum_{i=1}^{n}(\hat{g}_i - \bar{x})^2 = \sum_{i=1}^{n}\hat{g}_i^2 - 2\bar{x}\sum_{i=1}^{n}\hat{g}_i + \sum_{i=1}^{n}\bar{x}^2 = \sum_{i=1}^{n}\hat{g}_i^2 - 2n\bar{x}^2 + n\bar{x}^2 = \sum_{i=1}^{n}\hat{g}_i^2 - n\bar{x}^2 \\
&= 21304 - 23(21.7391)^2 = 10434.4348 \\
S_{xy} &= \sum_{i=1}^{n}(\hat{g}_i - \bar{x})(g_i - \bar{y}) = \sum_{i=1}^{n} g_i\hat{g}_i - \bar{y}\sum_{i=1}^{n}\hat{g}_i - \bar{x}\sum_{i=1}^{n} g_i + \sum_{i=1}^{n}\bar{x}\bar{y} \\
&= \sum_{i=1}^{n} g_i\hat{g}_i - n\bar{y}\bar{x} - n\bar{x}\bar{y} + n\bar{x}\bar{y} = \sum_{i=1}^{n} g_i\hat{g}_i - n\bar{x}\bar{y} \\
&= 20634.3 - 23(21.7391)(21.1043) = 10082.1261.
\end{aligned}$$

la pendiente m y la ordenada al origen b de la regresión se obtienen de la siguiente manera

$$m = \frac{S_{xy}}{S_{xx}} = \frac{10082.1261}{10434.4348} = 0.9663, \qquad b = \bar{y} - m\bar{x} = 21.1043 - 0.9663(21.7391) = 0.0992.$$

Teniendo los parámetros de la regresión lineal, en caso de que se recuperara información acerca de la demanda de referencia, es posible obtener rápidamente una aproximación de la demanda actual (estimada) sin necesidad de correr nuevamente los algoritmos. Por ejemplo; en el par O-D 4, sabemos que la demanda de referencia es de 23 viajes, así $g_4 = 0.9663(23) + 0.0992 = 22.3241$; en el par O-D 15, sabemos que la demanda de referencia es de 22 viajes, así $g_4 = 0.9663(22) + 0.0992 = 21.3578$.

La correlación de dos variables aleatorias X y Y es un número al cual también se le conoce como coeficiente de correlación y se define como:

$$R^2 := \frac{Cov(X,Y)}{\sigma_x\sigma_y},$$

donde $Cov(X,Y)$ es la covarianza de X y Y, σ_x es la desviación estándar de X y σ_y es la desviación estándar de Y. Si a valores grandes de X le corresponden valores grandes de Y y si a valores pequeños de X le corresponde valores pequeños de Y, entonces R^2 será positivo, de lo contrario será negativo. Así, el signo de $Cov(X,Y)$ proporciona información acerca de la relación entre X y Y. Además, se puede demostrar que $-1 \leq R^2 \leq 1$. La interpretación que se le da a este coeficiente es que si $|R^2|$ es cercano a 1, existe una recta $y = mx + b$, con $a \neq 0$, tal que los valores de (X,Y) tienen una probabilidad alta de estar cerca de de esta línea. Si por el contrario, el valor de R^2 es cercano a cero, se entenderá que no existe una recta que cumpla con estas características; es decir, las variables X y Y son linealmente independientes.

Ahora, para nuestro ejemplo se tienen que

$$\begin{aligned}
Cov(\hat{G},G) :&= E[(\hat{G}-\bar{x})(G-\bar{y})] = \frac{1}{n}\sum_{i=1}^{n}\hat{g}_i g_i - \frac{\bar{y}}{n}\sum_{i=1}^{n}\hat{g}_i - \frac{\bar{x}}{n}\sum_{i=1}^{n} g_i + \frac{1}{n}\sum_{i=1}^{n}\bar{x}\bar{y} \\
&= \frac{1}{n}\sum_{i=1}^{n}\hat{g}_i g_i - \bar{x}\bar{y} = \frac{1}{23}\sum_{i=1}^{23}\hat{g}_i g_i - (21.7391)(21.1043) \\
&= \frac{20634.3}{23} - (21.7391)(20.4148) = 438.355, \\
\sigma_x^2 &= \frac{S_{xx}}{n} = \frac{10434.4348}{23} = 453.6711, \\
\sigma_y^2 &= \frac{1}{n}\sum_{i=1}^{n} g_i^2 - \bar{y}^2 = \frac{20043.7}{23} - (21.1043)^2 = 426.0737, \\
R^2 &= \frac{438.355}{\sqrt{(453.6711)(426.0737)}} = 0.9970.
\end{aligned}$$

Para determinar qué tan buena en nuestra estimación, respecto a un conjunto de datos, es necesario introducir una medida del error. Una de las métricas de error más usada frecuentemente es la raíz del error cuadrático medio $RMSE$, el cual mide la diferencia entre los datos observados ($\hat{\mathbf{g}}$) y los valores calculados con un modelo ($\mathbf{g}$). Se define de la siguiente manera:

$$RMSE := \sqrt{\frac{\sum_{i=1}^{n}(\hat{g}_i - g_i)^2}{n}} = \sqrt{\frac{\sum_{i=1}^{23}(\hat{g}_i - g_i)^2}{23}} = \sqrt{\frac{73.1}{23}} = \sqrt{3.4391} = 1.8545$$

Una de las características del $RMSE$ es que, dado que los errores se elevan al cuadrado antes de hacer el promedio, el RMSE otorga un peso relativamente alto a los errores grandes. Esto significa que el $RMSE$ es más útil cuando los errores grandes son particularmente indeseables.

Algunos investigadores recomiendan usar otro tipo de métricas que permitan observar más claramente si la estimación tiene algún sesgo, por ejemplo el error medio porcentual (MPE), el cual se define como

$$MPE := \frac{100}{n}\sum_{i=1}^{n}\frac{\hat{g}_i - g_i}{\hat{g}_i},$$

sin embargo; esta métrica tiene la desventaja de que no está definida cuando uno de los datos es nulo. Si en nuestro ejemplo eliminamos los pares O-D con demanda nula, el MPE nos queda:

$$MPE = \frac{100}{18} \sum_{i=1}^{18} \frac{\hat{g}_i - g_i}{\hat{g}_i} = 2.24,$$

el cual indica una tendencia a subestimar la demanda alrededor del 2 %, respecto a la demanda *a priori*.

Apéndice B

Códigos

En este anexo se muestran los códigos generados en Matlab para programar cada uno de los algoritmos descritos.

Algoritmo de asignación lineal

```
%Este programa sirve para resolver el problema de asignación de tránsito en la
%red ejemplo de la sección 2.3, para cada par O-D (p,q).

function [S,vol]=EjemploNotasAsignacion(p,q)
%Entrada:
%       p: nodo origen
%       q: nodo destino
%Salida:
%       S  : conjunto de segmentos en la estrategia óptima
%       vol: volumen de personas en cada segmento usado

%Mi infinito
inf=1/eps;


%INTRODUCCIÓN DE DATOS
%Segmentos dirigidos en la red generalizada de la figura 2.4: inicio-fin
%Linea 1 Verde
A=[1 7;7 1;2 7;7 2;5 7;7 5];
%Línea 2 Roja
A=[A;1 8;8 1;3 8;8 3;4 8;8 4];
%Línea 3 Azul
A=[A;3 9;9 3;5 9;9 5;10 9;9 10];
%Línea 4 Negra
A=[A;10 6;6 10;4 6;6 4;5 6;6 5];

%Tiempo de viaje sobre cada segmento
%Linea 1 Verde
ta=[10 10 0 0 15 15];
%Línea 2 Roja
ta=[ta 10 10 0 0 7 7];
%Línea 3 Azul
```

```
ta=[ta 9 9 6 6 0 0];
% Línea 4 Negra
ta=[ta 2 2 0 0 6 6];
ta=ta(:);

% Frecuencia del segmento (2/hdwy)
% Linea 1 Verde hdwy = 7 min
fa=[2/7 inf 2/7 inf 2/7 inf];
% Línea 2 Roja hdwy = 12 min
fa=[fa 1/6 inf 1/6 inf 1/6 inf];
% Línea 3 Azul hdwy = 12 min
fa=[fa 1/6 inf 1/6 inf 1/6 inf];
% Línea 4 Negra hdwy = 5 min
fa=[fa 2/5 inf 2/5 inf 2/5 inf];
fa=fa(:);

%No. de segmentos (deben ser 22)
m = length(A);

%Número de nodos
n = max(A(:));

%Se enumeran los segmentos
B = [A (1:m) '];


%INICIALIZACIÓN
%Todos los tiempos de viaje son infinito excepto en el nodo destino que es cero
u = inf*ones(n,1); u(q) = 0;

%Todas las frecuencias en los nodos son cero
f = zeros(n,1);

%El volúmen en los nodos es cero, excepto en el origen y el destino
v = zeros(n,1); v(p) = demanda; v(q) = -demanda; vol=[];


%PRIMERA ETAPA. ESTRATEGIA ÓPTIMA
%Conjunto con los segmentos en la estrategia
S=[];
while m > 0
    mTT = inf; %mínimo tiempo total
    for id = 1:m
        %Id del segmento sobre el que se hace el test
        a = B(id,3);
        %Nodo al que llega el segmento
        j = A(a,2);
        if u(j) + ta(a) <= mTT
            %Nodo donde inicia el segmento
            i = A(a,1);
            %Actualizar el valor del mínimo tiempo total
            mTT = u(j) + ta(a);
            %Id del segmento con el mínimo tiempo total
            k = a;
```

```
87          end
88      end
89      if mTT < u(i)
90          % El segmento está en la estrategia y se guarda en S=[i j Id uj+ta]
91          S = [S;A(k,:) k mTT];
92
93          % Actualizar los índices usando el criterio: inf*0 = 1
94          if f(i)*u(i) == 0 && (f(i) >= inf || u(i) >= inf)
95              u(i) = (1 + fa(k)*mTT)/(f(i) + fa(k));
96          elseif fa(k)*mTT == 0 && (fa(k) >= inf || mTT >= inf)
97              u(i) = (f(i)*u(i) + 1)/(f(i) + fa(k));
98          else
99              u(i) = (f(i)*u(i) + fa(k)*mTT)/(f(i) + fa(k));
100         end
101         f(i) = f(i) + fa(k);
102     end
103
104     % Se elimina el segmento del conjunto de búsqueda
105     for a = 1:m
106         if B(a,3) == k
107             B = delrow(B,a);
108             break
109         end
110     end
111     % No. de iteraciones
112     m = m - 1;
113 end
114
115
116 %SEGUNDA ETAPA. ASIGNACIÓN DE VOLÚMENES
117 % Número de segmentos
118 m = size(S);
119
120 for k = m:-1:1 % Hacer en orden decreciente de uj+ta
121     % Nodo donde inicia el segmento
122     i = S(k,1);
123     % Nodo donde termina el segmento
124     j = S(k,2);
125     % Id del segmento
126     a = S(k,3);
127
128     % volumen calculado
129     va = fa(a)*v(i)/f(i);
130
131     % Probabilidad de que el par O-D (p,q) use el segmento a=(i,j)
132     pi = va/g;
133
134     if va ~= 0
135         vol = [vol; i j va pi];
136     end
137     v(j) = v(j) + va;
138 end
139 return
```

Donde la función *delrow* es la siguiente subrutina

```
function A=delrow(A,r)
%Esta función sirve para borrar el renglón r de la matriz A
%A = Matriz
%r = Renglón que se desea borrar
[m,n]=size(A);
if r==1
    A=A(2:m,:);
elseif r==m
    A=A(1:m-1,:);
else
    A=[A(1:r-1,:);A(r+1:m,:)];
end
return
```

Algoritmo de máximo descenso multiplicativo

```
function [g,Pg0,Z] = mdmOD(g0,P,v,tol,noMaxIt,k,t1)
%Este programa sirve para resolver el sistema de ecuaciones
%(I+kP'P)g=g0+kP'v, asociado al modelo penalizado para estimar matrices
%origen destino a partir de una matriz obsoleta g0 y volúmenes observados v.
%En este caso, la solución se calcula de manera iterativa mediante la
%formula g1 = g0 + alfa*dM0, con dM0 = g0.*d0.
%Resolviendo el problema de min_alfa Z(g0+alfa*dM0) se obtiene que
%alfa = -<N0,dM0>/<gama*dM0,A*dM0>
%     = -<N0,dM0>/gama*(<dM0,dM0>+k*<P*dM0,P*dM0>),
%en donde N0 = gama*k*P'(P*g0-v). El nuevo gradiente se calcula como
%N1 = N0 + gama*alfa*(I+k*P'*P)*dM0 = N0 + gama*alfa*(dM0+k*P'*P*dM0)
%y el gradiente multiplicativo se calcula NM1 = g1.*N1.
%La dirección de descenso contraria a la del
%gradiente dM1 = -GM1
%input
%   g0      : Punto de inicio (demanda obsoleta)
%   P       : Matriz de probabilidades de ruta
%   v       : Vector de volúmenes observados
%   tol     : Tolerancia para el criterio de paro
%   noMaxIt: Número máximo de iteraciones
%   k       : Factor de penalización en el modelo
%   t1      : Factor de reescalamiento para evitar overflow
%
%output
%   g : Demanda estimada
%   i : Número de iteraciones realizadas
%   Z : Función objetivo [Z_g,Z_v,Z], donde Z_g = <g-g0,g-g0>
%       Z_v = <Pg-v,Pg-v> y Z = (t1/2)Z_g
%
%NOTA: Este algoritmo se encuentra programado en una macro para EMME,
%      donde utilizo un factor de reescalamiento t1 = 0.001/k
%if nargin <6
%    k = 100;
%    t1 = 0.001/2;
```

```
35 %end
36 gama = t1;
37 g = g0;
38
39 %Cálculo del primer gradiente y su norma
40 Pg0 = P*g;
41 difv = Pg0 - v;
42 r0 = gama*(g - g0 + k*P'*difv);
43 paro = norm(r0); paro = tol*paro;
44
45 %Primera dirección de descenso
46 dM = -g.*r0;
47
48 %N?mero de iteraciones
49 i = 1; Z = [];
50 while (1)
51     %Valores de la función objetivo
52     Zg = (g-g0)'*(g-g0);
53     Zv = difv'*difv;
54     Zt = (gama/2)*(Zg + k*Zv);
55     Z = [Z; Zg, Zv, Zt];
56     if i == noMaxIt
57         disp('mdMOD: Se requieren mas iteraciones para alcanzar la convergencia')
58         fprintf('La norma del gradiente en la iteración %d es: %f \n',i,nr)
59         break
60     end
61
62     %Calcular el valor de alfa = -<r0,dM>/(gama<dM,(I+kP'P)*dM>)
63     PdM = P*dM;
64     AdM = dM + k*P'*PdM;
65     alfa = -(r0'*dM)/(gama*dM'*AdM);
66 %     if alfa >1
67 %         disp('Warning: alfa > 1')
68 %         alfa = min(alfa,1);
69 %     end
70
71     %Nuevo valor de g
72     g = g + alfa*dM;
73
74     %Proyectamos la solución en el conjunto factible
75     g = (g > 0).*g; %(Vollebregt, 2014)
76
77     Pg0 = P*g;
78     difv = Pg0 - v;
79
80     %Gradiente más actual y su norma r1 = r0 + gama*alfa*(dM0 + k*P'*PdM)
81     r1 = r0 + gama*alfa*AdM;
82     nr = norm(r1);
83
84     %Prueba de convergencia
85     if nr <= paro
86         fprintf('mdmOD: Criterio del gradiente en la iteración %d \n',i)
87         %fprintf('La norma del gradiente en la iteración %d es: %f \n',i,nr)
88         Zg = (g-g0)'*(g-g0);
```

```
        Zv = difv'*difv;
        Z = [Z; Zg, Zv, (gama/2)*(Zg + k*Zv)];
        break
    else
        % Cálculo de la nueva dirección de descenso
        dM = -g.*r1;
        r0 = r1;
        i = i +1;
    end
end
return
```

Algoritmo de gradiente conjugado multiplicativo

```
function [g,Pg0,grad,i,Z] = gcMOD(g0,P,v,tol,noMaxIt,k,t1)
% Este programa sirve para resolver el sistema de ecuaciones
% (I+kP'P)g=g0+kP'v, asociado al modelo penalizado para estimar matrices
% origen destino a partir de una matriz obsoleta g0 y volúmenes observados v.
% En este caso, la solución se calcula de manera iterativa mediante la
% formula g1 = g0 + alfa*dM0, con dM0 = g0.*d0
% Resolviendo el problema de min_alfa Z(g0+alfa*dM0) se obtiene que
% alfa = -<N0,dM0>/<t1*dM0,A*dM0>
%      = -<N0,dM0>/t1*(<dM0,dM0>+k*<P*dM0,P*dM0>)
% en donde N0 = t1*k*P'(P*g0-v). El nuevo gradiente se calcula como
% N1 = N0 + t1*alfa*(I+k*P'*P)*dM0 = N0 + t1*alfa*(dM0+k*P'*P*dM0)
% y el gradiente multiplicativo se calcula NM1 = g1.*N1.
% Se propone calcular la nueva dirección de descenso como:
% dM1 = -GM1 + beta*dM0, en donde el valor de beta para que <dM1,A*dM0> = 0
% debe ser: beta = <GM1,G1-G0>/<dM0,G1-G0>
% input
%    g0      : Punto de inicio (demanda obsoleta)
%    P       : Matriz de probabilidades de ruta
%    v       : Vector de volúmenes observados
%    tol     : Tolerancia para el criterio de paro
%    noMaxIt: Número máximo de iteraciones
%    k       : Factor de penalización en el modelo
%    t1      : Factor de reescalamiento para evitar overflow
%
% output
%    g : Demanda estimada
%    i : Número de iteraciones realizadas
%    Z : Función objetivo [Z_g,Z_v,Z], donde Z_g = <g-g0,g-g0>
%        Z_v = <Pg-v,Pg-v> y Z = (t1/2)Z_g
%
%NOTA: Este algoritmo se encuentra programado en una macro para EMME,
%      donde utilizo un factor de reescalamiento t1 = 0.001/k
% if nargin <6
%     k = 100;
%     t1 = 0.001/2;
% end
g = g0;

```

```
39 % Cálculo del primer gradiente y su norma
40 Pg0 = P*g;
41 difv = Pg0 - v;
42 r0 = t1*(g - g0 + k*P'*difv);
43 nr = norm(r0); paro = tol*nr;
44
45 grad = nr;
46 % Primera dirección de descenso
47 dM = -g.*r0;
48
49 %Número de iteraciones
50 i = 1; Z = [];
51 while (1)
52     % Valores de la función objetivo
53     Zg = (g-g0)'*(g-g0);
54     Zv = difv'*difv;
55     Zt = (t1/2)*(Zg + k*Zv);
56     Z = [Z; Zg, Zv, Zt];
57     if i == noMaxIt
58         disp('gcMOD: Se requieren mas iteraciones para alcanzar la convergencia')
59         fprintf('La norma del gradiente en la iteración %d es: %f \n',i,nr)
60         break
61     end
62
63     % Calcular el valor de alfa = -<N0,dM>/(t1<dM,dM> + 2*t1<P*dM,P*dM>)
64     PdM = P*dM;
65     AdM = dM + k*P'*PdM;
66     alfa = -(r0'*dM)/(t1*dM'*AdM);
67
68     % Nuevo valor de g
69     g = g + alfa*dM;
70
71     % Proyectamos la solucion en el conjunto factible
72     g = (g > 0).*g; % (Vollebregt, 2014)
73
74     Pg0 = P*g;
75     difv = Pg0 - v;
76
77     % Gradiente más actual y su norma r1 = r0 + t1*alfa*AdM
78     r1 = r0 + t1*alfa*AdM;
79     nr = norm(r1);
80     rm1 = g.*r1;
81     grad = [grad;nr];
82
83     % Prueba de convergencia
84     if nr <= paro
85         fprintf('gcMOD: Criterio del gradiente en la iteración %d \n',i)
86         Zg = (g-g0)'*(g-g0);
87         Zv = difv'*difv;
88         Z = [Z; Zg, Zv, (t1/2)*(Zg + k*Zv)];
89         break
90     else
91         % Cálculo de la nueva dirección de descenso
92         difN = r1-r0;
```

```
        beta = (rm1'*difN)/(dM'*difN);
        dM = -rm1 + beta*dM;
        r0 = r1;
        i = i +1;
    end
end
return
```

Algoritmo de ascenso dual y multiplicadores

```
%     Cargar datos (gOld,P,v_obs,rho,k)
load('ZMVM2.mat')

%Número de parámetros a estimar
no_zones = length(gOld);

%Número de segmentos donde se tienen conteos
no_cont = length(v_obs);

%Parámetros para los métodos de descenso
tol = 0.001; noMaxIt = 299;

%Parámetros para los métodos con Lagrangiano
ItAD = 49; tolAD = 0.1; tolADgc = 0.25;
paro = tol*norm(gOld);

disp('%Lagrangiano Aumentado ((1+r)I + kP*P)g = g^ + kP*v^ + mu + ry2  %')

fprintf('Las penalizaciones son: rho=%i, k=%i, k/(rho+1)=%f\n',rho,k,k/(rho+1))

%Inicializamos las variables
mu = 0*gOld; y2 = gOld; g = gOld; itGC = [];
j = 1; tinicio = cputime;
while j <= ItAD
        %Actualizar g
        [g,it] = gcLA(gOld,g,P,v_obs,mu,rho,y2,tol,noMaxIt,k,tolADgc);
        itGC = [itGC it];

        dg = [dg;norm(g-gOld)];
        dv = [dv;norm(P*g-v_obs)];
        %Actualizar y
        y2 = g - mu/rho; y2 = (y2 >= 0).*y2;

        %Prueba de convergencia
        if norm(y2-g) <= paro
                fprintf('Ascenso Dual termino en %d iteraciones \n',j)
                fprintf('Promedio de iteraciones de GC: %f\n',round(mean(itGC)))
                g = (g >= 0).*g;
                break
        else
                %Actualizar mu
                mu = mu + rho*(y2 - g);
```

```
43                  g = y2 + mu/rho;
44          end
45
46          j = j + 1;
47          if j >= ItAD
48                  fprintf('ADMM requiere más de %d iteraciones.\n',j)
49                  fprintf('Promedio de iteraciones de GC: %f\n',round(mean(itGC)))
50                  break
51          end
52  end
53
54  v = P*g;
55  fprintf('Costo computacional: %f \n',round(10*(cputime - tinicio))/10)
```

Donde la línea 26 manda llamar la siguiente subrutina de gradiente conjugado tradicional.

```
1   function [g,i] = gcLA(gOld,g0,P,v,mu,rho,y2,tol,noMaxIt,k,tolADgc)
2   g = g0;
3
4   % Cálculo del primer gradiente y su norma
5   Pg = P*g;
6   difv = Pg - v;
7   r0 = (1+rho)*g - gOld + k*P'*difv - mu - rho*y2;
8   nr = norm(r0); paro = tol*nr;
9
10  % Primera dirección de descenso
11  d = -r0;
12
13  % Número de iteraciones
14  i = 1;
15  while (1)
16      if i == noMaxIt
17          disp('gcLA: Se requieren mas iteraciones para alcanzar la convergencia')
18          % fprintf('La norma del gradiente en la iteración %d es: %f \n',i,nr)
19          break
20      end
21
22      % Calcular el valor de alfa = -<N0,dM>/(t1<dM,dM> + 2*t1<P*dM,P*dM>)
23      Pd = P*d;
24      Ad = (1+rho)*d + k*P'*Pd;
25      alfa = -(r0'*d)/(d'*Ad);
26
27      % Nuevo valor de g
28      g = g + alfa*d;
29      Pg = P*g;
30      difv = Pg - v;
31
32      % Gradiente más actual y su norma r1 = r0 + t1*alfa*AdM
33      r1 = r0 + alfa*Ad;
34      nr = norm(r1);
35
36      % Prueba de convergencia (usar un número pequeño de iteraciones)
37      if nr <= paro
38          fprintf('gcLA: Criterio del gradiente en la iteración %d \n',i)
39          break
```

```
    elseif min(g) <= -tolADgc
        % Salir para actualizar el multiplicador mu
        fprintf('gcLA: El mínimo de g es: %f, iteración %d \n',min(g),i)
        break
    else
        % Cálculo de la nueva dirección de descenso
        difN = r1-r0;
        beta = (r1'*difN)/(d'*difN);
        d = -r1 + beta*d;
        r0 = r1;
        i = i +1;
    end
end
return
```

Bibliografía

Alsger A., Assemi B., Mesbah M., Ferreira L. (2016) Validating and improving public transport origin-destination estimation algorithm using smart card fare data. *Transportation Research Part C: Emerging Technologies* 68:490–506.

Antoniou C., Barceló J., Breen M., Bullejos M., Casas J., Cipriani E., Ciuffo B., Djukic T., Hoogendoorn S., Marzano V., Montero L., Nigro M., Perarnau J., Punzo V., Toledo T., Lint H. (2016) Towards a generic benchmarking platform for origin-destination flows estimation/updating algorithms: Design, demonstration and validation. *Transportation Research Part C: Emerging Technologies* 66:79–98.

Ashok K., Ben-Akiva M. E. (2002) Estimation and Prediction of Time-Dependent Origin-Destination Flows with a Stochastic Mapping to Path Flows and Link Flows. *Transportation Science* 36(2):184–198.

Balakrishnan A., Magnanti T. L., Wong R. T. (1989) A Dual-Ascent Procedure for Large-Scale Uncapacitated Network Design. *Operations Research* 37(5):716–740.

Bell M. G. H. (1991) The estimation of origin-destination matrices by constrained generalized least squares. *Transportation Research Part B: Methodological* 25(1):13–22.

Bera S., Rao K. V. K. (2011) Estimation of origin-destination matrix from traffic counts: the state of the art. *European Transport* (49):3–23.

Bierlaire M. (1995) Mathematical models for transportation demand analysis. Ph.d. thesis, Facultés Universitaires Notre-Dame de la Paix de Namur, Faculté des Sciences, Départment de Mathématique, Namur.

Bierlaire M., Toint L. (1995) MEUSE: An origin-destination matrix estimator that exploits structure. *Transportation Research, Part B: Methodological* 29(1):47–60.

Boyd S., Parikh N., Chu E., Peleato B., Eckstein J. (2010) Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning* 3(1):1–122.

Cascetta E. (1984) Estimation of trip matrices from traffic counts and survey data: A generalized least squares estimator. *Transportation Research Part B: Methodological* 18(4-5):289–299.

Cascetta E., Nguyen S. (1988) A unified framework for estimating or updating origin/destination matrices from traffic counts. *Transportation Research Part B: Methodological* 22(6):437–455.

Cascetta E., Papola A., Marzano V., Simonelli F., Vitiello I. (2013) Quasi-dynamic estimation of o-d flows from traffic counts: Formulation, statistical validation and performance analysis on real data. *Transportation Research Part B: Methodological* 55:171–187.

Chávez M. V. (2014) Modelos matemáticos para análisis de demanda en transporte. Tesis de maestría, Universidad Autónoma Metropolitana-Iztapalapa, Departamento de matemáticas, Ciudad de México.

Chávez M. V., Juárez L. H. (2014) A Multiplicative Conjugate Gradient Method for the O-D Adjustment Matrix. *R. Z. Ríos-Mercado et al. (Eds.): Recent Advances in Theory, Methods, and Practice of Operations Research* pp. 97–104.

Chávez M. V., Juárez L. H. (2016) Demand Adjustment for the Transit Network of Mexico City and its Surroundings. *Congreso Panamericano de Ingeniería y Tránsito, Transporte y Logística (PAMAM 2016)* pp. 1338–1352.

Chávez M. V., Juárez L. H., Ríos Y. A. (2019) Penalization and Augmented Lagrangian for O-D Demand Matrix Estimation from Transit Segment Counts. Transportmetrica A: Transport Science 15(2):915–943.

Chen A., Pravinvongvuth S., Chootinan P., Lee M., Recker W. (2007) Strategies for Selecting Additional Traffic Counts for Improving O-D Trip Table Estimation. Transportmetrica A: Transpor Science 3(3):191–211.

Chootinan P., Chen A., Yang H. (2005) A Bi-Objective Traffic Counting Location Problem for Origin-Destination Trip Table Estimation. *Transportmetrica A: Transpor Science* 1(1):65–80.

Cipriani E., Florian M., Mahut M., Nigro M. (2011) A gradient approximation approach for adjusting temporal origin-destination matrices. *Transportation Research Part C: Emerging Technologies* 19(2):270–282.

Codina E., Barceló J. (2000) Adjustment of O-D trip matrices from traffic counts: an algorithmic approach based on conjugate directions. *Proceedings of the 8th Euro Working Group on Transportation* pp. 427–432.

Codina E., García R., Marín A. (2006) New algorithmic alternatives for the O-D matrix adjustment problem on traffic networks. *European Journal of Operation Research* 175(3):1484–1500.

Doblas J., Benitez F. G. (2005) An approach to estimating and updating origin-destination matrices based upon traffic counts preserving the prior structure of a survey matrix. *Transportation Research Part B: Methodological* 39(7):565–591.

Draper N. R., Smith H. (1998) Applied Regression Analysis, 3rd edn. *John Wiley and Sons*.

Etemadnia H., Abdelghany K. (2009) Distributed approach for estimation of dynamic origin-destination demand. *Journal of the Transportation Research Board* 2105:127–134.

Fernández A. G. (2013) Modelos matemáticos de asignación de tránsito: Aplicación a la red metropolitana de Ciudad de México. Tesis de maestría, Universidad Autónoma Metropolitana-Iztapalapa, Departamento de matemáticas, Ciudad de México.

Florian M., Chen Y. (1995) A coordinate descent method for the bi-level O/D matrix adjustment problem. *International Transactions on Operations Research* 2(2):165–179.

Frederix R., Viti F., Tampère C. M. J. (2013) Dynamic origin-destination estimation in congested networks: theoretical findings and implications in practice. *Transportmetrica A: Transport Science* 9(6):494–513.

Fujita M., Yamada S., Murakami S. (2016) Time Coefficient Estimation for Hourly Origin-Destination Demand from Observed Link Flow Based on Semidynamic Traffic Assignment. *Journal of Advanced Transportation* 2017:1–17.

Heidari A. A., Moayedi A., Abbaspour R. A. (2017) Estimating origin-destination matrices using an efficient moth flame-based spatial clustering approach. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLII-4/W4* pp. 381–387.

Hu X., Chiu Y., Villalobos J. A., Nava E. (2017) A Sequential Decomposition Framework and Method for Calibrating Dynamic Origin-Destination Demand in a Congested Network. *IEEE Transactions on intelligent transportation systems* 18(10):2790–2797.

INEGI (2007) Encuesta 2007, Origen-Destino. Tech. rep., Instituto Nacional de Estadística y Geografía.

INEGI (2018) Encuesta Origen-Destino en Hogares de la Zona Metropolitana del Valle de México 2017. Tech. rep., Instituto Nacional de Estadística y Geografía.

INRO (2018) The evolution of transport planning. Disponible en http://www.inrosoftware.com/en/products/emme/index.php.

Juárez L. H., Chávez M. V. (2014) O-D Matrix Adjustment for Transit Networks by Conjugate Gradient Iterations. *Investigación operacional* 36(2):115–126.

Juárez L. H., Fernández A. G., Delgado J., Chávez M. V., Omaña E. (2013) Asignación de tránsito en la red metropolitana del Valle de México y su impacto en el STC-Metro. *Contactos. Revista de educación en ciencias e ingeniería* (90):85–95.

Kumar A. A., Kang J. E., Kwon C., Nikolaeva A. (2016) Inferring origin-destination pairs and utility-based travel preferences of shared mobility system users in a multi-modal environment. *Transportation Research Part B: Methodological* 91:270–291.

Lundgren J., Peterson A. (2008) A heuristic for the bilevel origin–destination-matrix estimation problem. *Transportation Research Part B: Methodological* 42(4):339–354.

Malapert A., Kuusinen J. M. (2017) Estimation of elevator passenger traffic based on the most likely elevator trip origin-destination matrices. *Building Services Engeenering Research and Technology* 38(5):563–579.

Michau G., Pustelnik N., Borgnat P., Bhaskar A., Chung E. (2017) A primal-dual algorithm for link dependent origin destination matrix estimation. *IEEE Transactions on Signal and Information Processing over Networks* 3(1):104–113.

Nocedal J., Wright S. (2006) Numerical optimization, 2nd edn. *Springer Series in Operations Research and Financial Engineering*.

Noriega Y., Florian M. (2009) Some enhancements of the gradient method for O-D matrix adjustment. Tech. Rep. 4, CIRRELT.

Nuzzolo A., Comi A. (2016) Advanced public transport and intelligent transport systems: new modelling challenges. *Transportmetrica A: Transport Science* 12(8):674–699.

Pitombeira-Neto A. R., Grangeiro C. F., Carvalho L. E. (2016) Bayesian inference on dynamic linear models of day-to-day origin-destination flows in transportation networks. *Disponible en https://arxiv.org/abs/1608.06682*.

Shafiei S., Saberi M., Sarvi M. (2016) Application of an exact gradient method to estimate dynamic origin-destination demand for Melbourne network. *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)* pp. 1945–1950.

Shen W., Winter L. (2012) A new one-level convex optimization approach for estimating origin-destination demand. *Transportation Research Part B* 46(2012):1535–1555.

Sherali H. D., Sivanandan R., Hobeika A. G. (1994) A linear programming approach for synthesizing origin-destination trip tables from link traffic volumes. *Transportation Research Part B: Methodological* 8(3):213–233.

Spiess H. (1990) A gradient approach for the O-D matrix adjustment problem. EMME/2 Support Center, Switzerland, http://www.spiess.ch/emme2/demadj/demadj.html.

Spiess H., Florian M. (1989) Optimal strategies: A new assignment model for transit networks. *Transportation Research Part B: Methodological* 23(2):83–102.

Torres P. (2013) Subgerente de planeación estratégica, Sistema de Transporte Colectivo - Metro http://www.metro.cdmx.gob.mx

Verbas I. O., Mahmassani H. S., Zhang K. (2011) Time-Dependent Origin-Destination Demand Estimation. *Transportation Research Board* 2263:45–56.

Vollebregt E. A. H. (2014) The Bound-Constrained Conjugate Gradient Method for Nonnegative Matrices. *Journal of Optimization Theory and Applications* 162(3):931–953.

Xie C., Kockelman K. M., Travis S. (2011) A maximum entropy-least squares estimator for elastic origin-destination trip matrix estimation. *Transportation Research Part B: Methodological* 45(9):1465–1482.

Casa abierta al tiempo
UNIVERSIDAD AUTÓNOMA METROPOLITANA

ACTA DE DISERTACIÓN PÚBLICA

No. 00064
Matrícula: 2143805674

OPTIMIZACIÓN CONVEXA Y MÉTODOS VARIACIONALES PARA ESTIMAR MATRICES ORIGEN-DESTINO.

En la Ciudad de México, se presentaron a las 16:00 horas del día 3 del mes de julio del año 2019 en la Unidad Iztapalapa de la Universidad Autónoma Metropolitana, los suscritos miembros del jurado:

DR. DAVID GUILLERMO ROMERO VARGAS
DR. LORENZO HECTOR JUAREZ VALENCIA
DR. MIGUEL ANGEL GUTIERREZ ANDRADE
DR. MARCOS AURELIO CAPISTRAN OCAMPO
DR. JOAQUIN DELGADO FERNANDEZ

Bajo la Presidencia del primero y con carácter de Secretario el último, se reunieron a la presentación de la Disertación Pública cuya denominación aparece al margen, para la obtención del grado de:

DOCTORA EN CIENCIAS (MATEMATICAS)

DE: MARIA VICTORIA CHAVEZ HERNANDEZ

y de acuerdo con el artículo 78 fracción IV del Reglamento de Estudios Superiores de la Universidad Autónoma Metropolitana, los miembros del jurado resolvieron:

APROBAR

Acto continuo, el presidente del jurado comunicó a la interesada el resultado de la evaluación y, en caso aprobatorio, le fue tomada la protesta.

MARIA VICTORIA CHAVEZ HERNANDEZ
ALUMNA

REVISÓ

MTRA. ROSALIA SERRANO DE LA PAZ
DIRECTORA DE SISTEMAS ESCOLARES

DIRECTOR DE LA DIVISIÓN DE CBI	PRESIDENTE	VOCAL
DR. JESUS ALBERTO OCHOA TAPIA	DR. DAVID GUILLERMO ROMERO VARGAS	DR. LORENZO HECTOR JUAREZ VALENCIA
VOCAL	**VOCAL**	**SECRETARIO**
DR. MIGUEL ANGEL GUTIERREZ ANDRADE	DR. MARCOS AURELIO CAPISTRAN OCAMPO	DR. JOAQUIN DELGADO FERNANDEZ