DEPARTAMENTO DE MATEMÁTICA APLICADA Y ESTADÍSTICA

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS AERONÁUTICOS

METODOLOGÍA PARA EL DISEÑO DE REDES DE TRANSPORTE Y PARA LA ELABORACIÓN DE ALGORITMOS EN PROGRAMACIÓN MATEMÁTICA CONVEXA DIFERENCIABLE

Tesis doctoral

por

Ricardo García Ródenas
Licenciado en CC. Matemáticas

dirigida por

Ángel Marín Gracia
Doctor Ingeniero Aeronáutico

2001

Tribunal nombrado por el Mgfco. y Excmo. Sr. Rector de la Universidad Politécnica de Madrid, el día de de 2001

Presidente D.

Vocal D.

Vocal D.

Vocal D.

Secretario D.

Realizado el acto de defensa y lectura de la Tesis el día de de 2001 en

Calificación:

El Presidente El Secretario Los Vocales

A los que no están,
porque se han ido o
porque están por llegar.

AGRADECIMIENTOS

Desearía expresar mi agradecimiento a las personas e instituciones que han hecho posible la realización de esta tesis doctoral.

En primer lugar agradezco al profesor D. Ángel Marín, del Departamento de Matemática Aplicada y Estadística de la Universidad Politécnica de Madrid, el haberme introducido y guiado en la investigación durante todos estos años. Han sido tantas las reuniones y las vivencias que además de ser mi Director de Tesis también es mi amigo.

Al profesor Michael Patriksson, del Departamento de Matemáticas de la Universidad Tecnológica de Chalmers, por el intercambio de ideas que han enriquecido profundamente el contenido de esta tesis.

A los profesores Juan de Dios Ortúzar, Enrique Fernández y Joaquín De Cea, del Departamento de Ingeniería de Transporte de la Pontificia Universidad Católica de Chile, por su introducción en la modelización del transporte, en especial, de los modelos de asignación en redes de transporte público.

Al profesor Doroteo Verástegui, compañero de la Escuela Universitaria Politécnica de Almadén, le agradezco toda su generosidad y amistad que ha mostrado desde el primer día que nos conocimos.

A Sally Newton, mi profesora de inglés, por sus miles de redondeles rojos, que pese a su esfuerzo, siguen siendo imprescindibles.

A David Esteban, alumno de la E. T. S. I. Aeronáuticos, por realizar la experiencia numérica de la sección 6.7.

A la Universidad de Castilla-La Mancha y a la Universidad Politécnica de Madrid por darme todas las facilidades posibles, en particular, a la Escuela Universitaria Politécnica de Almadén y a la Escuela Técnica Superior de Ingenieros Aeronáuticos.

Al Ministerio de Educación y Cultura que a través de la ayuda TRA99-1156-C02-01 del Plan Nacional de I+D, Programa de Transporte, ha sufragado parte de los costes de esta investigación.

A mi Madre, que si hubiera sabido, sin duda, me la hubiera escrito. A toda mi Familia (estais todos, hasta los que, hoy por hoy, no sabeis leer estas líneas).

Las últimas palabras que escribo son para M$^{\underline{a}}$ Luz. Son las últimas porque son las más difíciles, porque las que hoy escribo mañana me parecen insuficientes, por esta incapacidad de circunscribir las palabras a mis emociones, como lo hacen los cardinales a los conjuntos.

Ciudad Real, 31 de mayo de 2001

Índice General

Índice de Figuras

Índice de Tablas

Resumen

Esta tesis doctoral se centra en el desarrollo de métodos y modelos para la planificación del transporte urbano. El denominador común en los modelos desarrollados, es que recogen explícitamente el transporte con modos combinados, esto es, viajes donde los usuarios emplean más de un modo de transporte.

Un tema central en esta tesis ha sido el desarrollo de los *métodos de descomposición simplicial* que constituyen el estado-del-arte en los modelos de asignación en equilibrio. Para este objetivo, se ha elaborado la clase CG/SD, que mejora significativamente los métodos clásicos de descomposición simplicial. Esta clase también constituye una generalización de los métodos de direcciones factibles empleados en programación matemática no lineal.

Los objetivos perseguidos han sido:

(a) Modelización del comportamiento de los usuarios de viajes con modos combinados.

(b) Resolución numérica de estos modelos mediante la clase CG/SD.

(c) Calibración de los datos de entrada de estos modelos.

(d) Diseño de redes de transporte con modos combinados.

Los objetivos (a) y (b) son abordados en los capítulos 1, 2 y 3 mediante la programación matemática de un solo nivel. Los objetivos (c) y (d) son estudiados, mediante la programación matemática binivel, en los capítulos 4, 5 y 6. Esto confiere dos partes diferenciadas, la primera basada en una metodología uninivel para la que se desarrollan algoritmos exactos y la segunda, basada en la programación matemática binivel, en la que, debido a la gran complejidad y dimensiones de los problemas, se desarrolla una metodología heurística para su resolución.

En el capítulo 1 se desarrolla un modelo de equilibrio en redes con modos combinados. La formulación matemática empleada es la de desigualdades variacionales y se elaboran dos algoritmos de descomposición simplicial para su resolución, que son testados numéricamente en redes simétricas. Se ilustra este modelo, el TAP-M, para el diseño paramétrico de intercambiadores multimodales urbanos.

La resolución de la versión *simétrica* del modelo TAP-M motivó el estudio de los algoritmos de descomposición simplicial y originó, junto a las aportaciones de Michael Patriksson, la clase CG/SD.

En el capítulo 2 se establece esta clase de algoritmos para problemas de programación matemática convexa diferenciable, demostrando su convergencia asintótica y dando condiciones suficientes, basadas en la geometría del problema, en condiciones de regularidad de las soluciones y en propiedades del algoritmo de generación de columnas, que garantizan la convergencia finita.

El capítulo 3 está dedicado al estudio numérico de la clase CG/SD, para lo que se emplean dos problemas test: el TAP-M y el problema convexo, no lineal, separable y uniproducto de flujo en redes. Se analiza la eficiencia de los métodos CG/SD en función de sus parámetros y del papel de la prolongación de las columnas generadas. Se hace una comparativa entre los nuevos métodos que introduce la clase CG/SD con los métodos clásicos de descomposición simplicial y de direcciones factibles.

El capítulo 4 está dedicado a la estimación de los datos necesarios para la aplicabilidad del TAP-M. Estos son una matriz de viajes origen-destino y un vector de parámetros que definen el modelo

de demanda. Se ha formulado un modelo binivel, denominado CDAM, para la estimación de estos valores que permite aprovechar la información derivada de mediciones de flujo en la red multimodal, de matrices desactualizadas y/o de la realización de encuestas. Se ha estudiado el problema de la *sobrespecificación de los parámetros* y se ha desarrollado una clase de algoritmos heurísticos que transcienden a la mera aplicación al CDAM.

En el capítulo 5 se aplica el modelo TAP-M al diseño de aparcamientos disuasorios en los que los usuarios aparcan su coche y completan su viaje en transporte público. Se ha formulado un modelo binivel, denominado NDP-M, para la determinación de las tarifas y capacidades de estos aparcamientos. Se ha aplicado la técnica del recocido simulado para la resolución de este problema de diseño de redes continuo, realizando un estudio numérico de las características de este algoritmo y una ilustración del uso del NDP-M.

El capítulo 6 lo hemos dedicado al estudio del problema de diseño de intercambiadores multimodales urbanos en un contexto de planificación estratégica, donde se desea determinar su localización, su alimentación mediante una red secundaria y el diseño de sus aparcamientos disuasorios. Se han aplicado las técnicas heurísticas de los algoritmos golosos, del recocido simulado y técnicas de intercambio para resolver este problema.

La tesis doctoral se completa con otros dos capítulos, uno dedicado a introducir el campo de estudio y el otro a recoger las conclusiones y futuras líneas de investigación.

Palabras claves:

Programación convexa diferenciable, desigualdades variacionales, programación matemática binivel, descomposición simplicial, generación de columnas, convergencia finita, algoritmos heurísticos, modelos de equilibrio en redes multimodales con modos combinados, problemas de diseño de redes, estimación de matrices O-D, calibración de parámetros.

Abstract

The topic of the thesis is the development of methods and models in order to plan and design the urban transport networks. These models take into account the combined trips, that is, trips where the users choose two or more modes of transportation.

One important task has been the development of the State-of-Art methodology used to solve the traffic equilibrium assignment models, the so called simplicial decomposition method. For this aim, the CG/SD method has been defined, which improve significantly the efficiency of the classic simplicial decomposition. The new CG/SD method class may be used to improve the descent feasible methods used on the context of nonlinear programming.

The objectives have been the following:

(a) Modeling of the user's behavior of combined trips.

(b) Solving numerically these models by means of CG/SD methods.

(c) Estimation of their input parameters.

(d) Applications to transport network design problems.

The Thesis objectives (a) and (b) have been dealt in chapters $1, 2$ and 3 by means of the mathematical programming with only one level of decision. The objectives (c) and (d) are studied using the bilevel mathematical programming (BP) in chapters 4, 5 and 6. In the first part, the developed algorithms are exact. In the second part, due to the complexity and large scale problems, the algorithms are heuristic.

In Chapter 1, a traffic assignment equilibrium model with combined modes has been developed. The variational inequalities have been used for the mathematical formulation of this problem. Two simplicial decomposition methods have been developed to solve it and they have been computationally tested on symmetric networks. The applications of this model, named as TAP-M, is illustrated in the parametric design of urban multi modal interchange problems.

The study of the simplicial decomposition/column generation algorithms is motivated to be able to solve the TAP-M (symmetric case) efficiently, and by the contributions of Michael Patriksson in the CG/SD class.

The asymptotic convergence of the CG/SD class for the differentiable convex programming problems is proved in Chapter 2. The finite convergence is established under conditions based on the geometry of the problems, regularity conditions, and the properties of the generation column algorithm.

Chapter 3 deals with the numerical study of the CG/SD algorithms, which are realized on two test models: the TAP-M and the (uni-commodity) convex cost network flow problem with separable cost function. The parameters of the CG/SD algorithms and the prolongation of the columns to the relative frontier are analyzed to improve the efficiency of these methods. A computational comparative is effected among the classic simplicial decomposition, feasible descent direction methods and the new algorithms that are introduced by the CG/SD class. This comparative proves that the performance of CG/SD methods is significantly better than the classical methods.

Chapter 4 deals with the estimation of TAP-M inputs. That consists of an origin-destination matrix and a vector of parameters. This problem has been formulated using the bilevel mathematical programming, which is denoted as CDAM. The CDAM uses all available information about link counts on multi modal network, out of date matrices, and/or output of surveys. The *over specification* of the parameters has been studied. A class of heuristic algorithms has been developed to solve CDAM. The applicability of those is more general than solving the CDAM.

TAP-M is used in Chapter 5 in the design of the parking lots where the users can complete their journeys using one or more transit lines. This problem has been also formulated using the bilevel mathematical programming, which model is denoted as NDP-M. The NDP-M computes the optimal policy of parking fares and capacity. The simulated annealing technique has been used to solve this continuous network design problem. A numerical study of this method has been implemented. The use of the NDP-M is illustrated on an example.

Chapter 6 deals with the design of urban multi modal interchanges at the strategical planning. At this planning level the localization of the interchanges, the design of a secondary network to feed the interchanges and the design of facilities of parking are decided. The greedy, k-interchanges and simulated annealing algorithms have been used in order to solve the discrete bilevel model.

The thesis is completed with two additional chapters: an introductory chapter and one of conclusions and future research.

Key words:

Differentiable convex programming, variational inequalities, bilevel mathematical programming, simplicial decomposition, column generation, finite convergence, heuristic algorithms, traffic assignment equilibrium models with combined modes, network design problems, O-D matrix adjustment, calibration of parameters.

Introducción y sumario

En este capítulo introducimos los principales temas sobre los que trata esta tesis doctoral, que son la programación matemática, los modelos de asignación en equilibrio y la planificación del transporte urbano. El capítulo recoge el sumario de los trabajos de investigación que forman el cuerpo de la tesis doctoral.

1 Modelos y métodos en optimización de redes no lineales

1.1 El problema de optimización

Un problema de optimización viene caracterizado por tres elementos: *las variables del problema* que definen el conjunto de decisiones, la *función objetivo* que evalúa el coste o el beneficio de la decisión y el *conjunto de soluciones* que determina las decisiones válidas que pueden llevarse a cabo. Más formalmente, sea $f : S \mapsto \Re$ una función, X un subconjunto de S que se denomina *conjunto factible de soluciones*, entonces el problema de optimización (versión minimización) se formula

$$\begin{aligned} \text{minimizar} \quad & f(\mathbf{x}) \\ \text{sujeto a} \quad & \mathbf{x} \in X. \end{aligned}$$

La formulación anterior es demasiado general para que su estudio conduzca a métodos sastisfactorios de resolución, ya que S puede ser un conjunto cualquiera y por tanto esta formulación recoge la situación donde S es un espacio de dimensión infinita (por ejemplo un espacio de funciones). Este caso se denomina optimización de *funcionales* y la función f recibe (usualmente) el nombre de *función de energía*.

La forma de abordar el problema anterior es analizar situaciones particulares. Se van a exigir tanto a f como a X propiedades que sean suficientemente generales para poder ser utilizados en las aplicaciones en estudio, y lo suficientemente *fuertes* para obtener resultados de interés. La hipótesis más débil que suele exigirse a X es que sea un conjunto cerrado. En los modelos que estudiaremos asumiremos que el conjunto X es convexo, incluso en la mayoría de los casos tendremos propiedades aún más fuertes como la de ser un conjunto poliédrico (definido mediante restricciones lineales). Las condiciones para f son de dos tipos. Por un lado se exigen ciertas condiciones de regularidad local para poder caracterizar los extremos (mínimos) locales del problema y por otro, se exigen propiedades acerca del comportamiento global de la función, de modo que permitan garantizar que tales extremos locales son también extremos globales.

Para el primer grupo de hipótesis, las propiedades más empleadas son la F(réchet) - diferenciabilidad, la teoría de los *subdiferenciales* (subgradientes) (Rockafellar [204]) para funciones convexas y la teoría de los *gradientes generalizados* para funciones localmente lipschitzianas (Clarke [52]). Las propiedades globales más empleadas son la *convexidad* o una generalización de este concepto (bajo la hipótesis de F-diferenciabilidad) denominado *pseudoconvexidad* (Bazaraa y otros [12]). El interés se centra, en este trabajo, en los *problemas de optimización convexa diferenciable*. Mas formalmente, nosotros abordaremos el problema

$$\begin{array}{ll} \text{minimizar} & f(\mathbf{x}) \\ \text{sujeto a} & \mathbf{x} \in X, \end{array} \qquad [\text{CDP}(f, X)]$$

donde la función $f : \Re^n \mapsto \Re$ es continuamente diferenciable, pseudoconvexa (por ejemplo cuando es convexa) y el conjunto X es un subconjunto convexo de $\Re^n$. Las soluciones a este problema vienen caracterizada por el teorema siguiente

TEOREMA 1.1 (Condiciones de optimalidad de CDP(f, X)). *El punto $\mathbf{x}^* \in X$ es una solución óptima de CDP(f, X) si y sólo si cumple*

$$\nabla f(\mathbf{x}^*)^T(\mathbf{x} - \mathbf{x}^*) \geq 0, \quad \forall \mathbf{x} \in X. \tag{1}$$

donde ∇f es el gradiente de la función f. (Ver, por ejemplo, el teorema 3.3.4 de Bazaraa y otros [12])

Las condiciones anteriores pueden ser reformuladas cuando el conjunto X está definido explícitamente por $X = \{\mathbf{x} \in \Re^n : g_i(\mathbf{x}) \leq 0,\ i \in \mathcal{I} \text{ y } h_j(\mathbf{x}) = 0,\ j \in \mathcal{J}\}$. Supongamos que las funciones g_i con $i \in \mathcal{I}$ son funciones convexas y que todas son diferenciables en $\Re^n$ y que las funciones h_j con $j \in \mathcal{J}$ son funciones afines . La condición de optimalidad (1) es equivalente (asumiendo alguna cualificación de las restricciones) a las condiciones de Karush-Kuhn-Tucker. Un punto $\mathbf{x}^*$ es una solución óptima para el CDP(f, X) si y sólo si el sistema de ecuaciones

$$\begin{aligned} \nabla f(\mathbf{x}^*) + \sum_{i \in \mathcal{I}} \mu_i \nabla g_i(\mathbf{x}^*) + \sum_{j \in \mathcal{J}} \lambda_j \nabla h_j(\mathbf{x}^*) &= \mathbf{0}, \\ g_i(\mathbf{x}^*)\mu_i = 0, \quad i \in \mathcal{I}, \\ g_i(\mathbf{x}^*) \leq 0, \quad i \in \mathcal{I}, \\ h_j(\mathbf{x}^*) = 0, \quad j \in \mathcal{J}, \\ \mu_i \geq 0, \quad i \in \mathcal{I}, \end{aligned}$$

tiene solución.

Métodos de optimización

El objetivo de hacer algoritmos eficientes para problemas reales (de gran tamaño y de estructura compleja) ha hecho que éstos sean altamente especializados y focalizados al problema que tratan de resolver. Es por ese motivo por lo que pensar en desarrollar algoritmos que sean eficientes para todos los problemas de programación matemática convexa diferenciable es inviable. Esto ha conducido a los investigadores a reducir la clase de aplicaciones a abordar con los algoritmos desarrollados.

Una primera reducción considera exclusivamente problemas convexos con restricciones lineales, pero aún así esta clase sigue siendo demasiado extensa. Una segunda reducción supone que este conjunto de restricciones lineales tiene estructura de red, lo que sería un de problema de flujos en redes. Una buena monografía sobre el tema es la dada por Bertsekas [20]. La reducción de la clase de problemas es todavía mayor en la literatura científica, y muchos de estos algoritmos están diseñados para resolver una aplicación concreta. Es por esto por lo que en este trabajo abordamos los modelos de redes en equilibrio para la asignación de tráfico. Estos modelos son formulados mediante un problema de programación convexa diferenciable no lineal sujeto a restricciones de red (un problema de flujos en redes multiproducto). Pese al esfuerzo en su acotación, el campo de estudio es aún demasiado extenso. Prueba de ello son las más de mil referencias bibliográficas sobre el *problema de asignación de tráfico* recogidas en Patriksson [196].

Patriksson [196] elabora una taxonomía de los algoritmos desarrollados para este problema de acuerdo a la combinación de tres principios algorítmicos:

1. Algoritmos de linealización parcial (Patriksson [193, 194, 195]).

   Es una clase de algoritmos de direcciones factibles de descenso que obtiene la dirección de búsqueda resolviendo un problema auxiliar convexo y diferenciable, definido por una aproximación a la función objetivo original. Más formalmente, consideran una función $\phi : X \times X \mapsto \Re$, donde $\phi(\mathbf{x}, \mathbf{y})$ es convexa y continuamente diferenciable en la variable $\mathbf{x}$ y continua respecto la variable $\mathbf{y}$ en el conjunto X. La función original se puede expresar

   $$f(\mathbf{x}) = \phi(\mathbf{x}, \mathbf{y}) + [f(\mathbf{x}) - \phi(\mathbf{x}, \mathbf{y})] .$$

   El segundo término es el error obtenido al reemplazar la función objetivo $f(\mathbf{x})$ por la aproximación $\phi(\mathbf{x}, \mathbf{y})$ en el punto $\mathbf{y}$. Los métodos de linealización parcial consideran una aproximación lineal de dicho error, lo que conduce a la aproximación de la función objetivo

   $$\tilde{f}(\mathbf{x}) = \phi(\mathbf{x}, \mathbf{y}) + f(\mathbf{y}) - \phi(\mathbf{y}, \mathbf{y}) + [\nabla f(\mathbf{y}) - \nabla_{\mathbf{x}} \phi(\mathbf{y}, \mathbf{y})]^T (\mathbf{x} - \mathbf{y}) .$$

   El subproblema para obtener la dirección de búsqueda en el punto $\mathbf{y}$ es

   $$\begin{array}{ll} \text{minimizar} & \tilde{f}(\mathbf{x}) \\ \text{sujeto a} & \mathbf{x} \in X, \end{array}$$

   y la dirección (de descenso) es entonces definida por $\mathbf{d} = \tilde{\mathbf{x}} - \mathbf{y}$, donde $\tilde{\mathbf{x}}$ es una solúcion óptima del problema anterior de optimización. El procedimiento se completa efectuando una búsqueda unidimensional en la dirección $\mathbf{d}$.

2. Métodos de generación de columnas (variables)

   Esta clase de métodos resuelve iterativamente dos problemas de optimización. El primero, denominado *problema maestro restringido* (RMP), es una aproximación al problema original construida mediante la sustitución de la región factible original por un subconjunto compacto (*aproximación interior*). Esta aproximación interior es mejorada mediante la generación de una nueva columna en la región factible, a través de la solución de otra aproximación al problema original, donde la función objetivo es aproximada. Esta fase es denominada *problema de generación de la columna* (CGP).

3. Algoritmos de descomposición.

   El conjunto factible del problema de asignación de tráfico tiene estructura de producto cartesiano. Esta estructura se aprovecha para transformar los subproblemas, que aparecen en los métodos de generación de columnas o de linealización parcial, en una secuencia de problemas de menor dimensión. Éstos pueden ser resueltos secuencialmente o mediante técnicas de computación paralela.

El algoritmo más conocido y aplicado es el de Frank-Wolfe [88]. Este algoritmo es un ejemplo de algoritmo de linealización parcial donde el subproblema de búsqueda de direcciones es obtenido eligiendo $\phi(\mathbf{x}, \mathbf{y}) = 0$. Éste es un problema lineal, conocido con el nombre de *problema de caminos mínimos* y puede ser resuelto eficientemente. En este subproblema se aprovecha la estructura de producto cartesiano para separar el cálculo de los caminos de coste mínimo por pares de origen-destino, por un origen-todos los destinos, etc.

El algoritmo de Frank-Wolfe la base de numerosos programas de ordenador empleados en las aplicaciones, por ejemplo TRAFFIC de Nguyen y James [180]. Quizás la primera implementación convergente del algoritmo de Frank-Wolfe, en aplicaciones de transporte, sea debida a Eash y otros [70] quienes la desarrollaron para el estudio del área de Chicago ([42]). Una versión no convergente fue realizada en el proyecto UTPS (Urban Transportation Planning System) [230] debido a que se empleaba una aproximación al problema de búsquda lineal y se aplicó en 1978 al área de los Angeles.

Se han sugerido numerosas mejoras del algoritmo de Frank-Wolfe como son las búsqueda lineales de Wolfe [238], la implementación de las tangentes paralelas PARTAN (LeBlanc y otros [146], las

búsquedas lineales inexactas de Marín [164]) o la modificación de la longitud de paso de Weintraub y otros [235].

Otros ejemplos de algoritmos de linealización parcial son los algoritmos de tipo Newton donde la función ϕ es $\phi(\mathbf{x},\mathbf{y}) = 1/2(\mathbf{x}-\mathbf{y})\mathbf{B}(\mathbf{y})(\mathbf{x}-\mathbf{y})$, siendo $\mathbf{B}(\mathbf{y})$ una aproximación a la matriz Hessiana de f en $\mathbf{y}$. Ciertas elecciones de $\mathbf{B}(\mathbf{y})$ conducen a los denominados algoritmos de *punto próximo* (Rockafellar [205]), *proyección del gradiente* (Polyak [201]) o *Newton restringido* (Polyak [201]). Dembo y Tulowitzki [65] aplican un método truncado de Newton, donde los subproblemas cuadráticos son resueltos aproximadamente mediante el algoritmo de Frank-Wolfe o su versión PARTAN. Este algoritmo llega a reducir el tiempo de computación hasta en un 60% comparado con el algoritmo Frank-Wolfe o su versión PARTAN.

Los algoritmos de tipo *Jacobi / Gauss-Seidel* son ejemplos de algoritmos de descomposición, éstos consideran la estructura de producto cartesiano de la región factible, es decir, $X = \prod_{i\in\mathcal{C}} X_i$ donde X_i es una copia de la red y hay tantas como productos distintos sean considerados. Cada X_i está asociado a los flujos en un producto. El problema original no es separable por productos debido a que la función objetivo no lo es.

Los métodos tipo Jacobi descomponen el problema original en tantos subproblemas (independientes) como número de productos sean transportados por la red. El $i-$ésimo subproblema se obtiene fijando todas las variables exceptuando la asociada al factor X_i, y entonces se resuelve dicho problema en la región de factibilidad X_i. Las soluciones de todos los subproblemas conducen a un nuevo punto donde reiniciar el proceso.

Los métodos de tipo Gauss/Seidel operan como los métodos de Jacobi pero en lugar de descomponer el problema original simultáneamente por productos lo hacen iterativamente producto por producto. Algunas aplicaciones de estos métodos al problema de asignación de tráfico vienen recogidas en Dafermos [57, 58], Dafermos y Sparrow [61], Petersen [200], Serra y Weintraub [209]

El algoritmo má conocido de generación de columnas en optimización no lineal es la *descomposición simplicial* (SD). La forma clásica de este método fue descrita por Holloway [128] y Von Hohenbalken [127] que la aplicaron a problemas restringidos linealmente. En este algoritmo los CGP son obtenidos linealizando la función objetivo, lo que conduce a problemas lineales. La región factible del RMP está definida como la envoltura convexa de un subconjunto de los puntos extremos generados anteriormente (columnas). Este método constituye una extensión del algoritmo de Frank-Wolfe.

Hearn y otros [123, 125] introducen un parámetro r en el algoritmo SD para limitar el número de puntos extremos retenidos en el RMP. Este método modifica las reglas de eliminación de columnas del SD para limitar a r el número máximo de puntos extremos retenidos en el RMP. Este algoritmo se denomina *descomposición simplicial restringida* (RSD).

Marín [167] especializa el RSD para problemas con restricciones laterales. En este algoritmo el problema maestro se obtiene como la intersección de un símplice con el conjunto definido por las restricciones laterales. El subproblema lineal es modificado para que la función objetivo recoja una estimación de los multiplicadores de Lagrange de las restricciones laterales.

Larsson y Patriksson [140] desarrollan la denominada *descomposición simplicial desagregada* (DSD) para problemas restringidos linealmente con estructura de producto cartesiano, la diferencia respecto al SD está en la formulación del RMP. En este método la envoltura convexa se toma producto a producto, obteniendo la región factible como un producto cartesiano de símplices.

Larsson y otros [154, 141] introducen la denominada *descomposición simplicial no lineal* (NSD). El método NSD se obtiene reemplazando del RSD el problema lineal en el CGP por uno no lineal y convexo. La introducción de este tipo de problemas (no lineales) hace que las soluciones ya no sean (necesariamente) puntos extremos y por tanto no están situadas en la frontera de X. Es por esto que estos autores prolongan las columnas generadas a la frontera (relativa).

La descomposicón simplicial también ha sido aplicada a problemas con restricciones no lineales. Ventura y Hearn [232] extienden el RSD a problemas con resticciones covexas (RSDCC). El CGP es transformado a lineal mediante la linealización de las restricciones, tal y como lo hace el esquema de Topkis-Veinott [228]. Por otro lado la NSD de Larsson y otros [154, 141] es aplicada directamente

a un problema cuyo conjunto factible es convexo. Una combinación de la *programación cuadrática secuencial* (ver Bazaraa y otros [12]) y el NSD es desarrollado en Patriksson [197]. En este algoritmo el subproblema del CGP se obtiene linealizando las restricciones no lineales y teniendo en cuenta en la función objetivo la curvatura de estas restricciones.

Esta clase de métodos también ha sido extendida a problemas convexos *no diferenciables* Larsson y otros [139, 142].

La realización de pruebas numéricas con los anteriores algoritmos para problemas de tráfico ha sido descrita en varios trabajos. Posiblemente sean los trabajos de Montero [173], Montero y Barceló [174] los que analizan más exhaustivamente las diversas posibilidades de implementación de los anteriores métodos.

1.2 Modelos de desigualdades variacionales

En esta sección introducimos los modelos de desigualdades variacionales que nos permiten abordar situaciones más generales que las posibilitadas por la programación convexa diferenciable. Formalmente este problema se plantea del siguiente modo. Se considera un conjunto cerrado y convexo $X \subset \Re^n$ y una función $\mathbf{F} : X \mapsto \Re^n$ continua en X. El problema de desigualdades variacionales consiste en encontrar un $\mathbf{x}^* \in X$ cumpliendo

$$\mathbf{F}(\mathbf{x}^*)^T(\mathbf{x} - \mathbf{x}^*) \geq 0, \quad \forall \mathbf{x} \in X \qquad [\mathrm{VIP}(\mathbf{F}, X)]$$

Este problema también es conocido con los nombres de *ecuación generalizada* o de *problema de punto estacionario.*

Han sido caracterizados varios conjuntos de hipótesis que garantizan la existencia de soluciones a este problema. Por ejemplo bajo la suposición de que X es un conjunto compacto (Hartman y Stampacchia [120], Brézis [32]) o bajo la hipótesis de que la aplicación $\mathbf{F}$ es *coercitiva* (Hartman y Stampacchia [120], Moré [175]).

Si la aplicación $\mathbf{F}$ es *fuertemente monótona* entonces el problema tiene solución (Stampacchia [222]), si además $\mathbf{F}$ es *pseudomonótona* entonces el conjunto de soluciones, $\mathrm{SOL}(\mathbf{F}, X)$, es un conjunto convexo, y bajo la *cualificación de restricciones de Slater* o la hipóteis de coercitividad de $\mathbf{F}$ el conjunto de soluciones está acotado. Se pude demostrar la unicidad de la solución bajo la hipótesis de que es *estrictamente monótona* (Stampacchia [222]).

Las condiciones de optimalidad (1) muestran que el problema $\mathrm{CDP}(f, X)$ puede ser formulado como un problema de desigualdades variacionales $\mathrm{VIP}(\nabla f, X)$. Cabría plantearse estudiar el camino inverso, es decir, deteminar bajo qué condiciones la función $\mathbf{F}$ es un gradiente, esto es, $\mathbf{F} = \nabla f$. Una condición suficiente (teorema 4.1.6 Ortega y Rheinboldt [188]) es la siguiente:

TEOREMA 1.2 (Condición suficiente para que $\mathbf{F} = \nabla f$). *Sea $\mathbf{F} : X \mapsto \Re^n$ de clase C^1 sobre un conjunto convexo y abierto $X_0 \subset X$. Entonces $\mathbf{F}$ es el gradiente de una aplicación en X_0 si y sólo si $\nabla \mathbf{F}(\mathbf{x})$ es simétrica para todo $\mathbf{x} \in X_0$.*

Bajo la condición de simetría, la integral definida por

$$f(\mathbf{x}) = \int_0^1 \sum_{j=1}^{n} F_j\left(\hat{\mathbf{x}} + s(\mathbf{x} - \hat{\mathbf{x}})\right)(x_j - \hat{x}_j)ds \tag{2}$$

donde $\hat{\mathbf{x}}$ es un elemento arbitrario de X_0, es independiente del camino elegido, y de acuerdo al teorema de Green, la función $\mathbf{F}$ es integrable. En este caso el problema $\mathrm{VIP}(\mathbf{F}, X)$ puede ser planteado como un programa matemático cuya función objetivo es f. El siguiente teorema muestra las relaciones entre $\mathbf{F} = \nabla f$ y la función f.

TEOREMA 1.3 (Relaciones entre la monotonía de f y de $\mathbf{F}$). *Sea $\mathbf{F} \equiv \nabla f$. Entonces se cumple:*

1. $\mathbf{F}$ *es monótona en* $X \Leftrightarrow f$ *es convexa en* X.
2. $\mathbf{F}$ *es estrictamente monótona en* $X \Leftrightarrow f$ *es estrictamente convexa en* X.
3. $\mathbf{F}$ *es fuertemente monótona en* $X \Leftrightarrow f$ *es fuertemente convexa en* X.

Las demostraciones de estas relaciones se pueden consultar en Ortega y Rheinboldt [188].

Métodos de resolución del problema de desigualdades variacionales

Los problemas de desigualdades variacionales están íntimamente relacionados con los problemas de optimización. Tanto es así que Patriksson [197] elabora una teoría unificada para estudiar los algoritmos y su convergencia de ambos tipos de problemas.

Los tres principios usados en la elaboración de algoritmos para los modelos de optimización son también empleados en los problemas de desigualdades variacionales. La clase de linealización parcial es extendida a la llamada *clase de aproximación de costes* Patriksson [195]. Estos algoritmos constan de dos fases

(a) Dado un punto del dominio, se construye un nuevo problema de desigualdades variacionales reemplazando la aplicación de costes $\mathbf{F}$ mediante una aproximación monótona, la solución (truncada) a este problema define la dirección factible de búsqueda. Más formalmente, se considera la reescritura de $\mathbf{F}(\cdot) = \Phi(\cdot) + [\mathbf{F}(\cdot) - \Phi(\cdot)]$ donde Φ es una aproximación monótona de $\mathbf{F}$, la función de costes del subproblema de desigualdades variacionales se obtiene fijando el segundo término al valor $\mathbf{x}$, esto es $\tilde{\mathbf{F}}(\mathbf{y}) = \Phi(\mathbf{y}) + \mathbf{F}(\mathbf{x}) - \Phi(\mathbf{x})$. El subproblema para obtener la dirección de búsqueda consiste en encontrar un $\mathbf{y}^* \in X$ cumpliendo

$$\tilde{\mathbf{F}}(\mathbf{y})^T(\mathbf{y} - \mathbf{y}^*) = [\Phi(\mathbf{y}) + \mathbf{F}(\mathbf{x}) - \Phi(\mathbf{x})]^T(\mathbf{y} - \mathbf{y}^*) \geq 0, \quad \forall \mathbf{y} \in X \qquad [\mathrm{VIP}(\tilde{\mathbf{F}}, X)]$$

(b) La segunda fase es equivalente a la búsqueda unidimensional de los algoritmos de linealización parcial, para ello se reformula el problema de desigualdades variacionales mediante un problema de optimización a través de las llamadas *funciones de salto*. Esta técnica es aplicable incluso en el caso de que el Jacobiano de la función de costes no sea simétrico.

Una función $\psi : X \mapsto \Re \cup \{-\infty, +\infty\}$ es una función de salto si cumple que ψ está restringida en signo y $\psi(\mathbf{x}^*) = 0$ si y sólo si $\mathbf{x}^*$ es una solución a la desigualdad variacional $\mathrm{VIP}(\mathbf{F}, X)$. Estas funciones dan una medida del incumplimiento del problema $\mathrm{VIP}(\mathbf{F}, X)$ en cada punto de $\mathbf{x} \in X$.

El siguiente problema de optimización (suponiendo que ψ es no negativa sobre X)

$$\begin{aligned} \text{minimizar} \quad & \psi(\mathbf{x}) \\ \text{sujeto a} \quad & \mathbf{x} \in X, \end{aligned}$$

es equivalente al problema $\mathrm{VIP}(\mathbf{F}, X)$. Esta equivalencia hace que la búsqueda unidimensional sea realizada sobre las funciones ψ que juegan el papel de las llamadas *funciones de mérito*. La longitud del paso es elegida de modo que ψ sea reducida suficientemente.

Los *algoritmos de linealización* (ver Pang y Chan [191]) son ejemplos de la clase de algoritmos de aproximación de costes. Estos algoritmos son obtenidos eligiendo como aproximación de $\mathbf{F}$ la fuunción $\Phi(\cdot) = \mathbf{A}(\mathbf{x})(\cdot - \mathbf{x})$, donde $\mathbf{A}(\mathbf{x})$ es una matriz semidefinida positiva en $\Re^{n \times n}$, esta elección conduce a la aplicación de costes afín $\tilde{\mathbf{F}}(\mathbf{y}) = \mathbf{F}(\mathbf{x}) + \mathbf{A}(\mathbf{x})(\mathbf{y} - \mathbf{x})$. En el contexto de asignación de tráfico se han hecho varias elecciones de la matriz $\mathbf{A}(\mathbf{x})$, el caso $\mathbf{A}(\mathbf{x}) = \frac{1}{\gamma}\mathbf{G}$ es conocido con el nombre de *método de proyección* y ha sido aplicado en Dafermos [59], Fisk y Nguyen [76], Harker [119]. En el caso de que la matriz $\mathbf{A}$ sea simétrica el suproblema variacional afín es equivalente (en el contexto de asignación de tráfico) a un problema cuadrático de flujos en redes multiproducto. Esto ha conducido a elegir como

$\mathbf{A}(\mathbf{x})$ varias aproximaciones simétricas de la matriz Jacobiana de $\mathbf{F}$, esto es $\mathbf{A}(\mathbf{x}) \approx \nabla\mathbf{F}(\mathbf{x})$, son los llamados *métodos quasi-Newton*, por ejemplo $\mathbf{A}(\mathbf{x}) = 1/2(\nabla\mathbf{F}(\mathbf{x}) + \nabla\mathbf{F}(\mathbf{x})^T)$ o $\mathbf{A}(\mathbf{x}) = \mathbf{D}(\mathbf{x})$ donde $\mathbf{D}(\mathbf{x})$ es la diagonal de la matrix Jacobiana. Este último método se conoce con el nombre de *algoritmo de Jacobi linealizado*. Algunos ejemplos de estos métodos aplicados al problema del asignación de tráfico se dan en Dafermos [58], Fisk y Nguyen [76].

La clase de métodos tipo Jacobi/Gauss-Seidel también han sido empleados para los problemas de desigualdades variacionales y se conocen como métodos de *diagonalización* o *relajación*, los cuales consisten en considerar aproximaciones de la función de coste cuya matriz Jacobiana sea diagonal y por tanto relajar las interacciones entre las variables del problema de esta manera, la desigualdad variacional es equivalente a un problema de optimización (ver el teorema 1.2) o a una secuencia de éstos que pueden ser resueltos con algoritmos de optimización.

Ejemplos de métodos tipo Jacobi aplicados al problema de asignación de tráfico son analizados en Abdulaal y LeBlanc [4], Sheffi [210], Harker [119], Mahmassani y Mouskos [156]. Sheffi [210] recomienda emplear una única iteración del método de Frank-Wolfe para resolver el subproblema de optimización y Mahmassani y Mouskos [156] emplean cuatro iteraciones en cada subproblema.

Los métodos de generación de columnas también han sido aplicados a problemas de desigualdades variacionales donde X es un conjunto poliedral. Lawphongpanich y Hearn [144] extiende la descomposición simplicial restringida al problema VIP $(\mathbf{F}, X)$. El RMP es formulado mediante una desigualdad variacional consistente en resolver la función de coste original definida sobre la envoltura convexa de los puntos extremos previamente generados. En el contexto de desigualdades variacionales se debe tener mayor cuidado para definir las reglas de eliminación de columnas, debido a que puede producirse el *ciclaje* y perder la convergencia. Hammond [115], Magnanti [155] dan contra-ejemplos a la convergencia de la descomposición simplicial restringida. Smith [214], Pang y Yu [192] mantienen en el problema maestro todos los puntos extremos generados anteriormente, mientras que Lawphongpanich y Hearn [144] eliminan columnas cuando se obtiene un suficiente descenso de una función de mérito. Un análisis general de la convergencia bajo reglas generales de eliminación de columnas y descenso en funciones de mérito es analizado en Patriksson [197].

Los métodos linealizados de Jacobi, en combinación con descomposición simplicial/ generación de columnas son considerados en Pang y Yu [192], Bertsekas y Gafni [21], Lawphongpanich y Hearn [144]. Algoritmos tipo Newton, en combinación con descomposición simplicial/generación de columnas, son presentados en Aashtiani [1], Aashtiani y Magananti [2], Lawphongpanich y Hearn [144], Hearn y otros [124]. Una revisión de estos métodos es analizada en Patriksson [197].

1.3 Programación matemática con restricciones de equilibrio (MPEC)

Un programa matemático con restricciones de equilibrio (MPEC) es un modelo de optimización en el que cierto conjunto de restricciones están definidas mediante una desigualdad variacional. En este problema se distinguen dos problema anidados: el de optimización, que recibe el nombre de *problema exterior o problema del nivel superior*, y el problema de la desigualdad variacional, que se le denomina *problema interior o problema del nivel inferior*. El término de *restricciones de equilibrio* hace referencia a que la desigualdad variacional modeliza ciertos equilibrios que aparecen en problemas de economía e ingeniería.

Este modelo considera dos conjuntos de variables, que denotaremos $\mathbf{x} \in \Re^n$ e $\mathbf{y} \in \Re^m$. Las variables $\mathbf{x}$ parametrizan una desigualdad variacional cuya solución define los valores de la variable $\mathbf{y}$. Las variables $\mathbf{x}$ reciben el nombre *variables del nivel superior* y las variables $\mathbf{y}$ el de *variables del nivel inferior*.

El MPEC puede ser formulado del siguiente modo. Se consideran dos funciones $f : \Re^{n+m} \mapsto \Re$ y $\mathbf{F} : \Re^{n+m} \mapsto \Re^m$, un conjunto cerrado y convexo $Z \subset \Re^{n+m}$ y una aplicación punto-conjunto $\Omega : \Re^{n+m} \mapsto \Re^m$ cuyas imágenes son conjuntos cerrados y convexos, es decir, para cada valor $\mathbf{x} \in \Re^n$, $\Omega(x)$, es un subconjunto cerrado y convexo de $\Re^m$. La función f es la *función objetivo* del problema de optimización; $\mathbf{F}$ es la *función de costes (equilibrio)* de la desigualdad variacional, Z es la región factible del problema de optimización para el par $(\mathbf{x}, \mathbf{y})$ y el conjunto $\Omega(\mathbf{x})$ define la región factible

para el problema de la desigualdad variacional.

El problema MPEC se formula por

$$\begin{array}{ll} \text{minimizar} & f(\mathbf{x},\mathbf{y}) \\ \text{sujeto a} & (\mathbf{x},\mathbf{y}) \in Z, \\ & \mathbf{y} \in \mathcal{S}(\mathbf{x}), \end{array} \qquad [\text{MPEC}]$$

donde $\mathcal{S}(\mathbf{x})$ es el conjunto de soluciones de la desigualdad variacional definida por el par $(\mathbf{F}(\mathbf{x},\cdot),\Omega(\mathbf{x}))$, es decir, el vector $\mathbf{y}$ pertenece al conjunto $\mathcal{S}(\mathbf{x})$ si y sólo si $\mathbf{y}$ es un elemento de $\Omega(\mathbf{x})$ que cumple la desigualdad

$$\mathbf{F}(\mathbf{x},\mathbf{y})^T(\mathbf{v}-\mathbf{y}) \geq 0, \quad \forall \mathbf{v} \in \Omega(\mathbf{x}) \qquad [\text{VIP}(\mathbf{F}(\mathbf{x},\cdot),\Omega(\mathbf{x}))]$$

La formulación de MPEC recoge numerosos casos particulares de interés. Uno especialmente importante y que puede ser considerado como la formulación predecesora del MPEC, aparece cuando $\mathbf{F}(\mathbf{x},\cdot)$ es el gradiente respecto a la variable $\mathbf{y}$ de cierta función $\theta : \Re^{n+m} \mapsto \Re$ de clase C^1, es decir, $\mathbf{F}(\mathbf{x},\mathbf{y}) = \nabla_{\mathbf{y}}\theta(\mathbf{x},\mathbf{y})$ donde $\nabla_{\mathbf{y}}$ denota la derivada parcial con respecto a la variable $\mathbf{y}$. En este caso las soluciones de la desigualdad variacional $\text{VIP}(\mathbf{F}(\mathbf{x},\cdot),\Omega(\mathbf{x}))$ coinciden con los puntos estacionarios del siguiente problema de optimización

$$\begin{array}{ll} \text{minimizar} & \theta(\mathbf{x},\mathbf{y}) \\ \text{sujeto a} & \mathbf{y} \in \Omega(\mathbf{x}). \end{array} \qquad [\text{LLP}(\mathbf{x})]$$

Cuando en la formulación del MPEC la restricción $\mathbf{y} \in \mathcal{S}(\mathbf{x})$ es reemplazada por

$$\mathbf{y} \in \arg\,\text{minimizar}\{\theta(\mathbf{x},\mathbf{y}) : \mathbf{y} \in \Omega(\mathbf{x})\},$$

donde "arg minimizar" denota el conjunto de soluciones del problema de optimización, entonces el problema MPEC recibe el nombre de *problema de optimización binivel*, $\text{LLP}(\mathbf{x})$ se denomina *problema del nivel inferior* y $\min_{(\mathbf{x},\mathbf{y})\in Z} f(\mathbf{x},\mathbf{y})$ se denomina *problema del nivel superior*. El MPEC recibe también el nombre de *problema binivel generalizado*.

Notar que cuando el conjunto $\Omega(\mathbf{x})$ es convexo se obtiene

$$\arg\,\text{minimizar}\{\theta(\mathbf{x},\mathbf{y}) : \mathbf{y} \in \Omega(\mathbf{x})\} \subset \mathcal{S}(\mathbf{x}).$$

La igualdad se cumple cuando la función $\theta(\mathbf{x},\cdot)$ es una función convexa en el segundo argumento, por el teorema 1.1.

La programación matemática binivel es un caso particular de la programación matemática jerarquizada que considera varios niveles de decisión (posiblemente más de dos). Un ejemplo importante en la toma de decisiones en dos niveles es el llamado *juego de Stackelberg* (Stackelberg [221], Baar y Olsder [9]). Este juego ha sido estudiado intensivamente por economistas y han encontrado amplias aplicaciones como el análisis de mercados oligopolísticos (Okuguchi [186], Sherali y otros [211]), diseño de nuevos productos (Choi y otros [51]), tarifación de la transmisión de la energía eléctrica (Hobbs y Kelly [126]), etc.

En el juego de Stackelberg existe un jugador especial, denominado *líder* que puede conocer las (re)acciones del resto de los jugadores a su estrategia, el resto de judadores se denominan *seguidores*. El líder puede elegir su estrategia dentro de un conjunto $X \subset \Re^n$, mientras que cada seguidor (supongamos el i) puede elegir una del conjunto de estrategias $Y_i(\mathbf{x}) \subset \Re^{m_i}$, suponiendo que el líder ha elegido la estrategia $\mathbf{x}$. Este conjunto se asume que es cerrado y convexo. La función de coste para el jugador i es $\theta_i(\mathbf{x},\cdot) : \prod_{j=1}^{M} X_j \mapsto \Re$, donde M es el número de seguidores. Notar que cada estrategia de un seguidor depende de la estrategia $\mathbf{x}$ del líder, además su coste depende tanto de las estrategias de los otros seguidores como de la del líder.

Supongamos que para cualquier vector $\hat{\mathbf{x}} \in X$ e $\hat{\mathbf{y}}_{\neq i} \equiv (\hat{\mathbf{y}}_j : j \neq i)$ la función

$$\theta_i(\hat{\mathbf{x}}, \mathbf{y}_i, \hat{\mathbf{y}}_{\neq i}) \tag{3}$$

es convexa y continuamente diferenciable en la variable $\mathbf{y}_i \in Y_i(\hat{\mathbf{x}})$. Los seguidores elegirán para cada $\mathbf{x} \in X$ una reacción conjunta

$$\mathbf{y}^* \equiv (\mathbf{y}_i^*)_{i=1}^M \in \prod_{i=1}^M Y_i(\mathbf{x}),$$

de modo que para todo $i = 1, \ldots, M$ se cumple

$$\mathbf{y}_i^* \in \arg \text{ minimizar}\{\theta_i(\mathbf{x}, \mathbf{y}_i, \mathbf{y}_{\neq i}^*) : \mathbf{y}_i \in Y_i(\mathbf{x})\}. \tag{4}$$

Por la convexiad de las *funciones de cada jugador* del juego (3) y de los conjuntos $Y_i(\mathbf{x})$ tendremos, por el teorema 1.1, que $\mathbf{y}^*$ cumple (4) para todo valor de $i = 1, \ldots, M$ si y sólo si $\mathbf{y}^*$ es solución de la desigualdad variacional VIP$(\mathbf{F}(\mathbf{x}, \cdot), \Omega(\mathbf{x}))$ donde $\mathbf{F}(\mathbf{x}, \mathbf{y}) \equiv (F_i(\mathbf{x}, \mathbf{y}))_{i=1}^M$ siendo

$$F_i(\mathbf{x}, \mathbf{y}) = \nabla_{\mathbf{y}_i} \theta_i(\mathbf{x}, \mathbf{y}), \quad i = 1, \ldots, M \qquad \text{y} \qquad \Omega(\mathbf{x}) = \prod_{i=1}^M Y_i(\mathbf{x}).$$

Dada la función $f : \Re^{n+p} \mapsto \Re$ de coste del líder, donde $p = \sum_{i=1}^M m_i$, que depende tanto de su propia estrategia como de la de los seguidores, el líder elegirá su estrategia de modo que minimice su coste. Esto conduce a la siguiente formulación del juego de Stackelberg

$$\begin{array}{ll} \text{minimizar} & f(\mathbf{x}, \mathbf{y}) \\ \text{sujeto a} & \mathbf{x} \in X, \\ & \mathbf{y} \in \text{ soluciones de } VIP(\mathbf{F}(\mathbf{x}, \cdot), \Omega(\mathbf{x})). \end{array} \tag{5}$$

En esta sección se ha puesto de manifiesto como el MPEC puede modelizar los juegos de Stackelberg que aparecen en la planificación del transporte urbano.

Complejidad del MPEC y algoritmos de resolución

El MPEC es extremadamente difícil de resolver. Este hecho se deriva de la complejidad de su conjunto factible, que denotamos por $\mathcal{F}$, pudiendo ser debida a diversas causas:

- ◇ No convexidad de $\mathcal{F}$. La región factible puede dejar de ser convexa, aunque todas las funciones y conjuntos que intervengan en su definición lo sean.
- ◇ No es un conjunto cerrado. El conjunto factible puede ser incluso no cerrado. Esta propiedad puede hacer peligrar hasta la existencia de soluciones.
- ◇ Naturaleza multievaluada de la función $\mathcal{S}(\mathbf{x})$. En muchas aplicaciones $\mathcal{S}(\mathbf{x})$ es un conjunto que contiene más de un elemento para cada $\mathbf{x}$.
- ◇ No diferenciabilidad de $\mathcal{S}(\mathbf{x})$. La situación más favorable es que la aplicación multievaluada $\mathcal{S}(\mathbf{x})$ defina una aplicación (tenga un único elemento cada conjunto), en este caso dicha función puede ser no diferenciable.
- ◇ Pérdida de la propiedad de conexidad. La región factible puede ser la unión de varios conjuntos disjuntos. Esta propiedad le confiere una naturaleza combinatoria, con *restricciones disyuntivas*, que son de gran complejidad computacional, incluso en problemas lineales.

Se podría pensar que estas patologías están presentes exclusivamente en situaciones excepcionales, y que la exigencia de ciertas propiedades sobre las funciones y conjuntos que definen el MPEC podrían evitarlas. Pero lo cierto es que en el caso menos exigente de *programación lineal binivel*, donde todas las funciones son indefinidamente diferenciables y convexas, y todos los conjuntos son poliedros, aparecen todas las anteriores patologías. Además, Jeroslow [133], Hansen y otros [118] han demostrado que el

problema lineal binivel pertenece a la clase de problemas *NP-duros*, es decir, que no existe actualmente un algoritmo que pueda resolver el problema en un número polinomial de operaciones en función de las dimensiones del mismo.

El siguiente problema binivel en $\Re^2$ tomado de Luo y otros [154] ilustra que $\mathcal{F}$ no tiene que ser necesariamente un conjunto convexo, y la función $\mathcal{S}(\mathbf{x})$ no es necesariamente diferenciable.

$$\begin{array}{ll} \text{minimizar} & f(x,y) \\ \text{sujeto a} & x \geq 0, \\ & y \in \text{ arg minimizar}\{y : y \in \Omega(x)\} \end{array}$$

donde

$$\Omega(x) = \{y \in \Re_+ : x + 2y \geq 10,\ x - 2y \leq 6,\ 2x - y \leq 21,\ x + 2y \leq 38,\ -x + 2y \leq 18\}$$

El conjunto $\Omega(x)$ es no vacío si y sólo si $x \leq 16$. Por tratarse de un problema de programación lineal se puede resolver de forma cerrada para cada $x \in [0, 16]$, obteniendo

$$\mathcal{S}(x) = \begin{cases} 5 - x/2, & \text{si } x \in [0, 8], \\ -3 + x/2, & \text{si } x \in [8, 12], \\ -21 + 2x, & \text{si } x \in [12, 16]. \end{cases}$$

La región factible es la unión de los siguientes segmentos

$$\mathcal{F} = \{(x, 5 - x/2) : x \in [0, 8]\} \cup \{(x, -3 + x/2) : x \in [8, 12]\} \cup \{(x, -21 + 2x) : x \in [12, 16]\}$$

Pese a la dificultad intrínseca de los problemas binivel se han desarrollado algoritmos exactos para problemas de muy pequeña dimensión. Para resolver el problema lineal binivel se han aplicado algoritmos basados en enumeración implícita, *branch-and-bound*, penalizaciones exactas, o métodos de descomposición. Bi y otros [23], Bialas y Karwn [24, 25], Hansen [118], Júdice y Faustino [134], White y Anandalingan [236].

El caso binivel no lineal ha sido abordado en la monografía de Shimizu y otros [212], en la que desarrollan dos algoritmos para un caso particular de la programación convexa binivel (todas las funciones que definen el problema son convexas). El primer algoritmo está basado en el trabajo previo de Bard [10] y aborda el caso particular de que el nivel inferior es un problema cuadrático convexo, y el nivel superior es un problema estrictamente convexo. Shimizu y otros [212] demostraron que este algoritmo converge a la solución óptima. El segundo algoritmo que presentaron se basa en el trabajo de Jaumard [132] y está desarrollado para el caso anterior y tiene garantizada la convergencia en un número finito de iteraciones.

Luo y otros [154] proponen tres tipo de algoritmos iterativos, el primero (PIPA) está basado en métodos de penalización interior, el segundo en una *programación implícita* y el tercero es una especialización de la *programación secuencial cuadrática* (SQP) empleada para resolver problemas de optimización no diferenciables. Los problemas que debe resolver este algoritmo son problemas diferenciables a trozos.

Los dos primeros algoritmos convergen a puntos estacionarios bajo la cualificación de restricciones de independencia lineal, mientras que el tercero tiene una covergencia local, esto es, depende de la proximidad entre el punto inicial y un punto estacionario. Por contra, tiene una velocidad de convergencia superlineal e incluso cuadrática en algunos casos. Luo y otros [141] realizaron un estudio computacional basado en el paquete MATLAB para el algoritmo de punto interior.

Algoritmos empleados en la resolución de los problemas MPEC aplicados a la planificación del transporte urbano

Son varios los tipos de problemas que se pueden encontrar en la literatura, según la finalidad que persiguen, pero bajo el punto de vista de su estructura matemática existen dos grandes grupos. Por

un lado está el problema de ajuste de matrices origen-destino (tipo I), donde las restricciones del nivel inferior están parametrizadas por las variables del nivel superior y por otro lado modelos de gestión de tráfico (tipo II), donde las restricciones del nivel inferior son independientes de las variables del nivel superior pero la función objetivo, al contrario de lo que ocurría en el caso primero, está parametrizada por ellas. Los algoritmos desarrollados están condicionados fuertemente por el problema a resolver. Algunas de las aplicaciones de los problemas MPEC a la planificación del transporte urbano serán tratados en la sección 2.5, no obstante, en esta sección, anticiparemos algunos algoritmos que han sido aplicados en su resolución.

Una segunda dificultad, además de su dificultad intrínseca, es la gran dimensión de estos problema en las aplicaciones reales. Por ejemplo, si se considera la estimación de matrices O-D en las redes de Madrid o Barcelona el número de variables del nivel inferior serían respectivamente 8659 y 2522, y el número de variables en el nivel superior de 26037 y 7922 respectivamente. En otro tipo de aplicaciones el número de variables en el nivel superior varía entre unas decenas a varios miles. Otra forma de entender el coste computacional es observando que la evaluación de la función objetivo puede llevar algunos minutos, debido a que tiene que resolver un problema de optimización de miles de variables. Esto ha hecho que en la actualidad no se hayan implementado algoritmos exactos para problemas reales de gran escala. Casi la totalidad de los algoritmos desarrollados son heurísticos y son escasas las pruebas computacionales realizadas en problemas de gran escala.

El algoritmo heurístico más extendido es el denominado algoritmo iterativo de *optimización-asignación*. Una iteración de este problema tiene dos fases. En la primera se resuelve el problema del nivel inferior (problema de asignación en equilibrio) y en la segunda se resuelve el problema del nivel superior (problema de optimización) considerando que las variables nivel inferior están fijadas a los valores encontrados en la primera fase. Este algoritmo fue originariamente propuesto por Allsop [6] para un problema de control de tráfico (tipo II) y por Steenbrink [223] para un problema de diseño de redes (tipo II). Tan y otros [225] demostraron computacionalmente que este algoritmo no tiene garantizada su convergencia a un punto estacionario y Marcotte [157] demostró teóricamente este hecho. No obstante, este algoritmo es el más extendido, y ha sido formulado en todos los modelos binivel aplicados al transporte, por ejemplo Hall y otros [114], Yang y otros [245] aplican una modificación de este algoritmo al problema de estimar matrices O-D.

Otra clase de algoritmos heurísticos están basados en el análisis de sensibilidad de los parámetros (ver Tobin y Friesz [226]). Mediante el análisis de sensibilidad se construye una aproximación lineal del problema del nivel inferior en un punto determinado que reemplaza al problema inferior, lo que conduce a un problema de un solo nivel, cuya solución proporciona el próximo punto donde repetir el proceso. Esta clase de algoritmos heurísticos es conocida como *formulación implícita*. Yang [241] utiliza esta metodología para desarrollar dos tipos de aproximaciones lineales (*factores de influencia*) que son aplicados a la estimación de matrices O-D, pero sólo se efectúan pruebas computacionales en redes de pequeñas dimensiones. Friesz y otros [91], Yang y Lam [244] lo aplican a modelos de gestión de tráfico. El algoritmo de Spiess [219] puede ser considerado en esta categoría y es aplicado al problema de estimación de matrices O-D sobre ejemplos de grandes dimensiones.

Se han empleado *métodos de búsqueda probabilística* como el *simulado recocido*. Este método requiere de númerosas evaluaciones del nivel inferior, esta desventaja es aprovechada para hacer una resolución muy eficiente del nivel inferior. Friesz y otros [89] lo aplican a problemas de diseño de redes. Otro tipo de metaheurísticas como los algoritmos genéticos no han sido aplicadas.

Otro tipo de algoritmos, que han sido aplicados, utilizan directamente un método de optimización sobre el problema. Abdulaal y LeBlanc [4] aplican el algoritmo de Hooke-Jeeves al problema de diseño de redes. Codina y Barceló [53] emplean el algoritmo de Wolfe, usado en optimización no diferenciable, para la resolución del problema de estimacón de matrices O-D. Recientemente, Patriksson y Rockafellar [199] adaptan el algoritmo PIPA para un problema binivel aplicado a la gestión del tráfico.

2 Modelos matemáticos aplicados a la planificación del transporte urbano

2.1 Planificación del transporte urbano

El crecimiento económico ha originado un incremento importante de la demanda de transporte tanto en los países desarrollados como en los en vías de desarrollo. Este aumento de la demanda ha conducido a que, en algunas regiones y para ciertos modos de transporte, la demanda supere a la oferta de servicios de transporte, originando que viejos problemas como son la congestión, polución, accidentes, deficit financieros, etc. aparezcan con nuevas apariencias y dificultades.

La demanda de servicios de transporte tiene una naturaleza dinámica, ésta varía durante cada momento del día, de un día a otro de la semana e incluso por meses. La oferta de servicios de transporte es altamente compleja, por un lado una autoridad provee la infraestructura y por otro lado los operadores (pueden existir varios operadores por cada modo de transporte) proveen los servicios.

Empleando el esquema seguido por Ortúzar y Willumsen [189], analizaremos la integración entre la oferta y la demanda de transporte. Considerese la existencia de una demanda de transporte **D**, de bienes o de personas, en un período de tiempo determinado para varios modos de transporte. El sistema se puede caracterizar por:

- ◇ Una infraestructura, por ejemplo, la red de tráfico.
- ◇ Un sistema de gestión, por ejemplo, el control semafórico y viario del tráfico.
- ◇ Un conjunto de modos de transporte y sus operadores.

Considérese unos volúmenes **V** (de pasajeros, de tráfico, etc.) en la red, a los que le corresponde cierto conjunto de velocidades **S**, y una capacidad de operación **Q**, bajo un sistema de gestión **M**. En términos generales la velocidad en la red de transporte puede ser representada por

$$\mathbf{S} = f(\mathbf{Q}, \mathbf{V}, \mathbf{M}). \tag{6}$$

La velocidad puede ser considerada como una aproximación inicial al concepto de *nivel de servicio* del sistema de transporte. En términos generales el nivel de servicio sería una combinación de velocidades (o tiempos de viajes, de espera, caminando, etc.) y costes (precios de los billetes, del combustible, etc).

Los volúmenes en la red dependerán de la demanda **D** y de la capacidad del sistema de transporte **Q**, esto es

$$\mathbf{V} = g(\mathbf{D}, \mathbf{Q}).$$

El sistema de gestión puede incluir esquemas de control de tráfico, de regulación aplicado a cada modo de transporte, etc. La capacidad **Q** depende del sistema de gestión y de los niveles de inversión **I**, entonces

$$\mathbf{Q} = h(\mathbf{I}, \mathbf{M}).$$

El sistema de gestión puede ser empleado para redistribuir la capacidad entre la infraestructura, produciendo **Q'** y dando prioridad a ciertos tipos de usuarios sobre otros (usuarios de transporte público, ciclistas, vehículos eléctricos, peatones, etc).

Como en el caso de muchos bienes de consumo y servicios, uno podría esperar que el nivel de demanda **D** dependiese del nivel de servicio prestado por el sistema de transporte y la localización de las actividades **A** en el espacio

$$\mathbf{D} = r(\mathbf{S}, \mathbf{A})$$

Combinando las relaciones anteriores y para una actividad fija, se encontraría el punto de equilibrio entre la oferta y la demanda mediante una ecuación del tipo

$$\mathbf{D} = q(\mathbf{D}, \mathbf{I}, \mathbf{M})$$

Realmente el nivel de actividad cambia tanto en el tiempo como en el espacio, y esto probablemente haría variar los niveles de servicio. Se podrían considerar dos situaciones de equilibrio, una a corto plazo y otra a largo plazo. La tarea de la planificación del transporte urbano es predecir y controlar la evolución de estos puntos de equilibrio en el tiempo, de modo que el bienestar de la sociedad sea maximizado. La modelización de estos puntos de equilibrio pemite entender mejor la evolución de éstos y ayuda al desarrollo e implementación del conjunto de estrategias de gestión **M** y a los programas de inversión **I**.

La modelización de esta situación de equilibrio ha motivado multitud de modelos matemáticos. La práctica desde la década de los años sesenta ha consolidado el esquema de aplicación llamado *modelo de cuatro etapas*

1. Fase de generación de viajes. El esquema empieza considerando una zonificación del área de estudio, una codificación de la red de transporte y la obtención de una base de datos para el estudio. Estos datos están referidos al nivel de actividad económica y demográfica de cada zona que incluye el nivel de empleo, localización de centros comerciales, zonas recreativas, centros escolares, etc. y son empleados para estimar el número de viajes generados y atraídos por cada zona considerada en el estudio.

   Tras esta fase se obtiene una modelización de la red de transporte mediante un grafo $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ donde $\mathcal{A}$ y $\mathcal{N}$ son el conjunto de arcos (dirigidos) y nodos respectivamente. El significado de los arcos depende de si la red es de tráfico o de transporte público. En el primer caso los arcos están asociados a las calles y los nodos a las intersecciones. En el segundo caso cada nodo está asociado a una parada y cada arco representa los posibles desplazamientos entre paradas que un usuario puede efectuar. Hay arcos asociados a movimientos en el vehículo, andando o esperando.

2. Fase de distribución. En esta fase se obtiene la distribución de los viajes sobre el espacio, esto es, se obtiene el número de viajes que se efectúan de una zona a otra, obteniéndose la denominada matriz de viajes origen-destino (O-D). En esta fase se obtiene un conjunto de pares ordenados de $\mathcal{N} \times \mathcal{N}$ y la demanda de viajes (que inicialmente consideraremos fija). Denotamos este conjunto de pares de demandas por W y cada par O-D por $\omega = (i, j)$, donde i es el origen y j es el destino. Denotamos por $\bar{g}_\omega$ la demanda total de viajes para el par ω.

3. Fase de partición modal. En esta fase se obtiene una matriz O-D para cada modo de transporte presente en el estudio.

4. Fase de asignación. Finalmente, cada matriz de demanda O-D es asignada a un conjunto de rutas en la red de transporte. Usualmente se efectúa una asignación de tráfico por un lado (vehículos privados) y por otro lado una asignación a la red de transporte público.

En la actualidad este esquema secuencial ha sido superado por métodos que integran dos o varias de estas etapas simultáneamente. No obstante, este esquema sigue siendo de utilidad a la hora de describir modelos o de comparar modelos alternativos.

2.2 Modelos de asignación de tráfico en equilibrio

En esta subsección estudiaremos algunos modelos matemáticos que se han sido planteados para resolver el problema de la asignación de tráfico a las rutas de la red de transporte y se clasifican en *dinámicos* o *estáticos* en función de como es considerado el aspecto dinámico de la demanda (esta tesis doctoral aborda exclusivamente modelos estáticos). Estos modelos se centran en unas pocas horas del día, como las horas puntas, y trabajan con valores medios (demandas, tiempos, flujos, etc.) durante el período de estudio.

Además deben de asumir un principio para la modelización de la elección que hacen los usuarios de la ruta en la red de transporte. Un marco para la elaboración de estos modelos, llamados modelos de *asignación en equilibrio*, lo constituye el *primer principio de Wardrop* [233] que se enuncia del siguiente modo:

> "En el equilibrio ningún usuario puede reducir el coste de su viaje mediante cambio de ruta."

Este prinicipo implica que todos los tiempos de viaje empleados en todas las rutas usadas para satisfacer el mismo par O-D deben ser iguales y menor o igual al tiempo de viaje en cualquier otra ruta no empleada para satisfacer dicho par de demanda. Este principio ha sido empleado para construir modelos de equilibrio, tanto en redes de tráfico como en redes de transporte público.

El primer principio de Wardrop, también denominado DUE (*deterministic user equilibirum*), asume que todos los usuarios perciben el coste de la misma manera y además, conocen los costes de todas las rutas (tienen información perfecta). En la realidad las percepciones de los costes están sujetas a variaciones y los usuarios eligen la ruta de acuerdo con su percepción. Se han elaborado modelos de equilibrio en los que los costes de viajes son la suma de una parte fija más una componente aleatoria, en este caso los usuarios eligen una u otra ruta dependiendo de la distribución de probabilidad de los costes aleatorios. Esta forma de asignación recibe el nombre *asignación estocástica* (SUE).

Existe otro marco en la elaboración de modelos de equilibrio, el llamado *segundo principio de Wardrop* que asume que los usuarios puden ser persuadidos a emplear cualquier ruta y por tanto, los usuarios serán asignados a las rutas que minimicen el tiempo total empleado por el sistema de transporte. Este principio se enuncia

> "Los usuarios eligen la ruta de modo que se minimice el tiempo total de transporte en la red."

El primer principio de Wardrop es utilizado para modelizar el comportamiento de los usuarios, mientras que el segundo principio es usado como un criterio para diseñar la red de transporte. El primer principio asume que los usuarios actúan individualmente mientras que el segundo asume que los usuarios buscan el óptimo del sistema (de todos los usuarios)

Ahora vamos a abordar las formulaciones matemáticas del problema de asignación de tráfico. Denotamos por P_ω el conjunto de rutas para el par O-D ω, con h_p el flujo en la ruta p y por $C_p := C_p(\mathbf{h})$ el coste de viaje en la ruta p experimentado por un usuario para un vector de flujo $\mathbf{h} \in \Re^{|P|}$, siendo $|P|$ el número total de rutas en la red. Empleando esta notación un vector de flujo $\mathbf{h}^*$ está en equilibrio si y sólo si

$$\begin{array}{l} \text{Si } h_p^* > 0 \Rightarrow C_p = U_\omega,\ p \in P_\omega,\ \omega \in W. \\ \text{Si } h_p^* = 0 \Rightarrow C_p \geq U_\omega,\ p \in P_\omega,\ \omega \in W. \end{array} \tag{7}$$

donde el valor de $U_\omega := U_\omega(\mathbf{h}^*)$ es el mínimo coste de transporte en las rutas del par O-D $\omega \in W$

Para formular matemáticamente las condiciones de equilibrio dadas en (6.15) se necesita describir los requerimientos de factibilidad de los flujos. Denotamos los flujos en los arcos como $\mathbf{f} \in \Re^{|\mathcal{A}|}$ y por P_ω el conjunto de rutas para el par O-D ω. El primer requerimiento es que la demanda para cada par O-D ω debe ser satisfecha, esto es

$$\sum_{p \in P_\omega} h_p = \bar{g}_\omega,\ \forall \omega \in W. \tag{8}$$

Además los flujos deben ser no negativos

$$h_p \geq 0,\ \forall p \in P. \tag{9}$$

La relación entre los flujos en los arcos y los flujos en las rutas viene definida por

$$\sum_{w \in W} \sum_{p \in P_\omega} \delta_{lp} h_p = f_l,\ \forall l \in \mathcal{A},$$

donde $\delta_{lp} = 1$ si la ruta p utiliza el arco l y cero en caso contrario. Esta restricción indica que el flujo en un arco l es la suma del flujo de todos los caminos que emplean dicho arco.

La primera formulación de las condiciones de equilibrio (7) mediante un modelo de optimización fue realizada por Beckman y otros [13]. Estos autores asumieron que el coste en cada arco depende exclusivamente de su flujo (*costes separables*). La solución del siguiente problema de optimización caracteriza la situación de equilibrio derivada del primer principio de Wardrop

$$\begin{array}{ll} \text{minimizar} & Z = \sum_{l\in\mathcal{A}} \int_0^{f_l} c_l(x)dx \\ \text{sujeto a} & \sum_{p\in P_\omega} h_p = \bar{g}_\omega,\ \forall\omega \in W, \\ & \sum_{\omega\in W}\sum_{p\in P_\omega} \delta_{lp}h_p = f_l,\ \ \forall l \in \mathcal{A}, \\ & h_p \geq 0,\ \forall p \in P_\omega,\ \forall\omega \in W. \end{array} \qquad \text{[TAP]}$$

La situación de equilibrio derivada del segundo principio de Wardrop (para costes separables y crecientes) se puede caracterizar como solución del siguiente modelo de optimización.

$$\begin{array}{ll} \text{minimizar} & Z = \sum_{l\in\mathcal{A}} c_l(f_l)f_l \\ \text{sujeto a} & \sum_{p\in P_\omega} h_p = \bar{g}_\omega,\ \forall\omega \in W, \\ & \sum_{\omega\in W}\sum_{p\in P_\omega} \delta_{ap}h_p = f_l,\ \ \forall l \in \mathcal{A}, \\ & h_p \geq 0,\ \forall p \in P_\omega,\ \forall\omega \in W. \end{array} \qquad \text{[TAP-SE]}$$

Se ha recurrido a modelos matemáticos más generales que los modelos de optimización (como son las desigualdades variacionales, problemas de complementariedad o formulaciones de punto fijo) para modelizar situaciones más realistas del problema de asignación, tales como que el coste de un arco no depende exclusivamente de su flujo sino que puede depender del flujo en otros arcos de la red. Esta situación es la común en las intersecciones de calles. Otros nuevos aspectos del problema que pueden ser tenidos en cuenta es la existencia de varios tipos de usuarios (*modelos multiusuarios*) o la existencia de varios modos de tranporte (*modelos multimodales*) como por ejemplo el transporte público y privado simultáneamente. Los costes derivados de este tipo de problemas tienen un Jacobiano típicamente asimétrico y por tanto no son formulables mediante un modelo de optimización. (Ver el teorema 1.2)

La formulación matemática más usual de las condiciones de equilibrio Wardropianas, se basa en los problemas de desigualdades variacionales en el espacio de flujo en las rutas $h_p,\ p \in P$, en este caso el espacio de factibilidad de los flujos está definido por las ecuaciones (8) y (9), y puede ser expresado matricialmente por

$$\bar{\mathbf{g}} = \delta^{\bar{\mathbf{g}}}\mathbf{h}, \qquad (10.a)$$

$$\mathbf{h} \geq \mathbf{0}, \qquad (10.b)$$

donde $\bar{\mathbf{g}} \in \Re_+^{|W|}$ es el vector de demandas, $\delta^{\bar{\mathbf{g}}}$ es la matriz de incidencia par origen/destino-ruta. El elemento $\delta_{\omega p}$ de esta matriz vale 1 si la ruta p une el par ω y 0 en caso contrario. El espacio de flujo en las rutas es

$$\Omega_{\mathbf{h}} = \left\{ \mathbf{h} \in \Re_+^{|P|} \,\middle|\, \delta^{\bar{\mathbf{g}}}\mathbf{h} = \bar{\mathbf{g}} \right\}.$$

Las condiciones de Wardrop (7) se pueden formular mediante el siguiente problema en desigualdades variacionales. Encontrar un $\mathbf{h}^* \in \Omega_{\mathbf{h}}$ cumpliendo la desigualdad

$$\mathbf{C}(\mathbf{h}^*)^T(\mathbf{h} - \mathbf{h}^*) \geq 0, \quad \forall \mathbf{h} \in \Omega_{\mathbf{h}} \qquad \text{[TAP-VIP}(\mathbf{C}(\cdot), \Omega_{\mathbf{h}})\text{]}$$

donde $\mathbf{C}(\mathbf{h}^*)$ es el coste en los caminos para el flujo $\mathbf{h}^*$. Notar que si existiesen varios grupos de usuarios o modos de transporte estas situaciones son fácilmente modelizables. Es suficiente crear tantas copias de la red de transporte como grupos de usuarios o modos de transporte, relacionando sus costes de viaje, en caso necesario, mediante el vector de costes $\mathbf{C}$.

En el caso de que el coste en una ruta fuese la suma de los costes en cada uno de sus arcos (*costes aditivos*), las condiciones de Wardrop pueden ser descritas en términos de los flujos en los arcos. El conjunto de factibilidad para los flujos en los arcos viene descrito por el conjunto (poliedro acotado)

$$\mathbf{f} = \delta^{\mathbf{f}}\mathbf{h}, \tag{11.a}$$
$$\bar{\mathbf{g}} = \delta^{\bar{\mathbf{g}}}\mathbf{h}, \tag{11.b}$$
$$\mathbf{h} \geq \mathbf{0}, \tag{11.c}$$

donde $\delta^{\mathbf{f}}$ es la matriz de incidencia arco-ruta, el elemento $\delta^{\mathbf{f}}_{lp}$ toma el valor 1 si la ruta p emplea el arco l y vale 0 en caso contrario. En este caso el conjunto de factibilidad viene definido por

$$\Omega_{\mathbf{f}} = \left\{ \mathbf{f} \in \Re^{|\mathcal{A}|} \,\middle|\, \exists \mathbf{h} \in \Omega_{\mathbf{h}} \text{ con } \mathbf{f} = \delta^{\mathbf{f}}\mathbf{h} \right\},$$

y el problema VIP$(\mathbf{C}(\cdot), \Omega_{\mathbf{f}})$ es equivalente a encontrar un $\mathbf{f}^* \in \Omega_{\mathbf{f}}$ cumpliendo la desigualdad

$$\mathbf{c}(\mathbf{f}^*)^T(\mathbf{f} - \mathbf{f}^*) \geq 0, \quad \forall \mathbf{f} \in \Omega_{\mathbf{f}} \qquad \text{[TAP-VIP}(\mathbf{c}(\cdot), \Omega_{\mathbf{f}})\text{]}$$

donde $\mathbf{c} : \Re^{|\mathcal{A}|} \mapsto \Re^{|\mathcal{A}|}$es la función de costes en los arcos. La relación entre los costes en los arcos y los costes en los caminos viene dado por $\mathbf{C}(\mathbf{h}) = {\delta^{\mathbf{f}}}^T\mathbf{c}(\mathbf{f})$.

El conjunto $\Omega_{\mathbf{f}}$ puede ser formulado sin necesidad de recurrir a las variables de flujos en los caminos. Esta representación es conocida con el nombre de formulación *nodo-arco*, e impone para cada par O-D las condiciones de consevación de flujo en cada nodo. Esto es

$$\mathbf{E}\mathbf{f}^{\omega} = \bar{\mathbf{g}}^{\omega}, \quad \forall \omega \in W,$$

donde $\mathbf{E} \in \{-1, 0, 1\}^{|\mathcal{N}| \times |\mathcal{A}|}$ es la matriz de incidencia nodo arco de la red, las componentes del vector $\bar{\mathbf{g}}^{\omega}$ están definidas por

$$\bar{g}^{\omega}_k = \begin{cases} 1, & \text{si } \omega = (k, j), \\ -1, & \text{si } \omega = (i, k), \\ 0, & \text{si } k \text{ no es el nodo origen o destino del par } \omega, \end{cases}$$

y el vector $\mathbf{f}^{\omega}$ es el vector de flujos en los arcos producidos por el par de demanda O-D ω. El flujo agregado en los arcos será la suma de todos los flujos producidos por todos los pares O-D, esto es $\mathbf{f} = \sum_{\omega \in W} \mathbf{f}^{\omega}$. Resumiendo, la región factible en los arcos puede venir definida directamente por

$$\Omega_{\hat{\mathbf{f}}} = \left\{ \mathbf{f} \in \Re^{|\mathcal{A}|} \,\middle|\, \exists \mathbf{f}^{\omega} \in \Re^{|\mathcal{A}|}_{+} \forall \omega \in W \text{ con } \mathbf{f} = \sum_{\omega \in W} \mathbf{f}^{\omega} \text{ y } \mathbf{E}\mathbf{f}^{\omega} = \bar{\mathbf{g}}^{\omega} \right\},$$

dando lugar al problema equivalente de encontrar un $\mathbf{f}^* \in \Omega_{\hat{\mathbf{f}}}$ que cumple la desigualdad

$$\mathbf{c}(\mathbf{f}^*)^T(\mathbf{f} - \mathbf{f}^*) \geq 0, \quad \forall \mathbf{f} \in \Omega_{\hat{\mathbf{f}}} \qquad \text{[TAP-VIP}(\mathbf{c}(\cdot), \Omega_{\hat{\mathbf{f}}})\text{]}$$

Charnes y Cooper [45] describen la situación de equilibrio en el sistema de transporte como un juego de equilibrio *no-cooperativo de Nash* [177]. Los jugadores de este juego están definidos para cada par O-D, y éstos compiten para minimizar sus costes de transporte mediante una adecuada asignación del flujo a las rutas. Estos flujos constituyen el conjunto de estrategias para cada jugador. Formalmente,

en un juego no cooperativo de M jugadores se asume la existencia de $\theta_i : \prod_{j=1}^{M} Y_j \mapsto \Re$ funciones de pago para cada jugador, definidas sobre el espacio de estrategias $Y = \prod_{j=1}^{M} Y_j$ y se supone que cada espacio individual Y_i es un conjunto convexo. Un punto $\mathbf{y}^*$ es una situación de equilibrio en el juego no cooperativo de Nash si y sólo si para cada $i \in \{1, \cdots, M\}$, se cumple

$$\mathbf{y}_i^* \in \arg \text{ minimizar}\{\theta_i(\mathbf{y}_i, \mathbf{y}_{\neq i}^*) : \mathbf{y}_i \in Y_i\}$$

donde $\mathbf{y}_{\neq i}^*$ es el conjunto de componentes de $\mathbf{y}^*$ distintas a $\mathbf{y}_i$. Es decir, la estrategia de un jugador es óptima respecto a su propia función de pago y teniendo en cuenta el resto de estrategias de los otros jugadores. Para el caso de funciones de coste de viaje separable las funciones de pago del juego vienen definidas por la expresión

$$\theta_\omega(\mathbf{f}) = \sum_{l \in \mathcal{A}^\omega} \int_0^{f_l} c_l(x)dx,$$

donde $\mathcal{A}^\omega$ es el conjunto de arcos empleados por el par $\omega \in W$.

En este juego, a diferencia del de Stackelberg, no existe un jugador especial (el líder) que pueda modificar el conjunto de estrategia del resto de jugadores.

2.3 Modelos de asignación en transporte público

El problema de asignación de tráfico aborda la modelización de cómo los usuarios de vehículos privados eligen su ruta. Esta modelización se complica para los usuarios de transporte público (autobús urbano, metro, tranvía, etc.). Antes de entrar en esta discusión conviene introducir las siguientes definiciones.

- ◇ Denominaremos línea de transporte público o simplemente *línea* a una flota de vehículos que opera entre dos puntos (*terminales*) sobre una red y que poseen generalmente las mismas características de tamaño, capacidad y velocidad. Estos vehículos paran en un conjunto de nodos dispuestos en su trayecto, en los que se está permitido subir y bajar del vehículo. En definitiva, cada línea de transporte público está definida por las características de los vehículos, su trayecto y su frecuencia.
- ◇ Una *sección de línea* es cualquier parte del trayecto entre dos nodos de una línea (no necesariamente consecutivos).
- ◇ Una *sección de ruta* es cualquier camino entre dos nodos de transferencia. Cada sección de ruta tiene asociada un conjunto de secciones de líneas que se denominan *líneas atractivas* o *líneas comunes*.
- ◇ Una *estrategia* es un conjunto de reglas que a un usuario le permite llegar a su destino.

Comenzaremos con el llamado *problema de líneas comunes*. Un usuario puede estar esperando en una parada por la que pasan varias líneas de transporte. Algunas de ellas pueden ser *atractivas* para realizar su viaje, entonces el usuario elegirá el primer vehículo de la primera línea atractiva que llegue a la parada. En este problema también los usuarios intentan minimizar su tiempo de viaje, pero este mínimo ya no se consigue mediante la elección de un único camino, sino mediante la elección de un conjunto de caminos, lo que se denomina *hipercamino*.

El primer modelo para el problema de líneas comunes en redes no congestionadas fue dado por Chiriqui [49] y por Chiriqui y Robillard [50]. Consideremos una sección de ruta definida por dos nodos (paradas) en la red de transpote público $s = (i, j)$ y sea A_s el conjunto de todas las líneas de autobús que un usuario dispone para ir del nodo i al nodo j. El usuario elegirá un subconjunto $\bar{A}_s \subset A_s$ de líneas que minimice el tiempo esperado de viaje. Este conjunto de líneas se denomina conjunto de líneas comunes o atractivas. Dada esta elección el usuario subirá en el primer vehículo que llegue al nodo i y perteneza a la línea $l \in \bar{A}_s$. El tiempo esperado de viaje entre el nodo i y j es

$$C_s = W_{\bar{A}_s} + T_{\bar{A}_s}$$

donde $W_{\bar{A}_s}$ y $T_{\bar{A}_s}$ representan los tiempos medios de espera y de viaje en el vehículo.

Si ϕ_l y t_l denotan respectivamente la frecuencia y tiempo de viaje en de la línea de transporte $l \in A_s$ (que consideraremos constante). El tiempo medio de espera será

$$W_{\bar{A}_s} = \frac{1}{\sum_{l \in \bar{A}_s} \phi_l} \tag{12}$$

y el tiempo medio de viaje es

$$T_{\bar{A}_s} = \sum_{l \in \bar{A}_s} t_l \pi_l^s, \tag{13}$$

donde π_l^s es la probabilidad de que el usuario tome el vehículo de la línea l. Esta probabilidad viene dada por las frecuencias de las líneas y es

$$\pi_l^s = \frac{\phi_l}{\sum_{l' \in \bar{A}_s} \phi_{l'}}.$$

El problema de encontrar el conjunto de líneas atractivas se puede formular mediante un problema de optimización. Consideremos las variables de decisión dicotómicas $x_l \in \{0,1\}$ con $l \in A_s$. Esta variable toma el valor 1 si la línea l es considerada una línea atractiva para viajar de i a j y 0 en caso contrario. La solución del siguiente problema definirá el conjunto de líneas comunes.

$$\begin{aligned} \text{minimizar} \quad & C_s = \frac{1+\sum_{l \in A_s} t_l \phi_l x_l}{\sum_{l \in A_s} \phi_l x_l} \\ \text{sujeto a} \quad & x_l \in \{0,1\}, \quad \forall l \in A_s. \end{aligned}$$

Este problema es un problema de programación lineal entera fraccional (ver García y otros [93]) y ha sido resuelto por Chiriqui y Robillard [50].

Para plantear el problema de asignación se construye una red de transporte $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ donde el conjunto de arcos $\mathcal{A}$ está asociado al conjunto de secciones de ruta y el conjunto de nodos $\mathcal{N}$ al conjunto de paradas. Cada arco $s \in \mathcal{A}$ tiene asociado un tiempo esperado de viaje C_s obtenido al resolver su problema de líneas comúnes asociado. Se asume que la red tiene un conjunto de demandas W, y que $\bar{g}_\omega$ es la demanda para cada par O-D $\omega \in W$. El problema de asignación es fomulado mediante el siguiente problema de caminos mínimos

$$\begin{aligned} \text{minimizar} \quad & Z = \textstyle\sum_{s \in \mathcal{A}} C_s V_s \\ \text{sujeto a} \quad & \textstyle\sum_{p \in P_\omega} h_p = \bar{g}_\omega, \ \forall \omega \in W, \\ & \textstyle\sum_{\omega \in W} \sum_{p \in P_\omega} \delta_{sp} h_p = V_s, \quad \forall s \in \mathcal{A}, \\ & h_p \geq 0, \ \forall p \in P_\omega, \ \forall \omega \in W \end{aligned}$$

donde V_s es el volumen de usuarios (o simplemente flujo) en la sección de ruta s, δ_{sp} vale uno si el camino p utiliza la sección de ruta s y 0 en caso contrario.

Notar que la resolución de este problema proporiciona los volúmenes en las secciones de ruta. Haría falta efectuar su descomposición en volúmenes de secciones de línea, y esto se obtiene mediante la relación

$$v_l^s = \frac{\phi_l}{\sum_{l \in \bar{A}_s} \phi_l} V_s$$

donde v_l^s es el volumen de pasajeros de la sección de ruta s asignados a la sección de línea l. El método aquí expuesto para calcular la asignación en transporte público no congestionado es el seguido por De Cea y Fernández [43]. Alternativamente Spiess y Florian [220] plantean un único problema para resolver ambos problemas (el de asignación y el de líneas comunes). El modelo que ellos plantearon es equivalente a

$$
\begin{aligned}
\text{minimizar} \quad & Z = \textstyle\sum_{s\in\mathcal{A}} C_s V_s \\
\text{sujeto a} \quad & C_s = \frac{1+\sum_{l\in A_s} t_l \phi_l x_l^s}{\sum_{l\in A_s} \phi_l x_l^s} \\
& \textstyle\sum_{p\in P_\omega} h_p = \bar{g}_\omega, \ \forall \omega \in W, \\
& \textstyle\sum_{\omega\in W} \sum_{p\in P_\omega} \delta_{sp} h_p = V_s, \quad \forall s \in \mathcal{A}, \\
& v_l^s = \frac{x_l^s \phi_l}{\sum_{l'\in A_s} x_{l'}^s \phi_{l'}} V_s, \quad \forall l \in \mathcal{A}_s, \ \forall s \in \mathcal{A} \\
& h_p \geq 0, \ \forall p \in P_\omega, \ \forall \omega \in W \\
& x_l^s \in \{0,1\}, \quad \forall l \in A_s, \ \forall s \in \mathcal{A}.
\end{aligned}
\qquad \text{[TEAP]}
$$

donde x_l^s vale 1 si la línea l es atractiva para los usuarios de la sección s y 0 en caso contrario.

El modelo descrito no tiene en cuenta el efecto de la congestión. Spiess [218] elabora una versión del modelo aquí descrito donde el tiempo en los vehículos es una función de los volúmenes en los arcos. Estas funciones miden lo que los autores denominan *discomfort*. Esta es una medida de las sensaciones del usuario en función del número de usuarios en un vehículo en relación a su capacidad. La principal desventaja es que el tiempo de espera no se ve afectado por la congestión. De Cea y Fernández [44] consideran un modelo de asignación de tranporte público con restricciones de capacidad donde el tiempo en las paradas depende de la demanda y de la capacidad de los vehículos. Consideran las denominadas *frecuencias efectivas* que son el número de vehículos donde existe capacidad para embarcar frente a las llamadas *frecuencias nominales* que son el número de vehículos (con y sin capacidad para embarcarse) por unidad de tiempo.

Nguyen y Pallotino [181] introducen el concepto de hipercamino para formular y describir las estrategias entre los pares O-D. Este modelo es equivalente al de Spiess [218] y este marco ha sido empleado en otros trabajos como los modelos de Wu and Florian [239], Wu y otros [240]. Estos modelos consideran el tiempo de espera en las paradas independiente del flujo. Recientemente han aparecido modelos que emplean la formulación en hipercaminos y consideran el tiempo de espera como una función del flujo. Una revisión de estos modelos la realiza Bouzaïene-Ayari y otros [28].

2.4 Modelos combinados

Se han desarrollado modelos que colapsan varias de las fases en un único modelo. Estos modelos reciben el nombre de *modelos combinados*.

La primera ventaja de los modelos combinados es que hacen consistentes las etapas. Por ejemplo, tras la fase de asignación se obtienen nuevos tiempos de viajes que serán (probablemente) diferentes de los empleados en la fase de distribución de viajes.

La segunda ventaja es que consideran simultáneamente todas las posibles respuestas que un viajero puede realizar cuando se incrementan significativamente los niveles de congestión como, por ejemplo, cambio de ruta, elección de aparcamientos, cambio de modo de transporte, cambio de destino, variación de la frecuencia de viajes y/o la hora de su realización, etc. Algunos de estos aspectos han sido ya recogidos en los modelos actuales, pero sobre todo la modelización de los aspectos dinámicos queda todavía por desarrollar.

Modelo combinado de asignación y elección de modo

El TAP ha sido formulado asumiendo que la demanda es fija (indepediente de los costes de transporte) pero es más realista considerar la naturaleza *elástica* de ésta. Si los costes de transporte crecen, un usuario podría decidir no realizar el viaje o realizarlo en un modo de transporte alternativo. Supondremos que el número de viajes para un par de demanda ω es una función del coste de transporte U_ω para dicho par, esto es

$$g_\omega = G_\omega(U_\omega) \tag{14}$$

Supondremos que la función G_ω es no negativa, continua y estríctamente decreciente. Su función inversa calcula el coste de viaje en función del número de usarios, es decir $U_\omega = G_\omega^{-1}(g_\omega)$. Notar que ahora g_ω es una variable y ésta tiene el mismo papel que el parámetro $\bar{g}_\omega$ en la formulación inelástica del TAP.

La solución del siguiente modelo, que combina asignación y demanda, satisface el primer principio de Wardrop y la ecuación (14).

$$\begin{array}{ll} \text{minimizar} & Z = \sum_{l\in\mathcal{A}} \int_0^{f_l} c_l(x)dx - \sum_{\omega\in W} \int_0^{g_\omega} G_\omega^{-1}(x)dx \\ \text{sujeto a} & \sum_{p\in P_\omega} h_p = g_\omega,\ \forall\omega \in W, \\ & \sum_{\omega\in W}\sum_{p\in P_\omega} \delta_{ap} h_p = f_l,\ \ \forall l \in \mathcal{A}, \\ & h_p \geq 0,\ \forall p \in P_\omega,\ \forall\omega \in W. \end{array} \qquad \text{[TAP-E]}$$

Beckman [13] fue el primero en formular el TAP-E empleando una formulación nodo-arco.

Para ilustrar la manera en que este modelo puede recoger la situación de partición modal, se puede suponer que los usuarios eligen entre dos modos de transporte *privado* y *público* y la distribución modal viene dada por una función logit (modelo de demanda). Esta función determina el número de usuarios en cada modo de transporte en función de sus costes de viaje mediante la expresión

$$g_\omega^k = G_\omega^k(\mathbf{U}_\omega) = \frac{\exp -\left(\alpha^k + \beta_1 U_\omega^k\right)}{\sum_{k'\in\{a,b\}} \exp -\left(\alpha^{k'} + \beta_1 U_\omega^{k'}\right)} \bar{g}_\omega \tag{15}$$

donde U_ω^k es el coste de viajar en modo $k \in \{a, b\}$ para el par O-D ω, $\mathbf{U}_\omega$ es el vector de costes de transporte en cada alternativa, $\bar{g}_\omega$ es la demanda total de viajes para el par O-D ω. Los coeficientes α^k, β_1 son los parámetros logit del modelo. Denotamos la alternativa en vehículo privado mediante (a) y en transporte público mediante (b). La expresión (15) se puede simplificar a

$$g_\omega^a = G_\omega^a(\mathbf{U}_\omega) = \frac{1}{1 + \exp -\left(\alpha^{ab} + \beta_1 (U_\omega^b - U_\omega^a)\right)} \bar{g}_\omega$$

donde $\alpha^{ab} = \alpha^b - \alpha^a$. El modelo asume que el coste de viaje U_ω^b mediante transporte público es independiente de los volúmenes de tráfico y por esa razón es considerado constante. La inversa de la función demanda es

$$U_\omega^a = G^{-1}(g_\omega^a) = U_\omega^b + \frac{1}{\beta_1}\left[\alpha^{ab} + \log(\bar{g}_\omega - g_\omega^a) - \log(g_\omega^a)\right]$$

Empleando la relación $g_\omega^a + g_\omega^b = \bar{g}_\omega$, podremos reescribir la expresión

$$-\int_0^{g_\omega^a} G^{-1}(x)dx = U_\omega^b g_\omega^b + (1/\beta_1) \sum_{k\in\{a,b\}} g_\omega^k(\log g_\omega^k - 1 + \alpha^k) + C$$

donde C es una constante.

La función objetivo del TAP-E para este caso es

$$\sum_{l\in\mathcal{A}} \int_0^{f_l} c_l(x)dx + \sum_{\omega\in W} U_\omega^b g_\omega^b + (1/\beta_1) \sum_{\omega\in W} \sum_{k\in\{a,b\}} g_\omega^k(\log g_\omega^k - 1 + \alpha^k)$$

Marín [165] resuelve este modelo empleando programación geométrica generalizada, obteniendo ventajas computacionales sobre el método de Frank-Wolfe .

Las condiciones de Wardrop se formulan en modo muy general y no asumen ninguna particular propiedad de las funciones de coste de viaje, ni de la función de demanda, solamente la no negatividad

de las mismas. La formulación de estas condiciones mediante el anterior modelo de optimización requiere que los costes de viaje sean aditivos y separables, y que la función de demanda sea separable. En caso de que alguna de las dos funciones no satisfaciesen la propiedad de separabilidad habría que recurrir a una formulación variacional del problema. El caso elástico considera el vector de demanda como nueva variable en la región de factiblidad, obteniendo

$$\Omega_{\mathbf{h}}^{\mathbf{g}} = \left\{ (\mathbf{h}, \mathbf{g}) \in \Re_+^{|P|} \times \Re_+^{|W|} \,\middle|\, \delta^{\bar{\mathbf{g}}}\mathbf{h} = \mathbf{g} \right\}$$

La formulación variacional del TAP-E consiste en encontrar un $(\mathbf{h}^*, \mathbf{g}^*) \in \Omega_{\mathbf{h}}^{\mathbf{g}}$ que satisfaga la siguiente desigualdad

$$\mathbf{C}(\mathbf{h}^*)^T(\mathbf{h} - \mathbf{h}^*) - \mathbf{G}^{-1}(\mathbf{g}^*)^T(\mathbf{g} - \mathbf{g}^*) \geq 0, \quad \forall (\mathbf{h}, \mathbf{g}) \in \Omega_{\mathbf{h}}^{\mathbf{g}} \qquad [\text{TAP-E-VIP}(\mathbf{C}, \Omega_{\mathbf{h}}^{\mathbf{g}})]$$

Fisk y Boyce [77] fueron los primeros que plantearon la formulación variacional del problema de asignación con demanda elástica en el espacio de flujo en las rutas. La correspondiente formulación mediante desigualdades variacionales en el espacio de flujo en los arcos fue dada por Florian [79] y Dafermos [60], ésta consiste en encontar un $(\mathbf{f}^*, \mathbf{g}^*) \in \Omega_{\mathbf{f}}^{\mathbf{g}}$ cumpliendo

$$\mathbf{c}(\mathbf{f}^*)^T(\mathbf{f} - \mathbf{f}^*) - \mathbf{G}^{-1}(\mathbf{g}^*)^T(\mathbf{g} - \mathbf{g}^*) \geq 0, \quad \forall (\mathbf{f}, \mathbf{g}) \in \Omega_{\mathbf{f}}^{\mathbf{g}} \qquad [\text{TAP-E-VIP}(\mathbf{c}, \Omega_{\mathbf{f}})]$$

donde

$$\Omega_{\mathbf{f}}^{\mathbf{g}} = \left\{ (\mathbf{f}, \mathbf{g}) \in \Re^{|\mathcal{A}|} \times \Re^{|\mathcal{W}|} \,\middle|\, \exists (\mathbf{h}, \mathbf{g}) \in \Omega_{\mathbf{h}}^{\mathbf{g}} \text{ con } \mathbf{f} = \delta^{\mathbf{f}}\mathbf{h} \right\}$$

Modelo combinado de asignación y distribución

En la fase de distribución de la demanda se asume que el número de viajes generados en cada origen es conocido, así como el número de viajes atraídos por cada destino. Denotaremos por O_i el número de usuarios que salen del orígen i, y por D_j el número de usuarios que llegan al destino j. El objetivo es determinar la matriz de viajes origen destino $\{g_\omega\}$ con $\omega = (i, j) \in W$. Ésta debe satisfacer

$$\sum_j g_{ij} = O_i, \quad \forall i, \tag{16.a}$$

$$\sum_i g_{ij} = D_j, \quad \forall j. \tag{16.b}$$

Los modelos de distribución asumen que el número de viajeros entre zonas dependen del potencial de cada una de estas zonas para atraer o generar viajes y de la distancia entre ambas zonas. Muchos de estos modelos tiene la expresión

$$g_{ij} = G_{ij}(U_{ij}) = \alpha O_i D_j p(U_{ij}) \tag{17}$$

donde α es un parámetro de proporcionalidad, y $p(U_{ij})$ es la *función de disuasión*. Esta función depende de uno o más parámetros que deberán ser calibrados y evalúa el efecto de la distancia en la generacion de viajes . Las expresiones funcionales más habituales son

$$\begin{array}{lcl} p(U_{ij}) = \exp(-\beta U_{ij}) & \rightarrow & \text{función exponencial} \\ p(U_{ij}) = U_{ij}^{-n} & \rightarrow & \text{función potencial} \\ p(U_{ij}) = U_{ij}^{n} \exp(-\beta U_{ij}) & \rightarrow & \text{función combinada} \end{array} \tag{18}$$

El siguiente modelo integra las fases de distribución y asignación.

$$
\begin{array}{ll}
\text{minimizar} & Z = \sum_{l\in\mathcal{A}} \int_0^{f_l} c_l(x)dx - \sum_{\omega\in W} \int_0^{g_\omega} G_\omega^{-1}(x)dx \\
\text{sujeto a} & \sum_j g_{ij} = O_i, \quad \forall i, \\
& \sum_i g_{ij} = D_j, \quad \forall j, \\
& \sum_{p\in P_\omega} h_p = g_\omega, \quad \forall \omega \in W, \\
& \sum_{\omega\in W} \sum_{p\in P_\omega} \delta_{ap} h_p = f_l, \quad \forall l \in \mathcal{A}, \\
& h_p \geq 0, \ \forall p \in P_\omega, \ \forall \omega \in W.
\end{array}
\qquad \text{[TAP-D]}
$$

Suponiendo que la función $p(x)$ es decreciente en el coste de viaje, la solución de TAP-D satisface las condiciones de equilibrio y la ecuación (17). Esto puede ser demostrado empleando las condiciones de optimalidad de KKT.

El caso particular más utilizado del modelo TAP-D es el que usa el *modelo de distribución gravitacional*. Este modelo se obtiene al emplear como función de disuasión la función exponencial. En este caso se obtiene

$$
\begin{aligned}
-\int_0^{g_\omega} G^{-1}(x)dx &= -\int_0^{g_\omega} \left[-\frac{1}{\beta}\log(x) + (d_i + d_j + \alpha')\right] dx \\
&= \frac{1}{\beta} g_\omega (\log g_\omega - 1) + (d_i + d_j + \alpha') g_\omega
\end{aligned}
$$

donde $d_i = \log O_i/\beta$, $d_j = \log D_j/\beta$, y $\alpha' = \alpha/\beta$. La función objetivo es

$$
\sum_{(i,j)\in\mathcal{A}} \int_0^{f_{ij}} c_{ij}(\mathbf{x})d\mathbf{x} + \frac{1}{\beta} \sum_{\omega\in W} g_\omega (\log g_\omega - 1) + \sum_{\omega\in W} (d_i + d_j + \alpha') g_\omega
$$

El término $\sum_{\omega\in W}(d_i + d_j + \alpha')g_\omega$ es constante en el conjunto factible debido a las restricciones (16.a)-(16.b), y puede, por tanto, ser eliminado.

Modelo combinado de asignación, distribución y partición modal

Florian y Nguyen [85] formulan un modelo de equilibrio que integra asignación, distribución y partición modal en un único modelo de optimización

$$
\begin{array}{ll}
\text{minimizar} & Z = \sum_{l\in\mathcal{A}} \int_0^{f_l} c_l(x)dx + \tau \sum_{\omega\in W} g_\omega^a \ln g_\omega^a + \sum_{\omega\in W} g_\omega^b \left(\tau \ln g_\omega^b + U_\omega^b\right) \\
\text{sujeto a} & \sum_j \left(g_{ij}^a + g_{ij}^b\right) = O_i, \quad \forall i, \\
& \sum_i \left(g_{ij}^a + g_{ij}^b\right) = D_j, \quad \forall j, \\
& \sum_{p\in P_\omega} h_p = g_\omega^a, \quad \forall \omega \in W, \\
& \sum_{\omega\in W} \sum_{p\in P_\omega} \delta_{ap} h_p + V_l^b = f_l^a, \quad \forall l \in \mathcal{A}, \\
& h_p \geq 0, \quad \forall p \in P_\omega, \ \forall \omega \in W,
\end{array}
$$

donde

U_ω^b es el coste de viaje en transporte público. Se supone que este coste es independiente de los volúmenes de tráfico.

V_l^b es la contribución del transporte público al flujo en el arco l. En algunos casos puede ser cero para modos de transporte separados de la red de tráfico.

Estos autores mostraron bajo las usuales hipótesis de convexidad que la solución del modelo satisface el criterio de Wardrop, que las demandas por modos satisfacen el modelo gravitacional de la forma

$$g_{ij}^a = a_i b_j \exp(-\beta U_{ij}^a),$$
$$g_{ij}^b = a_i b_j \exp(-\beta U_{ij}^b)$$

y la partición modal viene dada por un modelo logit bimodal con parámetro β.

Marín [163] analiza las propiedades de estos modelos combinados y la metodología para su resolución.

2.5 Algunos problemas MPEC en planificación de transporte urbano

Muchos de los problemas de planificación y de diseño de redes de transporte urbano son formulados mediante un modelo MPEC, debido a que su estructura binivel es adecuada para reflejar el proceso de tomas de decisiones. Los operadores del sistema planifican o diseñan el sistema de transporte teniendo en cuenta el comportamiento de los usuarios ante sus políticas de gestión o inversión. En el nivel superior se mimimizan los costes (sociales, económicos, etc) derivados de las políticas de los operadores y en el nivel inferior se describe el comportamiento de los usuarios en el sistema de transporte intervenido. En este apartado describimos los modelos más notables.

Problema continuo de diseño de redes

El problema de diseño de redes trata el problema de modificar la infraestructura de transporte, mediante la creación de nuevos arcos o mejorando la capacidad de los existentes, de modo que se maximice el beneficio social (reducción de la congestión) y/o se minimice el coste del diseño. La versión continua de este problema aparece cuando las variables de diseño toman valores continuos (por tanto no se contempla la posibilidad de añadir nuevos arcos) y éstas representan la mejora de la capacidad existente en los arcos. En el nivel inferior aparece el modelo de asignación de tráfico formulado (usualmente) mediante el TAP, aunque autores como Marcotte [159] emplean el TAP-VIP. Para concretar la situación consideramos que el problema interior es un problema con demanda fija, utilizando el principio DUE para efectuar la asignación, formulado en el espacio $\Omega_{\mathbf{f}}$, esto es, consideramos el modelo TAP-VIP($\mathbf{c}, \Omega_{\mathbf{f}}$).

Las variables del modelo son

- ◇ nivel inferior: el vector de flujo en los arcos $\mathbf{f} \in \Re^{|\mathcal{A}|}$
- ◇ nivel superior: las ampliaciones de las capacidades $\mathbf{x} \in \Re^{|\mathcal{A}|}$

El vector de capacidades $\mathbf{x}$ constituye un subconjunto de parámetros de las funciones de coste de viaje en los arcos, esto es, $\mathbf{c}(\mathbf{x}, \mathbf{f})$. Esta dependencia funcional pone de manifiesto como las nuevas infraestructuras determinan nuevos costes de transporte.

El modelo asume que existe una función $s(\mathbf{x})$ que proporciona los costes de inversión y operación asociado al diseño de red definido por la variable $\mathbf{x}$. Por otro lado, se dispone de un presupuesto para la realización de la nueva infraestructura, que denotamos por B y esta cantidad no puede ser sobrepasada. Este requerimiento es tenido en cuenta por la *restricción presupuestaria*

$$s(\mathbf{x}) \leq B.$$

Supongamos que existe otro conjunto de restricciones (técnicas) para las variables de diseño. Este conjunto lo denotamos por X, entonces el conjunto factible para las variables de diseño es

$$\tilde{X} := \{\mathbf{x} \in X \mid s(\mathbf{x}) \leq B\}$$

El problema de diseño de redes se formula

$$\begin{array}{ll} \text{minimizar} & Z = \sum_{l\in\mathcal{A}} c_l(\mathbf{x},\mathbf{f}) f_l \\ \text{sujeto a} & \mathbf{x} \in \tilde{X}, \\ & \mathbf{f} \text{ resuelve TAP-VIP}(\mathbf{c}(\mathbf{x},\cdot),\Omega_{\mathbf{f}}). \end{array} \qquad \text{[NDP]}$$

Algunos autores, basándose en la variable dual óptima asociada a la restricción presupuestaria (que denotamos μ^*), dan la formulación alternativa

$$\begin{array}{ll} \text{minimizar} & Z = \sum_{l\in\mathcal{A}} c_l(\mathbf{x},\mathbf{f}) f_l + \mu^* s(\mathbf{x}) \\ \text{sujeto a} & \mathbf{x} \in X, \\ & \mathbf{f} \text{ resuelve TAP-VIP}(\mathbf{c}(\mathbf{x},\cdot),\Omega_{\mathbf{f}}). \end{array} \qquad \text{[NDP]}$$

Un resultado que puede ilustrar la dificultad en la resolución de este problema es la llamada *paradoja de Braess* que afirma que no siempre la adición de un nuevo arco a la red, y por tanto un incremento de la capacidad de la red de transporte, conduce a una reducción de los tiempos de viaje que emplea cada usuario. Un ejemplo numérico que ilustra tal hecho se pude consultar en Florian y Hearn [82].

Esto es debido a que los usuarios eligen su ruta atendiendo exclusivamente a minimizar su tiempo de viaje (el llamado primer principio de Wardrop) y no tienen por objetivo reducir el tiempo total de viaje en toda la red de transporte. La paradoja aparece porque un incremento de la capacidad de la red siempre conducirá a la *posibilidad* de reducir el tiempo total de transporte si los usuarios eligiesen la ruta de acuerdo al segundo prinicipio de Wardrop. Notar que el NDP y el TAP-SE tienen la misma función objetivo.

Estimación de matrices origen-destino

El problema de estimar la matriz de viajes origen-destino es fundamental en la planificación del transporte. Es importante señalar que el problema NDP emplea esta matriz como dato para el modelo. Existe multitud de métodos para tal fin, pero el inconveniente de muchos de ellos es que se basan en la elaboración de encuestas domiciliarias y ello supone una gran coste económico. Por este motivo se han desarrollado métodos, como el expuesto aquí, para basar la estimación de estas matrices en información más económica de conseguir como son los volúmenes de tráfico obtenidos mediante cordones. Este modelo asume que un conjunto actualizado de volúmenes de tráfico está disponible. Estas observaciones son denotadas por $\hat{f}_l$, con $l \in \hat{\mathcal{A}} \subset \mathcal{A}$. Suponemos que una matriz O-D $\hat{\mathbf{g}} = \{\hat{g}_\omega\}_{\omega\in\hat{W}\subset W}$ desactualizada o que ha sido obtenida por otros métodos está disponible.

Notar que la región factible de los flujos en los arcos $\Omega_{\mathbf{f}}$ está parametrizada por la matriz O-D $\bar{\mathbf{g}}$. Este hecho lo resaltaremos denotando esta región por $\Omega_{\mathbf{f}}(\bar{\mathbf{g}})$.

Las variables del modelo son

- ◇ nivel inferior: el vector de flujo en los arcos $\mathbf{f} \in \Re^{|\mathcal{A}|}$ supuesto que se conoce la matriz O-D $\bar{\mathbf{g}}$
- ◇ nivel superior: la matriz O-D $\bar{\mathbf{g}} \in \Re^{|W|}$

El problema de estimación de las matrices O-D se formula como

$$\begin{array}{ll} \text{minimizar} & Z = \theta_1 F_1(\mathbf{f},\hat{\mathbf{f}}) + \theta_2 F_2(\bar{\mathbf{g}},\hat{\mathbf{g}}) \\ \text{sujeto a} & \bar{\mathbf{g}} \in \mathcal{G}, \\ & \mathbf{f} \text{ resuelve VIP-TAP}(\mathbf{c}(\cdot)\Omega_{\mathbf{f}}(\bar{\mathbf{g}})). \end{array} \qquad \text{[DAM]}$$

donde F_1 y F_2 son dos métricas que miden la discrepancia entre los datos observados y los predichos por el modelo, y $\mathcal{G}$ es un conjunto de restricciones para las matrices. $\mathcal{G}$ consiste generalmente en un conjunto de cotas superiores e inferiores para los pares O-D.

Chen y Florian [48] emplean la clásica estimación mínimo cuadrática, esto es

$$F_1(\mathbf{f}, \hat{\mathbf{f}}) = \sum_{l \in \hat{\mathcal{A}}} (f_l - \hat{f}_l)^2,$$
$$F_2(\bar{\mathbf{g}}, \hat{\mathbf{g}}) = \sum_{\omega \in \hat{W}} (\bar{g}_\omega - \hat{g}_\omega)^2$$

Bard [11] da una revisión de distintas funciones objetivos que han sido consideradas en la literatura. Los parámetros θ_1 y θ_2 miden el nivel de confianza en nuestras observaciones, y pueden ser ajustados mediante técnicas de programación matemática multiobjetivo.

Notar que existen dos diferencias esenciales entre el NDP y el DAM. En el NDP la función objetivo del nivel inferior está parametrizada por las variables del nivel superior y la región factible es independiente de éstas. Por el contrario el DAP tiene parametrizada la región factible del nivel inferior pero no la función objetivo. La segunda diferencia está en la naturaleza de las función objetivo del nivel superior.

Modelos de gestión de tráfico

En este apartado analizamos tres modelos de gestión de tráfico. Estos modelos persiguen reducir la congestión, mejorar la eficiencia de los viajes urbanos, y en algunos casos influenciar en el uso de transporte público. En estos problemas no hay una modificación de la infraestructura de la red de transporte como ocurría en el problema NDP, sino unos mecanismos de control de tráfico que permiten mejorar la eficiencia del sistema de transporte. El primer modelo que analizamos tiene como variable de control la tarifación de la congestión de los arcos de la red. El segundo modelo está diseñado para establecer la regulación semafórica del tráfico y utiliza como variable de control la distribución de tiempo en verde en las intersecciones. El tercer modelo aborda simultáneamente ambos problemas.

Modelo de tarifación de la congestión. Los usuarios eligen su ruta con el fin de minimizar su tiempo/coste de viaje. Este deseo no coincide con el objetivo del sistema, derivado del segundo principio de Wardrop, de minimizar el tiempo total de viaje en toda la red de transporte. Este modelo MPEC fuerza a los usuarios a que se comporten de acuerdo al segundo principio de Wardrop. El mecanismo de control que el operador dispone para conseguir el equilibrio bajo el punto de vista del sistema es un conjunto de tasas en los arcos de la red que denotamos por $\beta \in \Re^{|\mathcal{A}|}$. Las tasas no están restringidas en signo, lo que permite representar mediante valores negativos subsidios a ciertos modos de transporte público. Estas tasas modifican la percepción que un usuario tiene de los costes de viaje en la red. Estos nuevos costes son $\mathbf{c}(\mathbf{f}) + \beta$, y hacen que el patrón del flujo represente un óptimo bajo el punto de vista del sistema aunque las rutas sean elegidas bajo el punto de vista del usuario para estos nuevos costes.

El problema a resolver es

$$\begin{array}{ll} \text{minimizar} & Z = \sum_{l \in \mathcal{A}} c_l(\mathbf{f}) f_l \\ \text{sujeto a} & \mathbf{f} \in \mathcal{F}, \\ & (\mathbf{c}(\mathbf{f}) + \beta)^T (\hat{\mathbf{f}}^- \mathbf{f}) \geq 0, \quad \forall \hat{\mathbf{f}} \in \Omega_{\mathbf{f}}. \end{array}$$

El conjunto $\mathcal{F}$ suele estar formado por un conjunto de *restricciones laterales* que modelizan la capacidad de la red.

Es sabido que una solución al anterior problema es poner como tasa la diferencia entre el coste medio $c_l(f_l^*)$ y el coste marginal $c_l(f_l^*) + c_l'(f_l^*) f_l$, donde $\mathbf{f}^*$ es el flujo en equilibrio bajo el sistema, esto es, $\beta_1^* = \nabla \mathbf{c}(\mathbf{f}^*)\mathbf{f}^*$. Esta solución se le denomina *tasa del coste social marginal*. Otra solución es la denominada *tasa del coste del sistema* que consiste en dar a cada usuario en cada arco un subsidido

igual a la cola causada en ese arco bajo el punto de vista del sistema, esto es $\beta_2^* = -\mathbf{c}(\mathbf{f}^*)$. Esto pone de manifiesto que el conjunto de soluciones no es único. Bajo ciertas condiciones se puede garantizar que este conjunto es un conjunto convexo no vacío. Hearn y Ramana [122]

El verdadero problema no es tanto la obtención de una tarifación de arcos que conduzca al equilibrio bajo el punto de vista del sistema sino su implementación práctica. Ejemplos de implementaciones prácticas de un sistema de peaje se tienen al pagar por atravesar ciertos cordones que rodean determinadas zonas de la ciudad, como el centro urbano, periferia, etc. Por tanto, el problema es encontrar un sistema de tarifación válido que optimice una nueva función objetivo. Ejemplos de estas funciones son: i) minimizar la cantidad total recaudada por los peajes, ii) minimizar el máximo peaje sobre un arco, iii) minimizar el número de estaciones recaudatorias, iv) encontrar una sistema de tarifación con suma cero o v) combinar iii) y iv). Estos ejemplos son analizados en Hearn y Ramana [122]. Este problema puede ser visto como un problema en tres niveles jerárquicos.

Modelo de regulación semafórica del tráfico. Este problema tiene una larga historia que ha producido multitud de métodos, entre otros los desarrollados por Cantarella y Sforza [38] y Smith y Van Vuren [215]. Muchos de estos métodos consideran de una forma más o menos precisa el efecto que produce el sistema de señalización en los cambios de las rutas de los usuarios. Una primera clase de modelos que puede ser considerada como una resolución heurística de la formulación binivel del problema son los *métodos iterativos de optimización y asignación*. Estos métodos se basan en optimizar en cada intersección (localmente) la regulación semafórica de modo que se maximice el número de usuarios que la atraviesan para el flujo actual. Para dicha estrategia de control se calcula, en la fase de asignación, un nuevo flujo en equilibrio con el que se repetirá el procedimiento de optimización-asignación.

La formulación binivel del problema determina las proporciones de tiempo en verde en los controles semafóricos de modo que se minimice el tiempo de viaje en la red (o en una parte de la misma) teniendo en cuenta como los usuarios cambian de ruta en función de esta regulación. Las variables del modelo son

- ◇ nivel inferior: el vector de flujo en los arcos $\mathbf{f} \in \Re^{|\mathcal{A}|}$
- ◇ nivel superior: la proporción de tiempo en verde en los semáforos $\rho \in \Re^n$, donde n es el número de controles semafóricos.

Las variables del nivel superior influyen en los tiempos de viajes en los arcos. Estas variables son tenidas en cuenta como parte de la parametrización de estos costes. Más concretamente, el tiempo en una intersección regulada por un semáforo es la suma del tiempo empleado en la operación parada-arranque más la espera en el mismo. Webster [234] desarrolla un fórmula teniendo en cuenta estos dos efectos bajo la hipóteis de que la llegada a la cola del semáforo es una variable aleatoria Poisson de tasa de llegada constante. Su expresión analítica es

$$c_l(\rho_l, f_l) = \frac{9}{20}\left(\frac{\tau(1-\rho_l)^2}{1-f_l/k_l} + \frac{f_l}{k_l\rho_l(k_l\rho_l - f_l)}\right)$$

donde

- f_l es el flujo de entrada,
- ρ_l es la proporción efectiva del semáforo en verde,
- k_l capacidad del arco (flujo de saturación),
- τ tiempo empleado por cada ciclo.

El modelo MPEC para el problema de regulación semafórica del tráfico se formula como

$$\begin{array}{rl} \text{minimizar} & Z = \sum_{l\in\hat{\mathcal{A}}\subset\mathcal{A}} c_l(\rho, f_l) f_l \\ \text{sujeto a} & \rho \in \mathcal{P}, \\ & \mathbf{f} \text{ resuelve VIP-TAP}(\mathbf{c}(\rho,\cdot), \Omega_{\mathbf{f}}). \end{array}$$

donde $\mathcal{P}$ representa la compatibilidad de los distintos controles semafóricos y $\mathcal{F}$ las restricciones de capacidad de los arcos de la red.

Este problema tiene la misma estructura que el NDP, sólo que aquí no hay inversión en la infraestructura de la red y por ese motivo dichos costes de inversión no aparecen en la función objetivo.

Modelo combinado de control semafórico y tarifación de la congestión. Varios autores han considerado simultáneamente ambas acciones de gestión, entre otros Smith y otros [216], Patriksson y Rockafellar [199].

Supongamos que entre las posibles acciones de gestión la autoridad considera las decisiones (ρ, β) y desea optimizar una función $\varphi : \mathcal{P} \times \Re^{|A|} \times \Re^{|A|} \mapsto \Re$. Esta función puede incluir alguna medida de eficacia en la red, así como el beneficio/coste de las acciones.

Este modelo puede ser fomulado como

$$\begin{array}{ll} \text{minimizar} & Z = \varphi(\rho, \beta, \mathbf{f}) \\ \text{sujeto a} & \rho \in \mathcal{P}, \\ & \mathbf{f} \in \mathcal{F}, \\ & \mathbf{f} \text{ resuelve VIP-TAP}(\mathbf{c}(\rho, \cdot) + \beta, \Omega_{\mathbf{f}}), \end{array} \qquad \text{[MPEC-TAP]}$$

donde $\mathcal{P}$ representa la compatibilidad de los distintos controles semafóricos y $\mathcal{P}$ ciertas restricciones de capacidad en los arcos de la red.

Modelo para la planificación de frecuencias en redes de transporte público

Esta aplicación considera el problema de establecer las frecuencias óptimas para las líneas de transporte público. Este problema fue abordado por Marín [166] en un contexto de optimización de un solo nivel, en el que se optimizaba simultáneamente el interés de los usuarios y del operador del sistema.

En este apartado describimos el modelo de Constantin y Florian [55] que lo formula en un contexto binivel.

Consideremos una red de transporte público $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ donde $\mathcal{A}$ es el conjunto de secciones de ruta de la red de transporte público, y $\mathcal{N}$ es el conjunto de paradas. Denotamos mediante $\mathcal{L}$ el conjunto de líneas.

El operador del sistema desea establecer una asignación de la flota de N vehículos al conjunto de líneas $\mathcal{L}$ de modo que se minimice el tiempo total de viaje en la red de transporte. Este problema tiene una estructura binivel. En el nivel superior el operador diseña los servicios de la red y por otro lado los usuarios, en el nivel inferior, eligen la estrategia de viaje en la red diseñada.

Las variables del modelo son

- ◇ nivel inferior: el vector de flujo en las secciones de ruta $\mathbf{V} \in \Re^{|\mathcal{A}|}$, y el vector de variables dicotómicas que representan las línea comúnes a cada sección de ruta, esto es, $x_l^s \in \{0, 1\}$ para todo $s \in \mathcal{A}$ y $l \in A_s$
- ◇ nivel superior: las frecuencia para cada línea, esto es $\{\phi_l\}_{l \in \mathcal{L}}$

El tiempo total de viaje en la red, para un conjunto de frecuencias dado y para su correspondiente estrategia de viaje viene dado por

$$\sum_{s \in \mathcal{A}} C_s V_s,$$

donde C_s es el tiempo de viaje (tiempo de espera más tiempo en el vehículo) en cada sección de ruta, y V_s es el número de usuarios en la sección de ruta $s \in \mathcal{A}$.

El número de vehículos de una línea y su relación con la frecuencia de la línea es $\phi_l = \frac{T_l}{n_l}$ donde T_l es el tiempo de viaje del origen al final de la línea, n_l el número de vehículos y ϕ_l la frecuencia de la línea. Los requerimientos del tamaño de la flota son tenidos en cuenta mediante la restricción

$$\sum_{l \in \mathcal{L}} \frac{T_l}{\phi_l} \leq N.$$

Por otro lado se exige un nivel de servicio mínimo para cada línea, esto es

$$\phi_l \geq \underline{\phi}_l, \quad \forall l \in \mathcal{L}.$$

El nivel superior fija las frecuencias de las líneas que sólo pueden tomar un conjunto discreto de valores asociado a valores enteros del número de vehículos en cada línea. Denotamos mediante Φ el conjunto de frecuencias factibles, es decir

$$\Phi := \left\{ (\phi_l) \in \Re^{|\mathcal{L}|} : \quad \phi_l \geq \underline{\phi}_l, \ \forall l \in \mathcal{L}; \quad \sum_{l \in \mathcal{L}} \frac{T_l}{\phi_l} \leq N; \quad \frac{T_l}{\phi_l} \in \mathbb{N}, \ \forall l \in \mathcal{L} \right\}$$

donde $\mathbb{N}$ es el conjunto de números naturales (incluido el cero).

En el nivel inferior los usuarios minimizan su tiempo de viaje mediante la elección de una adecuada estrategia de viaje.

Este modelo se formula como

$$\begin{array}{ll} \text{minimizar} & Z = \sum_{s \in \mathcal{A}} C_s V_s \\ \text{sujeto a} & \phi \in \Phi, \\ & (\mathbf{C}, \mathbf{V}) \text{ resuelve el TEAP.} \end{array} \qquad \text{[NDP-TEAP]}$$

3 Sumario de la tesis

Los temas de investigación cubiertos por esta tesis doctoral han estado enmarcados por la programación matemática por un lado y por su aplicación a la planificación del transporte urbano por otro. En este contexto, se han desarrollado nuevos modelos de transporte, nuevos métodos para su resolución y nuevas aplicaciones. La investigación realizada ha cubierto tanto los aspectos teóricos, como computacionales de los problemas. Los resultados obtenidos se han agrupados en los siguientes seis capítulos

§1 : Modelos de equilibrio con modos combinados.

§2 : La clase de algoritmos CG/SD en programación convexa diferenciable: análisis de la convergencia.

§3 : La clase de algoritmos CG/SD: estudio computacional.

§4 : Calibración de parámetros y estimación de matrices O-D en modelos combinados.

§5 : Capacidad y tarifación de aparcamientos disuasorios: un problema de diseño de redes.

§6 : Diseño de intercambiadores multimodales urbanos.

La tesis se puede agrupar en dos partes. La primera constituida por los capítulos uno, dos y tres trata temas de programación matemática en un solo nivel y la segunda, formada por el resto de capítulos, aborda problemas de programación matemática binivel. En el capítulo 1 se formula un modelo combinado (de asignación y partición modal) para modelizar viajes que emplean más de un modo de transporte. Este modelo se denomina TAP-M. En el segundo capítulo se generaliza la clase de métodos de descomposición simplicial/generación de variables en el contexto de programación matemática convexa diferenciable. En él se estudian las propiedades de convergencia finita y asintótica. En el tercer capítulo se evalúa computacionalmente estos métodos sobre redes no lineales uni producto y sobre la versión simétrica del modelo desarrollado en el capítulo 1. En el capítulo 4 se generaliza el problema de estimación de matrices O-D (DAM) al problema de estimar los parámetros y matrices O-D de un modelo combinado (CDAM). En este capítulo se sientan las bases para la elaboración de algoritmos heurísticos. Esta discusión es analizada a través de la aplicación al modelo TAP-M. En el capítulo 5 nos planteamos un nuevo problema de diseño de redes-tarifación de la congestión para calcular las tarifas y capacidades de un conjunto de aparcamientos disuasorios. Su resolución se basa en la técnica heurística del *simulado recocido*. Concluimos nuestras discusiones con el análisis de un modelo (mixto) de diseño de redes para el problema de los llamados intercambiadores multimodales urbanos y problemas de expansión de la red de transporte público. Se adaptan técnicas clásicas para la elaboración de algoritmos heurísticos a este problema.

§1 : **Modelos de equilibrio con modos combinados**

Los viajes combinados son aquellos viajes que emplean varios modos de transporte. El ejemplo más usual es el denominado viaje *park'n ride* que consiste en emplear el coche privado para viajar del lugar de origen a una parada de transporte público (por ejemplo una estación de metro, cercanías, tren regional, etc.) aparcar, y completar el intinerario empleando una o varias líneas de transporte público.

En este capítulo se formula mediante desigualdades variacionales un modelo combinado de asignación y partición modal donde los usuarios eligen ruta, modo de transporte y estación donde efectuar el intercambio modal. Se ha introducido un modelo de demanda *logit anidado* basado en dos niveles, para modelizar primeramente la elección modal y posteriormente el nodo de intercambio. Este modelo puede ser considerado una extensión al contexto de desigualdades variacionales del desarrollado en Fernández y otros [73] y lo denominaremos TAP-M.

También se ha adaptado la descomposición simplicial y una generalización del algoritmo de Evans para este problema de desigualdades variacionales. Se han dado condiciones necesarias para su convergencia y se ha estudiado computacionalmente el caso simétrico del modelo propuesto.

En la sección 1.5 se ilustra su uso para el diseño paramétrico de intercambiadores multimodales urbanos.

Marín y García [168] presentaron un análisis de las características que debían de tener los modelos para diseñar intercambiadores multimodales urbanos. Este trabajo constituye el punto de partida del modelo aquí desarrollado. La sección 1.5 fue presentada al *III Congreso Nacional de Transporte* celebrado en Barcelona en junio de 1998. (García y Marín [99])

§2 : **La clase de algoritmos CG/SD en optimización convexa diferenciable: análisis de la convergencia**

En este capítulo generalizamos la clase NSD de Larsson y otros [154, 141] elaborada para resolver el CDP(f, X). La principal diferencia entre ambas clases de algorimos radica en el principio de generación de columnas. Los algorimos de la clase NSD obtienen las columnas como solución (truncada) de una aproximación cuadrática del problema original. En esta nueva clase, denominada de generación de columnas /descomposición simplicial (CG/SD), la columna es obtenida mediante la aplicación de varias iteraciones de un algoritmo cerrado y de descenso (Zangwill [248]) a una función de mérito que puede ser la propia función objetivo. Esta visión reemplaza el papel de los subproblemas CGP

por el de algoritmos para generar columnas. La clase NSD es un caso particular de CG/SD donde los algoritmos para obtener columnas son definidos a través de subproblemas. La segunda diferencia radica en la gran libertad en la definición de la región factible del RMP. Esta nueva formulación emplea conjuntos compactos y convexos generales, no necesariamente conjuntos compactos poliédricos.

Los algoritmos CG/SD (como se pondrá en evidencia en el capítulo 3) se pueden interpretar como un medio de acelerar la convergencia de un algoritmo de optimización de puntos factibles mediante un esquema de descomposición simplicial (generalizada). En este contexto nos hemos planteado si las propiedades de identificación de restricciones y convergencia finita de la sucesión generada por el algoritmo de optimización es heredada por la sucesión generada por los problemas RMP.

La formulación de esta clase de algoritmos ha sido un proceso largo. En García y Marín [96] se obtuvieron las adaptaciones del algoritmo de Frank-Wolfe [88] y Evans [72] para el problema TAP-M en su versión simétrica. Comprobamos que la adaptación de Evans tiene mejores propiedades de convergencia. Este resultado es conocido en otro tipo de aplicaciones como Evans [72], Boyce [29], Williams y otros [237]. En un principio optamos por mejorar el algoritmo de Evans introduciendo la *modificación de Horowitz* [129] que cambia la función objetivo del problema de búsqueda unidimensional. Esta modificación no tiene garantizada, la convergencia como constatamos computacionalmente para el TAP-M, y nos planteamos dos soluciones alternativas. La primera fue continuar con el camino iniciado e introducir las modificaciones de Huang y Lam [130] para garantizar la convergencia. En esencia consiste en aplicar la iteración original de Evans cuando se llega a un *punto muerto* y mientras tanto utilizar la modificación de Horowitz. La otra solución (que fue el camino elegido) era adaptar la descomposición simplicial para modelos combinados. El primer avance fue reemplazar el problema lineal de la fase de CGP en SD por los subproblemas de linelización parcial de Patriksson [193]. Este marco contenía el algoritmo de interés donde el CGP era un subproblema de tipo Evans. Los resultados se presentaron en el congreso de *ISMP* 16^{th} *International Symposium on Mathematical Programming* en agosto de 1997 (García y Marín [96]). Los resultados presentados están en el trabajo García y otros [103]. Esta clase es equivalente (sin la prolongación de las columnas) a la NSD. La versión presentada en la tesis coincide con la dada en el trabajo García y otros [104] y es la presentada en el congreso *ISMP 2000 17th International Symposium on Mathematical Programming* (García y otros [105]).

§3 : **La clase de algoritmos CG/SD: estudio computacional**

En este capítulo se hace un estudio computacional de la clase de algoritmos CG/SD para dos problemas de flujos en redes. El primero es un problema uniproducto de flujo en redes no lineales y el segundo es la versión simétrica del TAP-M.

El objetivo de este capítulo, además de probar la eficiencia de los métodos CG/SD, es entender el por qué de esta mejora con respecto a las versiones clásicas de métodos de descomposición simplicial tales como RSD y SD.

El contenido aquí expuesto coincide con el trabajo García y otros [106]. Los resultados de este capítulo fueron aceptados para ser presentados en el congreso 8^{th} *EURO Working Group on Transportation* (García y otros [107]) y en el *TRISTRAN IV* (García y otros [109]). Otras aplicaciones de la clase CG/SD, diferentes a las analizadas en esta tesis, se encuentran en el trabajo García y otros [108].

§4 : **Calibración de parámetros y estimación de matrices O-D en modelos combinados**

Este capítulo está dedicado al problema de calibrar los parámetros y estimar (actualizar) la matriz O-D en los modelos combinados de equilibrio. Este problema es tratado a través del estudio del modelo combinado TAP-M (desarrollado en el capítulo 1).

Se formula mediante la programación matemática binivel un modelo, denominado CDAM, que

unifica ambos problemas en uno solo. El nivel superior decide la combinación del vector de parámetros y de la matriz O-D, de modo que el modelo combinado reproduzca lo más fielmente posible toda la información que se dispone (aforos, matrices desactualizadas, resultados de encuesta, etc.) En este trabajo se demuestra la existencia de soluciones, incluso cuando el conjunto de aforos sea inconsistente o incompleto. Este resultado está basado en el trabajo de Chen y Florian [48] que demuestran la existencia de soluciones para el problema de estimar las matrices O-D en el problema de asignación de tráfico (DAM). En este contexto, se requiere emplear una adecuada formulación de las condiciones de equilibrio para estudiar la dependencia continua entre los flujos en equilibrio y los parámetros del modelo.

La motivación para la elaboración del CDAM es doble. Por un lado se busca una mejor estimación tanto de los parámetros como de la matriz (situación que es ilustrada mediante un ejemplo simple) basándose en el hecho de considerar ambos problemas simultáneamente y por otro lado se desea obtener dichas estimaciones empleando toda la información disponible tales como aforos, resultados de encuestas, matrices desactualizadas, etc. Esta flexibilidad permite realizar la estimación sin necesidad de recurrir a encuestas y por tanto a un procedimiento muy económico.

La sección dedicada a la calibración del modelo y los problemas derivados de la *sobreespecificación* de los parámetros fue presentado en el congreso *EURO XV–INFORMS XXXIV Joint International Meeting* (García y Marín [95]) celebrado en Barcelona en 1997. La primera formulación conjuta de los problemas de estimación y calibración, así como la adaptación de varios algoritmos heurísticos, distintos de los aquí desarrollados, fue presentada en el *III Congreso de Ingeniería del Transporte* celebrado en 1998 en Barcelona (García y Marín [98]). La versión actual fue presentada en el congreso *6^{th} EURO Working Group on Transportation* (García y Marín [97]), la cual ha sido enviada para su publicación.

§5 : **Capacidad y tarifación de aparcamientos disuasorios: un problema de diseño de redes**

En este capítulo se aborda el problema de diseñar aparcamientos disuasorios, donde los usuarios puedan aparcar sus coches y completar su viajes en transporte público. Hemos considerado como variables de interés para este problema las tarifas de los aparcamientos y sus capacidades. Suponiendo que la localización de los aparcamientos ya ha sido decidida, el problema es un problema continuo de diseño de redes. En el nivel superior se fija un plan de aparcamientos, definido por las variables capacidad y tarifa de los aparcamientos, y en el nivel inferior los usuarios eligen su ruta, modo de transporte y aparcamientos en la red de transporte diseñada. El modelo asume restricciones de inversión y el objetivo es disminuir la congestión en parte de la red de transporte (por ejemplo, en la red de tráfico). Este modelo puede considerarse un híbrido entre un problema puro de diseño de redes (determinar la capacidad de ciertos arcos) y un problema de tarifación de la congestión (tarifación de los aparcamientos).

En la red multimodal los aparcamientos están representados por arcos y su función de congestión representa el coste generalizado de aparcamiento en función de la capacidad, número de usuarios, tarifa, distancia a la parada, etc. El problema es encontrar la parametrización adecuada de estas funciones. (Dos parámetros por aparcamiento).

Este trabajo ha sido presentado en *X Congreso Latino Americano de Invetiagción de Operaciones*, celebrado en septiembre del 2000 en México D.F (García y Marín [101]) y ha sido enviado para su publicación.

§6 : **Metodología para el diseño de intercambiadores multimodales urbanos**

En este capítulo se aborda el diseño de intercambiadores multimodales urbanos en un contexto de planificación estratégica.

Se asume que se está gestionando un sistema de transporte público formado por dos redes de trans-

porte. La principal (por ejemplo cercanías-metro) ofrece viajes de larga distancia, mientras que la red secundaria (formada por ejemplo autobuses locales) tiene el objetivo de alimentar a las líneas principales. Se supone que nuevas líneas de la red principal han sido diseñadas y se desea localizar sobre ella nuevas estaciones. Las estaciones que disponen facilidades adicionales como aparcamientos disuasorios o que están alimentadas por la red secundaria se denominan intercambiadores multimodales urbanos. El modelo que planteamos resuelve la localización de los intercambiadores, la capacidad y tarifas de sus aparcamientos disuasorios y el tipo de diseño adecuado de la red de alimentación. Este es un problema de diseño de redes mixto, es decir, con variables discretas (localización de intercambiadores y diseño de la red secundaria) y continuas (precio y capacidades).

El modelo ha sido formulado mediante programación matemática binivel. La complejidad del problema junto al horizonte temporal de la planificicación han conducido a un nuevo modelo de equilibrio multimodal entre oferta y demanda. Las diferencias respecto al TAP-M va en dos direcciones. La primera es un nuevo nivel en el modelo (de demanda) logit anidado con el fin de recoger la elección, de aparcar en el intercambiador o fuera de él, que hacen los usuarios. La segunda diferencia está en el modelo de red de transporte (oferta de transporte). En este modelo no se tiene en cuenta como nuestras variables de diseño afectan a la congestión, es decir, se ha considerado un nivel de congestión independiente de nuestras variables de diseño. Se han considerado dos formulaciones del modelo una mediante programación matemática y otra mediante una formulación de *tipo punto fijo*, así mismo se ha adaptado un algoritmo de Gauss-Seidel para la resolución de la segunda formulación.

La complejidad del modelo nos ha hecho considerar únicamente el problema de localizar los intercambiadores y elegir el diseño de la red de acceso (modelo de programación binivel no lineal entera). Hemos aplicado tres métodos heurísticos para resolverlo. El primer método está basado en los algoritmos *golosos* (progresivo y regresivo), el segundo es de *búsqueda local* o de *intercambio* y el tercero es una versión discreta del *simulado recocido*. (Ver por ejemplo Nemhauser y Wolsey [178]).

Este trabajo ha sido presentado en el congreso 7^{th} *EURO Working Group on Transportation* celebrado en 1999 en Espoo, Finlandia (García y Marín [100]) y ha sido publicado en el libro *Mathematical Methods on Optimization in Transportation Systems* de Kluwer Academic Publishers. García y Marín [102].

Capítulo 1

Modelos de equilibrio con modos combinados

Resumen

En este capítulo desarrollamos un nuevo modelo para el problema de asignación en equilibrio en redes multimodales con modos combinados (TAP-M). TAP-M está formulado en base a un modelo genérico de asignación de usuarios en redes de transporte público, de un modelo general de asignación de tráfico y de un modelo de demanda logit anidado. Se han considerado dos niveles de anidamiento. En el primer nivel se elige el modo de transporte entre las alternativas consideradas que se clasifican en puras (un único modo de transporte) y combinadas (varios modos de transporte). Para los viajes combinados se introduce un segundo nivel de anidamiento, para representar la elección del nodo de transferencia entre las redes de tráfico y transporte público.

Se plantean las condiciones de equilibrio para las tres elecciones consideradas: elección de ruta, elección de modo y elección de nodo de transferencia, y posteriormente son formuladas matemáticamente mediante un problema de desigualdades variacionales en el espacio de flujo en los *hipercaminos*. En el caso de que los costes sean simétricos este modelo es formulado mediante un problema de optimización. (Apéndice III)

Se han desarrollado dos algoritmos de generación de columnas/descomposición simplicial para este modelo. El primero es la adaptación de la clásica descomposición simplicial restringida y el segundo es el algoritmo de linealización parcial (Evans [72]) integrado en un esquema de descomposición simplicial. Se ha considerado una condición suficiente para la convergencia de estos algoritmos.

Se ha especializado la adaptación de los algoritmos anteriores al caso de que el problema variacional pudiera ser formulado en el espacio de flujo en los arcos. En este caso se muestra, basándose en el hecho de que la función de demanda logit anidada es separable, que el RSD aplicado al problema de tráfico es equivalente a la adaptación del RSD para el TAP-M.

Se han incluido pruebas computacionales para el caso simétrico del modelo, obteniendo que la adaptación tipo Evans posee mejores propiedades de convergencia.

El capítulo finaliza con una demostración de como puede ser empleado el TAP-M en el diseño paramétrico de intercambiadores multimodales urbanos. En esta sección se ha considerado una red de transporte de prueba y sobre ella se muestra cómo el modelo permite evaluar ciertas intervenciones.

Palabras clave: Modelos de equilibrio en redes multimodales con modos combinados, modelos de gestión de transporte urbano, descomposición simplicial, desigualdades variacionales, intercambiadores multimodales urbanos, diseño de redes.

1.1 Introducción

La atención de los investigadores hacia la modelización de redes urbanas se ha visto incrementada recientemente por un auge de los programas de gestión del tráfico y de su congestión. Muchos de estos modelos son formulados mediante problemas de optimización con restricciones de equilibrio. El problema interior es un problema de asignación a redes en equilibrio y el problema exterior toma las decisiones buscando mejorar la eficiencia de la red de transporte.

En la planificación estratégica de sistemas urbanos de transporte los llamados *modelos combinados* son una herramienta adecuada para describir el comportamiento de los usuarios (en el nivel interior) de la red de transporte. En la actualidad se puede considerar satisfactoria la investigación en los modelos combinados con un único modo de transporte. Se han desarrollado formulaciones mediante desigualdades variacionales y modelos de optimización, modelos de asignación determinista y estocástica que han dado excelentes resultados en las aplicaciones. Evans [72], Florian y otros [86], Erlander [71], LeBlanc y Farhangiam [145], Ludgren y Patriksson [151]. Revisiones del estado de arte en estos modelos se pueden encontrar en Boyce [29], Fernández y Friesz [74].

Los problemas de diseño de redes (NDP) que están focalizados en la tarifación de los servicios de transporte público, la construcción de una nueva línea de transporte público o el establecimiento de frecuencias de las líneas, requieren de la representación de la red multimodal. Esto ha motivado la extensión de los modelos combinados con un único modo de transporte a su forma más general de modelos combinados multiusuario-multimodales. Estos modelos tienen la ventaja añadida de que pueden recoger las interacciones entre modos o la representación de varios tipos de usuarios. Por ejemplo, Ferrari [75] propone un modelo de gestión de transporte urbano donde la autoridad tiene el control sobre decisiones tales como tarifación de la red viaria, precios y características del servicio del transporte público. Este modelo está formulado mediante la programación binivel, el nivel inferior está definido por un modelo combinado multimodal.

Importantes avances han sido realizados en los pasados veinte años en la formulación y análisis de los modelos de equilibrio multimodales (Florian [78], Abdulaal y LeBlanc [4], Dafermos [60], Florian y Spiess [87], Ferrari [75]). Estos modelos consideran varias alternativas de viajar de un origen a un destino mediante un modo de transporte *puro*, por ejemplo, usando el coche privado o mediante transporte público. Asumiendo una función de demanda (por ejemplo un modelo tipo logit) se produce la partición modal de viajes de acuerdo al coste de transporte en cada una de las alternativas consideradas.

Lam y Huang [137] proponen un modelo combinado de distribución y asignación para múltiples clases de usuarios en la red (multimodal) de tráfico en la que el tiempo de viaje en los arcos es idéntica para todos los usuarios. Toint y Wynter [227] abordan la formulación del problema asimétrico de asignación de tráfico multiusuario proponiendo una formulación general para evitar inconsistencias en la modelización del comportamiento.

Bifulco [27] desarrolla un modelo de asignación estocástica multiusuario para evaluar políticas de planificación de aparcamientos en áreas urbanas. Este autor emplea un modelo probit como modelo de demanda para representar las elecciones de aparcamiento y camino. Como modelo de oferta emplea una representación de la red de tráfico generalizada para incluir los aparcamientos e incluye los mecanismos de interacción entre oferta y demanda. La extensión de este modelo para considerar la elección del modo de transporte puede realizarse empleando un esquema de hipercaminos.

Muchos viajes urbanos emplean más de un modo de transporte (viajes *combinados*), el más usual es el denominado *park'n ride*. La promoción de los viajes combinados requiere de herramientas adecuadas que permitan tener en cuenta los atractivos de estos viajes. Estas herramientas deben recoger la congestión de la red de tráfico, las frecuencias de los servicios, precios del transporte público y de los aparcamientos, y todas sus interrelaciones. Hay dos modelos en la literatura que consideran explícitamente los viajes combinados. Florian y Los [83] desarrollaron un modelo para la distribución de usuarios de tipo *park'n ride* que determina la matriz origen-destino de la primera componente del viaje combinado, es decir, de su origen al aparcamiento. El interés de este problema se deriva de la necesidad de predecir los cambios en los flujos de tráfico en función de las políticas de aparcamientos

relativas a capacidades de aparcamiento, creación de nuevos aparcamientos y cambio en los precios. El inconveniente, es que la matriz O-D de viajes combinados es un dato del problema, considerándose además una matriz fija, lo que no permite tener en cuenta la naturaleza elástica del problema en función de las facilidades de transferencia entre redes.

La elección del nodo de transferencia (intercambiador) en los anteriores modelos es una consecuencia de la fase de asignación en las que se calculan los caminos empleados en la red multimodal y se asigna la correspondiente matriz de viajes O-D. Es decir, la elección del intercambiador está implícita en la elección del camino.

En muchas grandes ciudades, la política de planificación para promover el uso de transporte público en detrimento del vehículo privado se basa en el diseño de sistemas de transporte público de alta calidad y altamente interconectados. Este objetivo es alcanzado a través de la introducción de los intercambiadores multimodales urbanos donde se interconectan varias redes de transporte público. Unas redes pueden ser consideradas como principales (por ejemplo la red de cercanías y metro) cuyo objetivo es ofrecer viajes de gran distancia y las otras redes pueden ser consideradas secundarias (autobús, tranvías, etc.) cuyo objetivo es alimentar las redes principales. Además, estos intercambiadores ofrecen facilidades para otras formas de transporte como coches privados, bicicletas, taxi, andando, etc.

Un adecuado marco de modelización del diseño de estos intercambiadores debe tener en cuenta explícitamente la elección del nodo de transferencia por los usuarios. Para ilustrar su importancia basta considerar un viaje combinado cercanías-metro mediante dos nodos de transferencia distintos. El tiempo de viaje en ambos caminos puede no diferir significativamente. Si los modelos asignan la demanda al camino mínimo, ningún usuario emplearía el intercambiador asociado al mayor coste de transporte. En la realidad, si esta diferencia de coste no es significativa, los usuarios elegirán el intercambiador atendiendo a su atracción relativa basada fundamentalmente en factores no incluidos en el coste generalizado del viaje, como son la seguridad, confort, etc. y por tanto los usuarios emplearán ambos intercambiadores.

Esta misma cuestión puede plantearse en la elección de la parada donde comenzar el viaje (puro) en transporte público. Como el acceso a las paradas candidatas se realiza andando existe (posiblemente) gran diferencia entre los costes de viajes de las alternativas, haciendo determinante el coste de viaje.

El problema de diseño de intercambiadores requiere de la modelización de la elección que realizan los usuarios del nodo de transferencia entre redes de transporte público, así como un modelo de asignación de pasajeros en redes en equilibrio, como los desarrollados en Nguyen y Pallotino [181], Spiess y Florian [220], De Cea y Fernández [44], Wu y otros [240]. Para abordar el problema de la congestión en redes de transporte público se requiere de costes asimétricos y por tanto, de una formulación variacional del problema.

Modelos de asignación donde explícitamente se identifique la elección de los usuarios del nodo de transferencia no han recibido mucha atención en la literatura. Fernández y otros [73] presentaron varios modelos con modos combinados. Su modelo P3 tiene en cuenta explícitamente la elección del nodo de transferencia para los viajes combinados mediante un modelo de demanda logit anidado. La formulación empleada asume costes simétricos. Esta limitación reduce su aplicabilidad.

El modelo desarrollado en este capítulo puede ser considerado una extensión del desarrollado en Fernández y otros [73] al caso de costes asimétricos, imprescindibles para recoger un modelo genérico de asignación de usuarios en transporte público, que asume el primer principio de Wardrop (DUE) en cada red modal para modelar la elección de la ruta. Ejemplos de este tipo de asignación en transporte público son Nguyen y Pallotino [181], Spiess y Florian [220], De Cea y Fernández [44], Wu y otros [240], etc. Además posee una gran flexibilidad ya que su formulación permite emplear un determinado modelo de asignación de transporte público y una determinada representación de la red de tráfico.

El modelo propuesto emplea una distribución logit anidada para describir la elección del modo de transporte y nodo de transferencia. Si se deseasen considerar modelos más generales de demanda se podría recurrir al esquema de punto fijo de Cantarella [37].

Se han desarrollado dos algoritmos para la formulación variacional del TAP-M basada en el esquema

de descomposición simplicial de los trabajos de Lawphongpanich y Hearn [144], Larsson y otros [141], Patriksson [198]. El primer algoritmo es una descomposición simplicial desagregada (DSD, ver Larsson y Patriksson [140]) para el TAP-M, y el segundo algoritmo es obtenido reemplazando el subproblema lineal por el subproblema de linealización parcial del método de Evans [72]. Este algoritmo, en un contexto de optimización (no de desigualdades variacionales), ha sido empleado en Damberg y otros [64], Ludgren y Patriksson [151], García y Marín [94].

El modelo de asignación de tráfico puede ser formulado en el espacio de flujo en los arcos. Esta situación no siempre es posible en asignación de transporte público. Si el modelo de asignación de pasajeros (TEAP) empleado tuviese esta propiedad entonce el TAP-M se podría formular en el espacio de flujo en los arcos. Hemos discutido la especialización de los dos anteriores algoritmos para este caso, demostrando que éste es equivalente a la aplicación del RSD al TAP, donde las variables de demanda son tratadas como si fueran nuevos flujos.

1.2 Modelos de equilibrio con modos combinados

La principal cuestión a la hora de modelar los viajes en modos combinados es decidir que elecciones son representadas por el modelo de oferta (red de transporte) y cuáles por el modelo de demanda. En nuestro modelo la elección de la ruta está representada en el modelo de red multimodal y la elección de modo de transporte e intercambiador en el modelo de demanda. En las próximas secciones formulamos estos dos elementos del modelo TAP-M.

1.2.1 Modelización de la demanda

La promoción de los viajes combinados requiere de modelos que ayuden a la toma de decisiones. Estos modelos deben tener en cuenta la competencia entre modos de transporte (Ben-Akiva y Bowman [15]). Nuestro modelo considera las siguientes alternativas de viaje:

(a) Coche privado. El número de usuarios que emplea esta alternativa de transporte para la demanda O-D ω es denotado por g_ω^a.

(b) Transporte público con acceso andando o en bicicleta. El número de usuarios que emplean esta alternativa para el par O-D ω es denotado por g_ω^b.

(c) Transporte público con acceso en coche privado (*park'n ride*). El número de usuarios en esta alternativa empleando el nodo de transferencia $t \in T$ los denotaremos por $g_{\omega,t}^c$.

(d) Otros (andando, en motocicleta o en bicicleta). Esta alternativa está formada por el resto de formas de transporte que no han sido son consideradas anteriormente. El número de usuarios para esta alternativa es denotado por g_ω^d para el par O-D ω.

El modo de transporte (c) es un viaje en modo combinado sobre el sistema de transporte. Este modo incluye las alternativas automóvil-tren regional, automóvil-metro, etc. El modo de transporte (b) incluye viajes combinados como autobús-metro además de los modos puros: metro, autobús, tren regional, etc. Por claridad en la exposición supondremos que solamente los usuarios de *park'n ride* eligen nodo de transferencia explícitamente y para el resto de viajes combinados esta elección está determinada por la elección de la ruta. La alternativa (d) recoge todos los modos de transporte que no están afectados por la congestión de la red de transporte.

Hemos considerado un modelo de demanda logit anidado para desagregar el número total de viajes por modos de transporte e intercambiadores. La figura 1.8 muestra la estructura jerárquica en las elecciones efectuadas por los usuarios. Primeramente se elige el modo de transporte y posteriormente los usuarios de modos combinados eligen el nodo de transferencia.

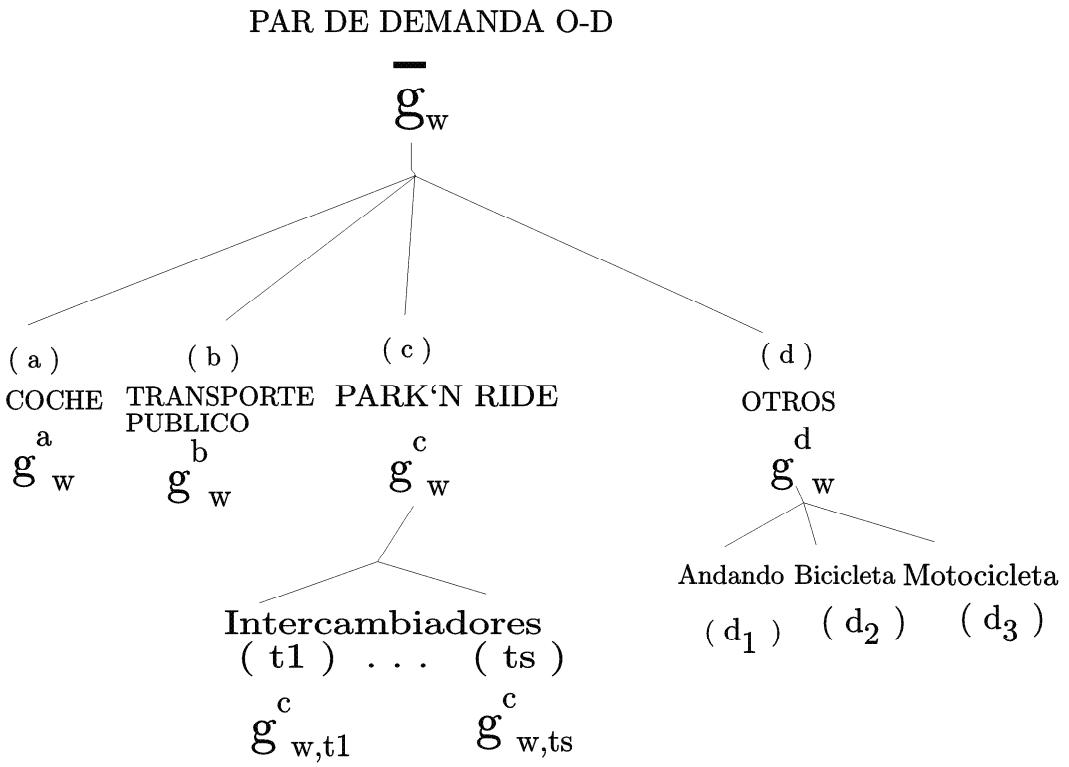


Figura 1.1: Modelo de demanda del modelo TAP-M

Las proporciones de viajeros están dadas para cada par O-D ω, y cada modo $k \in \{a, b, c, d\}$, mediante la fórmula

$$G^k_\omega(\mathbf{U}^*_\omega) = \frac{\exp\left\{-(\alpha^k + \beta_1 U^{k*}_\omega)\right\}}{\sum_{k' \in \{a,b,c,d\}} \exp\left\{-(\alpha^{k'} + \beta_1 U^{k'*}_\omega)\right\}}, \tag{1.1}$$

donde U^{k*}_ω es el coste generalizado de viajar en el par O-D ω mediante el modo k, que corresponde al uso óptimo de la red, $\{\mathbf{U}^*_\omega\}$ es el vector de costes generalizados para todos los modos considerados y α^k, β_1 son parámetros de la función logit.

Los modelos logit anidados son ampliamente empleados en al literatura (ver Hunt y Tepley [131], Fernández y otros [73]) para modelar la elección de aparcamientos. El modelo explícitamente incluye la elección del nodo de transferencia t (aparcamiento) en el contexto de viajes *park'n ride*, (c), y para la demanda ω mediante la función logit

$$G^c_{\omega,t}(U^{c*}_\omega) = \frac{\exp\left\{-(\alpha^c_t + \beta_2 U^{c*}_{\omega,t})\right\}}{\sum_{t' \in T_\omega} \exp\left\{-(\alpha^c_{t'} + \beta_2 U^{c*}_{\omega,t'})\right\}}, \tag{1.2}$$

donde T_ω es el conjunto de nodos de transferencia para el par ω y $U^{c*}_{\omega,t}$ es el coste de viaje en modo combinado para el par ω a través del nodo de transferencia t. Denotamos por $\mathbf{U}^{c*}_\omega$ el vector de estos costes.

El coste de la alternativa (c) para el par ω, U^{c*}_ω, es calculado como el "log-suma" de los costes a través de los nodos de transferencia

$$U^{c*}_\omega = \frac{-1}{\beta_2} \log\left(\sum_{t \in T_\omega} \exp\left\{-(\alpha^c_t + \beta_2 U^{c*}_{\omega,t})\right\}\right). \tag{1.3}$$

Los parámetros α_t^c con $t \in T$ representan el atractivo del nodo de transferencia t, debido a factores no incluidos en los costes generalizados $U_{\omega,t}^{c*}$ tales como: seguridad, confort, etc. y β_2 pondera la importancia de los costes generalizados en el proceso de decisión.

El modo (d) considera un conjunto de alternativas que no están afectadas por la congestión, tales como el modo andando (d_1), en bicicleta (d_2), en motocicleta (d_3), etc. El coste generalizado para el modo (d) será calculado como el "log-suma" de las utilidades de cada subalternativa, es decir

$$U_\omega^{d*} = \frac{-1}{\beta_3} \log \left(\sum_{k \in \{d_1, d_2, d_3\}} \exp\left\{ -(\alpha^k + \beta_3 U_\omega^{k*}) \right\} \right),$$

donde $\beta_3, \alpha^{d_1}, \alpha^{d_2}, \alpha^{d_3}$ son parámetros a estimar. Este coste es constante e independiente de los niveles de servicio en la red.

Las relaciones (1.1) y (1.2) desagregan la demanda. El número de usuarios que viajan en el modo $k \in \{a, b, c, d\}$ en el par O-D ω es calculado por

$$g_\omega^k = G_\omega^k(\mathbf{U}_\omega^*)\bar{g}_\omega, \tag{1.4}$$

y el número de viajes para el par ω empleando la alternativa *park'n ride* a través del intercambiador t, es calculado mediante la fórmula

$$g_{\omega,t}^c = G_{\omega,t}^c(\mathbf{U}_\omega^{\mathbf{c}*}) G_\omega^c(\mathbf{U}_\omega^*)\bar{g}_\omega. \tag{1.5}$$

1.2.2 Modelización de la red de transporte

En este modelo consideramos una red de transporte multimodal $\mathcal{G}$ que está formada por una red de tráfico $\mathcal{G}^a = (N^a, A)$, una red de transporte público $\mathcal{G}^b = (N^b, B)$ y un conjunto T de nodos de transferencia entre ambas redes. Para simplificar el modelo se asume que los arcos empleados en la representación de los intercambiadores son incluidos en el conjunto de arcos $A \cup B$, donde los arcos pedestres son incluidos en B, mientras que los asociados con los aparcamientos son incluidos en A.

El modelo de demanda considera un conjunto de pares $\omega = (i, j)$, donde i es el nodo origen y j el nodo destino. Denotamos mediante W el conjunto de estos pares. Los viajes combinados inducen un conjunto de demandas adicionales en cada red modal. En la red de tráfico aparecen demandas del tipo (i, t) que unen los nodos orígenes con el aparcamiento de los intercambiadores. En la red de transporte público se añaden a los pares definidos por la matriz de viajes O-D la demanda de viajes entre el intercambiador y el final del trayecto, esto es, aparecen nuevos pares de la forma (t, j). Definimos

$$W^a = W \cup \{(i, t) \mid t \in T_\omega, \text{ y } \omega = (i, j) \in W\},$$
$$W^b = W \cup \{(t, j) \mid t \in T_\omega, \text{ y } \omega = (i, j) \in W\},$$

donde T_ω es el conjunto de intercambiadores empleados por el par ω.

En la red de tráfico el conjunto factible de flujos en los arcos, Ω^a, está definido

$$\Omega^a = \left\{ \mathbf{h} \in \Re^{\bar{a}} \mid \sum_{p \in P_\omega^a} h_p = g_\omega^a, \quad \forall \omega \in W^a; \ \mathbf{h} \geq 0 \right\},$$

donde P_ω^a es el conjunto de caminos que conectan el par ω sobre la red $\mathcal{G}^a$, h_p es el flujo a través del camino p; y $\bar{a}$ es el cardinal del conjunto de caminos $P^a = \cup_{\omega \in W^a} P_\omega^a$. Suponemos que la elección de la ruta en la red $\mathcal{G}^a$ satisface el primer principio de Wardrop [233]

$$C_p^* - U_\omega^{a*} \begin{cases} = 0, & \text{si } h_p^* > 0; \\ \geq 0, & \text{si } h_p^* = 0. \end{cases} \quad \forall p \in P_\omega^a, \quad \forall \omega \in W^a, \tag{1.6}$$

donde U_ω^{a*} es el mínimo coste de transporte de i a j y C_p^* es el coste en equilibrio del camino p. Se puede demostrar que las anteriores condiciones de equilibrio (1.6) pueden formularse alternativamente mediante el siguiente problema de desigualdades variacionales: encontrar un vector $\mathbf{h}^* \in \Omega^a$ satisfaciendo

$$\mathbf{C}^a(\mathbf{h}^*)^T(\mathbf{h}-\mathbf{h}^*) \geq 0, \quad \forall \mathbf{h} \in \Omega^a, \qquad [\text{VIP}(\mathbf{C}^a, \Omega^a)]$$

donde $\mathbf{h}^*$ es el flujo de los caminos en el equilibrio y $\mathbf{C}^a(\mathbf{h})$ es la función vectorial de coste en los caminos cuyas componentes son $(C_p(\mathbf{h}))_{p\in P^a}$. La primera formulación variacional del problema de asignación de tráfico para el caso asimétrico fue realizada por Smith [213] y Dafermos [59].

Este problema también puede ser formulado en el espacio de los flujos en los arcos

$$\Omega_{\mathbf{f}}^a = \{\mathbf{f} \mid \mathbf{f} = \delta^{a\mathbf{f}}\mathbf{h}, \text{ con } \mathbf{h} \in \Omega^a\}$$

donde $\delta^{a\mathbf{f}}$ es la matriz de incidencia arco-camino. El valor del elemento $\delta_{lp}^{a\mathbf{f}}$ de la matriz $\delta^{a\mathbf{f}}$ es 1 si el camino $p \in P^a$ contiene el arco l y 0 en caso contrario. VIP$(\mathbf{C}^a, \Omega^a)$ es equivalente a encontrar un $\mathbf{f}^* \in \Omega_{\mathbf{f}}^a$

$$\mathbf{c}^a(\mathbf{f}^*)^T(\mathbf{f}-\mathbf{f}^*) \geq 0, \quad \forall \mathbf{f} \in \Omega_{\mathbf{f}}^a, \qquad [\text{VIP}(\mathbf{c}, \Omega_{\mathbf{f}}^a)]$$

donde $\mathbf{f}^*$ es el flujo en los arcos en equilibrio, $\mathbf{c}^a(\mathbf{f})$ es la función vectorial de coste en los arcos cuyas componentes son $(c_l(\mathbf{f}))_{l\in A}$, siendo $c_l(\mathbf{f})$ el tiempo de viaje en el arco $l \in A$ para el vector de flujo en los arcos $\mathbf{f}$.

La relación entre el coste en los caminos y en los arcos viene dada por la expresión

$$C_p(\mathbf{h}) = \sum_{l\in A} c_l(\mathbf{f})\delta_{lp}^a, \quad p \in P_\omega^a, \ \ \omega \in W^a. \tag{1.7}$$

La red de transporte público consta de un conjunto de líneas de metro y cercanías. Esta red es representada mediante un grafo que contiene cuatro tipos de arcos: arcos pedestres, arcos de espera, arcos asociados a desplazamientos en vehículos y arcos de transbordo/acceso a líneas. Es posible incluir el problema de líneas comunes de autobús, metro o trenes regionales mediante el concepto de hipercamino, que es más general que el concepto de camino. Nguyen y Pallotino [181].

En las cuatro décadas anteriores se han formulado varios modelos para la asignación de pasajeros en redes de transporte público congestionadas. Estos modelos se basan en diferentes hipótesis para describir el comportamiento de los usuarios (Bouzaïene-Ayari y otros [28]). En este trabajo no se asume ningún modelo determinado para el TEAP, debido a que el interés del modelo está en describir como los usuarios pueden emplear la red de tráfico y transporte público para realizar su viaje.

Denotamos el espacio de flujo en los hipercaminos por Ω^b. Este conjunto viene definido por

$$\Omega^b = \left\{ \mathbf{h} \in \Re^{\bar{b}} \;\middle|\; \sum_{p\in P_\omega^b} h_p = g_\omega^b, \quad \forall \omega \in W^b; \ \mathbf{h} \geq 0 \right\},$$

donde h_p es el flujo en el hipercamino p; g_ω^b es la demanda de viajes del origen $i \in N^b$ al destino $j \in N^b$, $\omega = (i,j)$, P_ω^b es el conjunto de hipercaminos del origen i al destino j en la red $\mathcal{G}^b$, y $\bar{b}$ es la cardinalidad del conjunto $P^b = \cup_{\omega\in W^b} P_\omega^b$.

Supondremos que los usuarios eligen la ruta dentro de $\mathcal{G}^b$ de acuerdo al primer principio de Wardrop [233]

$$C_p^* - U_\omega^{b*} \begin{cases} = 0, & \text{si } \ h_p^* > 0; \\ \geq 0, & \text{si } \ h_p^* = 0. \end{cases} \qquad p \in P_\omega^b, \ \ \omega \in W^b, \tag{1.8}$$

donde U_ω^{b*} es el coste mínimo de viaje entre i y j y C_p^* el coste del hipercamino p en el equilibrio. Se puede demostrar que las condiciones de equilibrio (1.8) pueden fomularse alternativamente mediante

el siguiente problema de desigualdades variacionales (Bouzaïene-Ayari y otros [28]): encontrar un $\mathbf{h}^* \in \Omega^b$ cumpliendo

$$\mathbf{C}^b(\mathbf{h}^*)^T(\mathbf{h}-\mathbf{h}^*) \geq 0, \quad \forall \mathbf{h} \in \Omega^b, \tag{VIP($\mathbf{C}^b, \Omega^b$)}$$

donde $\mathbf{h}^*$ es el flujo de los hipercaminos en equilibrio y $\mathbf{C}^b(\mathbf{h})$ es el coste en los hipercaminos, cuyas componentes son $(C_p(\mathbf{h}))_{p\in P^b}$.

El coste (esperado) de los hipercaminos es calculado en función de los costes de viajes y de los tiempos de espera en las paradas. Este modelo asume que el coste se puede calcular para cualquier hipercamino.

La red multimodal $\mathcal{G}$ integra las redes de transporte $\mathcal{G}^a$ y $\mathcal{G}^b$, y un usuario puede viajar en $\mathcal{G}$ empleando un camino $p \in P^a$, un hipercamino $p \in P^b$ o un hipercamino combinado $p = (p_a, p_b)$ donde $p_a \in P^a$ y $p_b \in P^b$. Denotamos por P^c el conjunto de hipercaminos combinados. Este cojunto cumple $P^c \subset P^a \times P^b$ y $P = P^a \cup P^b \cup P^c$ que es el conjunto de todos los caminos/hipercaminos en la red multimodal $\mathcal{G}$. Denotamos por P_ω el subconjunto de hipercaminos de P que pueden ser empleados para satisfacer el par O-D ω.

Hay dos asuntos fundamentales en la modelización de hipercaminos combinados:

◇ Compatibilidad de flujos. La primera dificultad aparece del hecho de que los flujos en la red de tráfico son calculados en unidades vehiculares, mientras que en la red de transporte público el flujo se calcula en número de usuarios. Hemos introducido la tasa de ocupación vehicular para transformar número de vehículos en número de usuarios. Este parámetro es denotado por γ_ω con $\omega \in W$. El flujo h_p para cualquier $p \in P$ y los flujos en los arcos de la red $\mathcal{G}^b$ son evaluados en número de usuarios y el flujo en los arcos de la red de tráfico son valorados en número de vehículos. Las restricciones para transformar el flujo en los caminos/hipercaminos en flujo en los arcos son

$$f_l = \sum_{\omega\in W} \frac{1}{\gamma_\omega}\left(\sum_{p\in P_\omega} \delta_{lp} h_p\right), \quad l \in A, \tag{1.9}$$

$$f_l = \sum_{\omega\in W} \left(\sum_{p\in P_\omega} \delta_{lp} h_p\right), \quad l \in B. \tag{1.10}$$

Estas restricciones pueden ser expresadas matricialmente por $\mathbf{f} = \delta^{\mathbf{f}}\mathbf{h}$, donde $\delta^{\mathbf{f}}$ es la matriz de incidencia arco/ruta en la red multimodal $\mathcal{G}$ cuyos elementos δ_{lp} están definidos por

$$\delta_{lp} = \begin{cases} \frac{1}{\gamma_\omega}, & \text{si el arco } l \in A \text{ y } l \text{ es usado por el camino } p \in P_\omega; \\ 1, & \text{si el arco } l \in B \text{ y } l \text{ es usado por el camino } p \in P_\omega; \\ 0, & \text{en otro caso.} \end{cases} \qquad l \in A \cup B,\ p \in P_\omega,\ \omega \in W.$$

◇ Compatibilidad de costes. Una importante cuestión es saber cómo los costes generalizados se pueden incluir en el modelo de demanda para asegurar compatibilidad entre las medidas obtenidas de las dos redes diferentes. Los costes de viaje en la red de transporte público y en la red de tráfico tienen diferente naturaleza. Por ejemplo, se puede considerar que los tiempos andando son más importantes para los usuarios que los tiempos en los vehículos. Hemos incluido dos parámetros, θ_a y θ_b, para homogeneizar los costes de ambas redes.

El coste en un camino de la red de tráfico está dado en función del número de vehículos. La introducción de γ_ω, (tasa de ocupación vehicular para el par O-D ω) nos permite transformar el coste $C_p,\ p \in P^a_\omega$, que está expresado en unidades vehiculares, a coste por número de usuarios. Esto nos permite comparar con los costes $C_p,\ p \in P^b_\omega$ en la red de transporte público.

Definimos

$$\bar{C}_p(\mathbf{h}) = \begin{cases} \frac{\theta_a}{\gamma_\omega} C_p(\mathbf{h}), & p \in P^a_\omega,\ \omega \in W; \\ \theta_b C_p(\mathbf{h}), & p \in P^b_\omega,\ \omega \in W. \end{cases}, \tag{1.11}$$

donde C_p representa el coste del camino/hipercamino en cada red $\mathcal{G}^a$ y $\mathcal{G}^b$.

El coste de un camino combinado $p = (p_a, p_b) \in P^c_\omega$ se calcula como la suma del coste en cada componente

$$\bar{C}_p(\mathbf{h}) = \frac{\theta_a}{\gamma_\omega} C_{p_a}(\mathbf{h}) + \theta_b C_{p_b}(\mathbf{h}). \tag{1.12}$$

Notar que el mismo camino en la red de tráfico uniendo un origen i y un nodo de transferencia t, conduce a dos costes diferentes, dependiendo del hipercamino combinado en el que se integre. Esto es debido a las posibles diferentes tasas de ocupación según los pares O–D ω. Para resaltar este hecho introducimos la notación

$$\bar{C}_{p_a,\omega}(\mathbf{h}) = \frac{\theta_a}{\gamma_\omega} C_{p_a}(\mathbf{h}), \quad p_a \in P^a_\omega,\ \omega \in W^a,$$

y el coste del hipercamino combinado es expresado por

$$\bar{C}_p(\mathbf{h}) = \bar{C}_{p_a,\omega}(\mathbf{h}) + \bar{C}_{p_b}(\mathbf{h}), \quad p = (p_a, p_b) \in P^c_\omega,\ \omega \in W.$$

Para representar las condiciones de equilibrio es necesario introducir los siguiente conjuntos de hipercaminos

$P^c_{\omega,t}$ es el subconjunto de hipercaminos de P^c_ω que viajan a través del nodo de transferencia t.

P^a_{it} es el conjunto de caminos que conecta el origen i con el nodo de transferencia t sobre la red $\mathcal{G}^a$.

P^b_{tj} es el conjunto de hipercaminos conectando el nodo de transferencia t con el destino j sobre la red $\mathcal{G}^b$.

La alternativa (d) no está afectada por la congestión y por esta razón la red de transporte relativa a esta alternativa no es tenida en cuenta. Lo relevante es que su coste de transporte es constante y éste lo denotamos por U^{d*}_ω con $\omega \in W$. Consideramos que el conjunto de caminos que satisfacen la demanda ω mediante el modo (d) es unitario. Este conjunto es $P^d_\omega = \{p^d_\omega\}$ y $\bar{C}_{p^d_\omega}(h_{p^d_\omega}) = U^{d*}_\omega$.

1.2.3 Condiciones de equilibrio

Las condiciones de equilibrio constan de los siguientes tres conjuntos de condiciones.

C1: Elección de ruta. Un modelo de asignación en redes en equilibrio tiene como objetivo proveer una descripción macroscópica de los volúmenes de tráfico-usuarios resultantes de la elección de ruta efectuada en la red multimodal. Esta elección en cada red modal es efectuada mediante el primer principio de Wardrop, y puede ser formulado por

$$\begin{aligned}
(\theta_a/\gamma_\omega)C^*_p - U^{a*}_\omega \quad & \begin{cases} = 0, & \text{si } h^*_p > 0, \\ \geq 0, & \text{si } h^*_p = 0, \end{cases} && p \in P^a_\omega,\ \omega \in W^a, \\
(\theta_a/\gamma_\omega)C^*_p - U^{a*}_{it,\omega} \quad & \begin{cases} = 0, & \text{si } h^*_p > 0, \\ \geq 0, & \text{si } h^*_p = 0, \end{cases} && p \in P^a_{it},\ (i,t) \in W^a, \\
\theta_b C^*_p - U^{b*}_\omega \quad & \begin{cases} = 0, & \text{si } h^*_p > 0, \\ \geq 0, & \text{si } h^*_p = 0, \end{cases} && p \in P^b_\omega,\ \omega \in W^b, \\
\theta_b C^*_p - U^{b*}_{tj} \quad & \begin{cases} = 0, & \text{si } h^*_p > 0, \\ \geq 0, & \text{si } h^*_p = 0, \end{cases} && p \in P^b_{tj},\ (t,j) \in W^b.
\end{aligned} \tag{1.13}$$

Para los viajes combinados de tipo *park'n ride* el coste generalizado es

$$U^{c*}_{\omega,t} = U^{a*}_{it,\omega} + U^{b*}_{tj}, \quad t \in T_\omega,\ \omega \in W. \tag{1.14}$$

C2: Elección del modo de transporte. La proporción de usuarios en cada modo $k \in \{a, b, c, d\}$ y para cada par $\omega \in W$, es dada mediante la función de demanda G_ω^k definida en (1.1), donde la utilidad para la alternativa (c), U_ω^{c*}, es calculado por (1.3). Cuando estas proporciones son alcanzadas ningún usuario tiene unilateralmente el incentivo de cambiar de modo de transporte.

C3: Elección del nodo de transferencia. La proporción de usuarios en modo combinado *park'n ride* eligen cada nodo de transferencia $t \in T_\omega$ y para cada par $\omega \in W$ de acuerdo a la función de demanda $G_{\omega,t}^c$ definida en (1.2). Cuando estas proporciones son alcanzadas ningún usuario del modo combinado tiene el incentivo de cambiar unilateralmente de nodo de transferencia.

Condición unificada de equilibrio

Las condiciones de equilibrio se han introducido mediante tres conjuntos diferentes de condiciones **C1**, **C2** y **C3**. El primer conjunto describe la elección de la ruta/estrategia en cada red de transporte, el segundo la elección del modo de transporte y el tercer conjunto el nodo de transferencia para los viajes combinados. En esta sección se da una formulación unificada de las tres condiciones. Si un usuario elige un hipercamino en la red multimodal $\mathcal{G}$, implícitamente ha elegido un modo de transporte, ruta/estrategia y un nodo de transferencia para la alternativa en modo combinado. Consideremos que cada hipercamino en la red multimodal tiene un *coste de equilibrio extendido* que modeliza estas elecciones implícitas. El comportamiento de los usuarios es por tanto formulado como una versión del primer principio de Wardrop, donde un usuario elige el hipercamino para realizar el viaje con menor coste extendido. Todos los hipercaminos usados en el equilibrio poseen el mismo coste extendido y éste es igual o menor al coste extendido del resto de hipercaminos.

Denotamos por Ω el conjunto de flujo factible en los hipercaminos en la red multimodal $\mathcal{G}$ y viene definido por

$$\Omega = \left\{ \mathbf{h} \in \Re^M \left| \sum_{k\in\{a,b,c,d\}} \sum_{p\in P_\omega^k} h_p = \bar{g}_\omega,\ \forall \omega \in W;\ \mathbf{h} \geq 0 \right. \right\}, \tag{1.15}$$

donde M es el cardinal del conjunto P.

A continuación se desarrolla las relaciones entre el flujo en los hipercaminos y las variables de demanda. Éstas definen un conjunto factible de flujo en los hipercaminos y desagregación modal y por nodos de transferencia de la demanda total. La primera restricción es la partición modal de la demanda total, que impone para cada par O-D ω que la demanda total debe ser igual a la suma de la demanda satisfecha en cada una de las alternativas, esto es

$$\bar{g}_\omega = \sum_{k\in\{a,b,c,d\}} g_\omega^k, \tag{1.16}$$

donde la demanda en modo combinado (c) en el par O-D ω viene dada por

$$g_\omega^c = \sum_{t\in T_\omega} g_{\omega,t}^c. \tag{1.17}$$

La siguiente restricción relaciona la demanda en cada alternativa de transporte con los flujos en los hipercaminos.

$$g_\omega^k = \sum_{p\in P_\omega^k} h_p, \quad k \in \{a, b, c, d\},\ \omega \in W, \tag{1.18}$$

$$g_{\omega,t}^c = \sum_{p\in P_{\omega,t}^c} h_p, \quad t \in T_\omega,\ \omega \in W. \tag{1.19}$$

Las relaciones (1.18) y (1.19) pueden ser expresadas en forma matricial por

$$\mathbf{g} = \delta^{\mathbf{g}}\mathbf{h}, \tag{1.20}$$

donde $\delta^{\mathbf{g}}$ es la matriz de incidencia demanda/hipercamino.

Para formular las condiciones de equilibrio consideramos la inversa de la función de demanda que es denotada por $\lambda_p(\mathbf{g})$. La expresión analítica de estas funciones está mostrada en el teorema 1.2.1. Estas funciones producirán los costes asociados a la elección de modo y nodo de transferencia. Como los costes de elección de rutas están expresados en función de los flujos en los hipercaminos, también representaremos los costes de modo e intercambiador en función de la variable $\mathbf{h}$. Para ello emplearemos la relación (1.20) y definimos

$$\Lambda(\mathbf{h}) = \lambda(\delta^{\mathbf{g}}\mathbf{h}),$$

donde $\lambda(\mathbf{g}) = (\lambda_p(\mathbf{g}))_{p\in P}$. El denominado "coste extendido" es la suma del coste de transporte más el coste $\Lambda_p(\mathbf{h})$.

El teorema 1.2.1 proporciona explícitamente las reglas para calcular los costes extendidos y caracteriza los hipercaminos empleados en el equilibrio para cada par de demanda ω. Los coeficientes λ^*_ω juegan el papel de los costes extendidos en el equilibrio para el par ω.

TEOREMA 1.2.1 (Condición unificada de equilibrio para el TAP-M.) *Un vector $\mathbf{h}^* \in \Omega$ es un flujo en equilibrio para el TAP-M si y sólo si existe un conjunto de valores λ^*_ω para todo par $\omega \in W$ cumpliendo*

$$\left[\bar{C}_p(\mathbf{h}^*) - \Lambda_p(\mathbf{h}^*)\right] - \lambda^*_\omega \begin{cases} = 0, & si\ h^*_p > 0, \\ \geq 0, & si\ h^*_p = 0, \end{cases} \qquad \forall p \in P_\omega,\ \forall \omega \in W. \tag{1.21}$$

donde $\Lambda_p(\mathbf{h}) = \lambda_p(\delta^{\mathbf{g}}\mathbf{h})$ y $\lambda_p(\mathbf{g})$ está definida por

$$\lambda_p(\mathbf{g}) = \begin{cases} -\frac{\ln g^k_\omega + \alpha^k}{\beta_1}, & si\ p \in P^k_\omega,\ k \in \{a,b,d\},\ \omega \in W, \\ -\frac{\ln g^c_\omega + \alpha^k}{\beta_1} - \frac{-\ln g^c_{\omega,t} + \ln g^c_{\omega,t} + \alpha^c_t}{\beta_2}, & si\ p \in P^c_{\omega,t},\ \omega \in W, \end{cases} \tag{1.22}$$

DEMOSTRACIÓN. Probaremos que si $\mathbf{h}^*$ cumple las condiciones (1.21) entonces $\mathbf{h}^*$ satisface las condiciones **C1**, **C2** y **C3**.

Primero veremos la condición **C1**. Consideraremos la partición $\{P^a_\omega, P^b_\omega, P^c_{\omega,t}, P^d_\omega\}$ del conjunto P_ω. Sean p_1 y p_2 dos hipercaminos con flujo positivo pertenecientes a la misma componente de la partición. Si $p_1, p_2 \in P^d_\omega$ entonces nada debe ser probado. Si $k \in \{a,b\}$, y usando (1.21), se satisface

$$\lambda^*_\omega = \bar{C}_{p_1}(\mathbf{h}^*) - \Lambda_{p_1}(\mathbf{h}^*) = \bar{C}_{p_2}(\mathbf{h}^*) - \Lambda_{p_2}(\mathbf{h}^*).$$

La relación (1.22) implica que $-\Lambda_p(\mathbf{h}^*)$ es constante sobre los hipercaminos de la misma componente P^k_ω donde $k \in \{a,b,c,d\}$. Denotamos este valor por ε^k_ω que depende solamente de la variable de demanda $\mathbf{g}$ y $\varepsilon^k_\omega = -\Lambda_{p_1}(\mathbf{h}^*) = -\Lambda_{p_2}(\mathbf{h}^*)$.

El coste extendido es la suma de un coste que depende del hipercamino p_i más el valor ε^k_ω, que sólo depende del modo de transporte empleado para el par O-D ω. Entonces obtenemos $\bar{C}_{p_1}(\mathbf{h}^*) = \bar{C}_{p_2}(\mathbf{h}^*)$. Esto implica que todos los hipercaminos usados para satisfacer el mismo par de demanda ω, pertenecientes a la misma componente, tienen el mismo coste. Denotamos por

$$U^{k*}_\omega = \bar{C}_p(\mathbf{h}^*),\ \text{si}\ p \in P^k_\omega,\ k \in \{a,b\}\ \text{y}\ h^*_p > 0. \tag{1.23}$$

Empleando la relación (1.21)

$$U^{k*}_\omega = \lambda^*_\omega - \varepsilon^k_\omega \leq \bar{C}_p + \varepsilon^k_\omega - \varepsilon^k_\omega = \bar{C}_p, \quad \forall p \in P^k_\omega,\ k \in \{a,b\}. \tag{1.24}$$

La relaciones (1.23) y (1.24) muestran que los hipercaminos de los modos puros (a) y (b) satisfacen el primer principio de Wardrop.

Ahora consideraremos el caso de que $p_1, p_2 \in P^c_{\omega,t}$. Supondremos que p_1 y p_2 son dos hipercaminos combinados con flujo positivo. Análogamente a la anterior discusión se prueba que

$$\lambda^*_\omega = \bar{C}_{p_1}(\mathbf{h}^*) + \varepsilon^c_{\omega,t} = \bar{C}_{p_2}(\mathbf{h}^*) + \varepsilon^c_{\omega,t},$$

donde $\varepsilon^c_{\omega,t}$ depende de g^c_ω y $g^c_{\omega,t}$. Este valor es el mismo para cualquier hipercamino de $P^c_{\omega,t}$. Esto implica que $\bar{C}_{p_1}(\mathbf{h}^*) = \bar{C}_{p_2}(\mathbf{h}^*)$.

Denotamos

$$U^{c*}_{\omega,t} = \bar{C}_p(\mathbf{h}^*), \text{ si } p \in P^c_{\omega,t} \text{ y } h^*_p > 0. \tag{1.25}$$

Se cumple por (1.21)

$$U^{c*}_{\omega,t} = \lambda^*_\omega - \varepsilon^c_{\omega,t} \leq \bar{C}_p + \varepsilon^c_{\omega,t} - \varepsilon^c_{\omega,t} = \bar{C}_p, \forall p \in P^c_{\omega,t}. \tag{1.26}$$

Consideremos un hipercamino $p' = (p'_a, p'_b) \in P^c_{\omega,t}$ con flujo positivo. El coste del hipercamino p' puede ser expresado

$$\bar{C}_{p'} = \bar{C}_{p'_a,\omega}(\mathbf{h}^*) + \bar{C}_{p'_b}(\mathbf{h}^*).$$

Las relaciones (1.25) y (1.26) muestran que p' es el hipercamino de coste mínimo de i a j a través del intercambiador t. Empleando el principio de optimalidad de Bellman, el camino p'_a y el hipercamino p'_b deben también ser óptimos. Esto significa que el mínimo coste de i a t, que nosotros denotamos por $U^{a*}_{it,\omega}$, es $\bar{C}_{p'_a,\omega}(\mathbf{h}^*)$ y de t a j, que denotamos por U^{b*}_{tj}, es $\bar{C}_{p'_b}(\mathbf{h}^*)$. Obtenemos

$$\begin{aligned} U^{a*}_{it,\omega} &= \bar{C}_{p'_a,\omega}(\mathbf{h}^*), \\ U^{b*}_{tj} &= \bar{C}_{p'_b}(\mathbf{h}^*). \end{aligned}$$

Por otro lado, sea $p = (p_a, p_b) \in P^c_{\omega,t}$. Empleando la optimalidad de $U^{a*}_{it,\omega}$ y U^{b*}_{tj}, obtenemos

$$\begin{aligned} U^{a*}_{it,\omega} &\leq \bar{C}_{p_a,\omega}(\mathbf{h}^*) = \theta_a/\gamma_\omega C_{p_a}(\mathbf{h}^*), \\ U^{b*}_{tj} &\leq \bar{C}_{p_b}(\mathbf{h}^*) = \theta_b C_{p_b}(\mathbf{h}^*). \end{aligned}$$

Esto completa la demostración de **C1**.

Ahora probaremos la condición **C3**. Consideraremos que $\mathbf{g} = \delta^{\mathbf{g}}\mathbf{h}^*$. Sea $p \in P^c_{\omega,t}$ tal que $h^*_p > 0$, entonces $\bar{C}_p = U^{c*}_{\omega,t}$ y empleando la expresión (1.21) y (1.22), obtenemos la relación

$$\lambda^*_\omega = \frac{\ln g^c_{\omega,t} + \alpha^c_t}{\beta_2} + \lambda^c_\omega + U^{c*}_{\omega,t}, \tag{1.27}$$

donde

$$\lambda^c_\omega = \frac{\ln g^c_\omega + \alpha^c}{\beta_1} - \frac{\ln g^c_\omega}{\beta_2}, \tag{1.28}$$

y despejando $g^c_{\omega,t}$ de (1.27) obtenemos

$$g^c_{\omega,t} = \exp\left(-\{\alpha^c_t + \beta_2 U^{c*}_{\omega,t}\}\right) \exp\left(\beta_2 \tilde{\lambda}^c_\omega\right), \tag{1.29}$$

donde $\tilde{\lambda}^c_\omega = \lambda^*_\omega - \lambda^c_\omega$.

Empleando (1.17) obtenemos

$$g^c_\omega = \sum_{t \in T_\omega} \exp\left(-\{\alpha^c_t + \beta_2 U^{c*}_{\omega,t}\}\right) \exp\left(\beta_2 \tilde{\lambda}^c_\omega\right), \tag{1.30}$$

y despejando $\tilde{\lambda}^c_\omega$ de (1.30) obtenemos

$$\tilde{\lambda}^c_\omega = \frac{1}{\beta_2} \ln g^c_\omega + U^{c*}_\omega, \tag{1.31}$$

donde U^{c*}_ω es el "log-suma" de los costes de transporte por cada intercambiador, ver (1.3).

Sustituyendo la expresión de $\tilde{\lambda}^c_\omega$ en (1.29) obtenemos

$$g^c_{\omega,t} = \frac{\exp - \left(\alpha^c_t + \beta_2 U^{c*}_{\omega,t}\right)}{\sum_{t' \in T_\omega} \exp - \left(\alpha^c_{t'} + \beta_2 U^{c*}_{\omega,t'}\right)} g^c_\omega.$$

La anterior expresión muestra que se cumple **C3**.

Ahora probaremos la condición **C2**. Sustituyendo el valor de λ_ω^c dado en (1.28) y el valor de $\tilde{\lambda}^c$ dado en (1.31) en la relación $\lambda_\omega^* = \lambda_\omega^c + \tilde{\lambda}_\omega^c$, obtenemos

$$\lambda_\omega^* = \lambda_\omega^c + \frac{1}{\beta_2}\ln g_\omega^c + U_\omega^{c*} = \frac{\ln g_\omega^c + \alpha^c}{\beta_1} - \frac{1}{\beta_2}\ln g_\omega^c + \frac{1}{\beta_2}\ln g_\omega^c + U_\omega^{c*} = \frac{\ln g_\omega^c + \alpha^c}{\beta_1} + U_\omega^{c*}. \quad (1.32)$$

Por otro lado, empleando la condición de equilibrio (1.21) obtenemos

$$\lambda_\omega^* = \frac{\ln g_\omega^k + \alpha^k}{\beta_1} + U_\omega^{k*}, \quad k \in \{a, b, d\}, \quad (1.33)$$

y despejando g_ω^k de (1.32) y sustituyendo en (1.33), obtenemos

$$g_\omega^k = \exp\left(-\{\alpha^k + \beta_1 U_\omega^{k*}\}\right)\exp\left(\beta_1 \lambda_\omega^*\right), \quad k \in \{a, b, c, d\}. \quad (1.34)$$

Sustituyendo en la relación (1.16) las expresiones que hemos obtenido en (1.34) y despejando el valor de λ_ω^*, llegamos a la expresión

$$\lambda_\omega^* = \frac{-1}{\beta_1}\log\left[\frac{\bar{g}_\omega}{\sum_{k\in\{a,b,c,d\}}\exp\left(-\{\alpha^k + \beta_1 U_\omega^{k*}\}\right)}\right]. \quad (1.35)$$

Substituyendo (1.35) en (1.34) queda demostrada que **C2** se satisface.

La otra implicación es que si $\mathbf{h}^*$ satisface **C1**, **C2** y **C3** entonces se cumple (1.21). Esta implicación se demuestra empleando argumentos similares, por lo que se ha omitido su inclusión. □

1.2.4 Formulación matemática de las condiciones de equilibrio

Sea $\bar{\mathbf{C}} - \Lambda : \Re^M \mapsto \Re^M$ una función vectorial cuyas componentes son $\bar{C}_p(\mathbf{h}) - \Lambda_p(\mathbf{h})$ para todo $p \in P_\omega$ y para todo $\omega \in W$. El siguiente teorema formula las condiciones de equilibrio para el TAP-M mediante un problema de desigualdades variacionales.

TEOREMA 1.2.2 (Formulación del TAP-M mediante desigualdades variacionales.) *Un vector $\mathbf{h}^* \in \Omega$ es un flujo (en los hipercaminos) en equilibrio para el TAP-M si y sólo si cumple la siguiente desigualdad variacional*

$$[\bar{\mathbf{C}}(\mathbf{h}^*) - \Lambda(\mathbf{h}^*)]^T(\mathbf{h} - \mathbf{h}^*) \geq 0, \quad \forall \mathbf{h} \in \Omega. \qquad [\text{TAP-MVIP}(\bar{\mathbf{C}} - \Lambda, \Omega)]$$

DEMOSTRACIÓN. La demostración de este teorema está basada en la presentada en el trabajo Ferrari [75]. Es fácil verificar que $\mathbf{h}^*$ es una solución del TAP-MVIP($\bar{\mathbf{C}} - \Lambda, \Omega$) si y sólo si es solución del siguiente problema de optimización

$$\begin{array}{rl} \text{minimizar} & Z = \phi(\mathbf{h}) \\ \text{sujeto a} & \mathbf{h} \in \Omega, \end{array} \quad (1.36)$$

donde $\phi(\mathbf{h}) = [\bar{\mathbf{C}}(\mathbf{h}^*) - \Lambda(\mathbf{h}^*)]^T(\mathbf{h} - \mathbf{h}^*)$.

Escribiendo las restricciones que definen el conjunto Ω explícitamente

$$G_\omega(\mathbf{h}) = \sum_{k\in\{a,b,c,,d\}}\sum_{p\in P_\omega^k} h_p - \bar{g}_\omega = 0, \ \forall \omega \in W, \quad (1.37)$$

$$s_p(\mathbf{h}) = -\mathbf{h}_p \leq \mathbf{0}, \quad \forall p \in P. \quad (1.38)$$

La función objetivo $\phi(\mathbf{h})$ y las restricciones (1.37) y (1.38) son lineales, entonces $\mathbf{h}^*$ es solución del problema (1.36) sii verifica la condiciones de optimalidad de KKT

$$\bar{\mathbf{C}}(\mathbf{h}^*) - \Lambda(\mathbf{h}^*) + \sum_{p\in P} u_p \nabla s_p(\mathbf{h}^*) + \sum_{\omega\in W} \rho_\omega \nabla G_\omega(\mathbf{h}^*) = \mathbf{0},$$

donde los multiplicadores u_p para todo $p \in P$, son no negativos y verifican la *condición de complementariedad* (CS), $u_p s_p(\mathbf{h}^*) = 0$, y donde los multiplicadores ρ_ω para todo $\omega \in W$ pueden tener cualquier signo.

Si $h_p^* > 0$ con $p \in P$, empleando la condición CS $u_p = 0$ y $\bar{C}_p(\mathbf{h}^*) - \Lambda_p(\mathbf{h}^*) = -\rho_\omega$; por otro lado, si $h_p^* = 0$, $p \in P$ obtenemos $u_p \geq 0$ y entonces $\bar{C}_p(\mathbf{h}^*) - \Lambda_p(\mathbf{h}^*) \geq -\rho_\omega$. Tomando $\lambda_\omega^* = -\rho_\omega$, y empleando el teorema 1.2.1 obtenemos que $\mathbf{h}^*$ es un vector de flujo en equilibrio si y sólo si es solución de TAP-MVIP($\bar{\mathbf{C}} - \Lambda, \Omega$). □

1.3 Algoritmos de generación de columnas/descomposición simplicial

En esta sección derivamos dos algoritmos de generación de columnas / descomposición simplicial (CG/SD) (ver Patriksson [198]) aplicados al problema de desigualdades variacionales TAP-MVIP($\bar{\mathbf{C}} - \Lambda, \Omega$).

Estos algoritmos resuelven iterativamente dos problemas de desigualdades variacionales: el llamado *problema de generación de columnas* (CGPVIP) y el llamado *problema maestro restringido* (RMPVIP). CGPVIP genera una nueva columna de la región factible mediante la aproximación de la función de costes y resolviendo esta aproximación en la región factible original, y el RMPVIP resuelve la función de coste original sobre un subconjunto de la región factible, definido mediante las columnas generadas anteriormente. El RMPVIP es un problema de desigualdades variacionales con restricciones simples cuya solución define el próximo punto donde formular el nuevo CGPVIP. Formalmente, el esquema iterativo anterior se define como sigue:

1. CGPVIP$_\varphi(\ell)$. Sea φ una función vectorial $\Re^M \times \Re^M \mapsto \Re^M$. En el punto $\mathbf{h}^{\ell-1}$ el CGPVIP se define como encontrar un $\bar{\mathbf{h}}^\ell \in \Omega$ cumpliendo

$$\varphi(\mathbf{h}^{\ell-1}, \bar{\mathbf{h}}^\ell)^T(\mathbf{h} - \bar{\mathbf{h}}^\ell) \geq 0, \quad \forall \mathbf{h} \in \Omega. \qquad [\text{CGPVIP}(\varphi(\mathbf{h}^{\ell-1}, \cdot), \Omega))]$$

   donde $\varphi(\mathbf{h}^{\ell-1}, \cdot)$ es una aproximación de la función de coste $(\bar{\mathbf{C}} - \Lambda)(\cdot)$ en el punto $\mathbf{h}^{\ell-1}$. En este capítulo tratamos con dos posibles elecciones de la función φ:

   (a) Decomposición simplicial (SD). La primera conduce a la clásica formulación del algoritmo SD (ver Lawphongpanich y Hearn [144], Pang y Yu [192], Marcotte y Guélat [161], Montero y Barceló [174]) y ésta es

$$\varphi_{SD}(\mathbf{h}^{\ell-1}, \mathbf{s}) = \bar{\mathbf{C}}(\mathbf{h}^{\ell-1}) - \Lambda(\mathbf{h}^{\ell-1}).$$

   (b) Algoritmo de Evans con búsqueda multidimensional (E). Este algoritmo usa un CGPVIP basado en el subproblema de Evans [72]. Este algoritmo se deriva de la elección

$$\varphi_E(\mathbf{h}^{\ell-1}, \mathbf{s}) = \bar{\mathbf{C}}(\mathbf{h}^{\ell-1}) - \Lambda(\mathbf{s}).$$

   Los CGPVIP$_\varphi$ con $\varphi \in \{SD, E\}$ pueden ser resueltos explícitamente (ver apéndice I y II). CGPVIP$_E$ requiere esencialmente la misma cantidad de trabajo para resolverlo que CGPVIP$_{SD}$, pero la eficiencia de todo el procedimiento es mucho mayor.

2. RMPVIP(ℓ). El problema original se resuelve sobre un subconjunto compacto y convexo Ω^ℓ de la región factible Ω. Este conjunto se define por el algoritmo CG/SD. El RMPVIP en la iteración ℓ se define del siguiente modo: encontrar un $\mathbf{h}^\ell \in \Omega^\ell$ cumpliendo

$$\bar{\mathbf{C}}(\mathbf{h}^\ell) - \Lambda(\mathbf{h}^\ell)]^T(\mathbf{h} - \mathbf{h}^\ell) \geq \epsilon, \quad \forall \mathbf{h} \in \Omega^\ell, \qquad [\text{RMPVIP}(\bar{\mathbf{C}} - \Lambda, \Omega^\ell)]$$

   donde ϵ es un parámetro dado. El punto $\mathbf{h}^\ell$ es denominado una solución $\epsilon-$óptima de RMPVIP(ℓ) y define el próximo CGPVIP.

Tabla 1.1: Algoritmos CG/SD para el TAP-MVIP

0. (*Inicialización*): Elegir un punto inicial $\mathbf{h}^0 \in \Omega$, dos parámetros de tolerancia $\epsilon_1, \epsilon_2 > 0$, tomar $\ell := 1$ y $\Omega^0 = \{\mathbf{h}^0\}$.
1. (*Problema de generación de columnas*): Resolver CGPVIP$_\varphi(\ell)$. Sea $\bar{\mathbf{h}}^\ell$ una solución.
2. (*Criterio de convergencia*): Si $\mathbf{h}^{\ell-1}$ resuelve CGPVIP$_\varphi(\ell)$ entonces finalizar, ($\mathbf{h}^{\ell-1}$ resuelve TAP-MVIP$(\bar{\mathbf{C}} - \Lambda, \Omega)$). En caso contrario continuar.
3. (*Criterio de terminación*): Sea la función de mérito GAP$_\varphi(\mathbf{h}^{\ell-1})$ definida por

$$\text{GAP}_\varphi(\mathbf{h}^{\ell-1}) = \min_{\mathbf{h}\in\Omega} \varphi(\mathbf{h}^{\ell-1}, \mathbf{h})^T(\mathbf{h}^{\ell-1} - \mathbf{h}) = \varphi(\mathbf{h}^{\ell-1}, \bar{\mathbf{h}}^\ell)^T(\mathbf{h}^{\ell-1} - \bar{\mathbf{h}}^\ell).$$

   Si GAP$_\varphi(\mathbf{h}^\ell) \leq \epsilon_1$ entonces parar.
4. (*Aumento del conjunto*): Sea Ω^ℓ un subconjunto convexo y compacto de Ω, cumpliendo

$$\bar{\mathbf{h}}^\ell \in \Omega^\ell \supseteq \Omega^{\ell-1}.$$

5. (*Problema maestro restringido*): Encontrar una solución ϵ_2- óptima $\mathbf{h}^\ell$ de RMPVIP(ℓ).
6. (*Actualización*): Sean $\ell := \ell + 1$ y volver al paso 1.

Estos algoritmos pertenecen a la clase CG/SD y pueden ser formulados como en la tabla 1.1.

Patriksson [198] garantiza la convergencia (teorema 9.16) del algoritmo previo, bajo la hipótesis de que la función de coste $\bar{\mathbf{C}}(\mathbf{h})$ es monótona, continuamente lipschitziana, de clase $\mathcal{C}^1$ en Ω (entonces la aplicación de coste $\bar{\mathbf{C}}(\mathbf{h}) - \Lambda(\mathbf{h})$ también cumple las anteriores propiedades) y bajo una solución exacta al problema RMPVIP ($\epsilon_2 = 0$).

1.3.1 Resolución del RMPVIP

Para que un algoritmo CG/SD sea operativo, dos cuestiones deben ser analizadas. La primera es cómo se define la región factible para el RMPVIP y la segunda el procedimiento para resolverlo. Ambos problemas están interrelacionados y deben ser abordados simultáneamente.

Comenzamos analizando la definición de Ω^ℓ. Una propiedad importante para desarrollar algoritmos eficientes es garantizar que la región Ω^ℓ mantenga la estructura de producto cartesiano (por pares de demanda O-D) de Ω. Larsson y Patiksson [140] emplea conjuntos que preservan esta propiedad en la descomposición simplicial desagregada (DSD) en el contexto de modelos de optimización. Estos conjuntos serán empleados en los algoritmos de este capítulo.

En principio asumiremos que está siendo empleado el CGPVIP$_{SD}(\ell)$. Este subproblema genera en cada iteración un hipercamino por cada par de demanda O-D ω (denotado por p_ω^ℓ). Consideramos la descomposición por pares O-D del flujo en los hipercaminos $\bar{\mathbf{h}}^\ell$, esto es

$$\bar{\mathbf{h}}^\ell = (\bar{\mathbf{h}}_\omega^\ell)_{\omega\in W},$$

donde $\bar{\mathbf{h}}_\omega^\ell$ es el flujo en los hipercaminos asociados con el par O-D ω en la iteración ℓ.

En función del actual conjunto de hipercaminos para el par O-D ω, denotado por $\mathcal{Q}_\omega^{\ell-1}$, y de sus flujos, denotado por $\mathcal{P}_\omega^{\ell-1}$, se define en la iteración ℓ

$$\mathcal{P}_\omega^\ell = \mathcal{P}_\omega^{\ell-1} \cup \{\bar{\mathbf{h}}_\omega^\ell\},$$
$$\mathcal{Q}_\omega^\ell = \mathcal{Q}_\omega^{\ell-1} \cup \{p_\omega^\ell\},$$

y el conjunto Ω^ℓ se define por

$$\Omega^\ell = \prod_{\omega\in W} \Omega_\omega^\ell, \tag{1.39}$$

donde $\Omega_\omega^\ell = \texttt{conv}\,(\mathcal{P}_\omega^\ell)$, siendo $\texttt{conv}\,(\mathcal{P}_\omega^\ell)$ la envoltura convexa de los puntos de $\mathcal{P}_\omega^\ell$. Como CGPVIP$_{SD}(\ell)$ asigna la demanda total $\bar{g}_\omega$ al hipercamino p_ω^ℓ, entonces las columnas $\mathcal{P}_\omega^\ell$ son de la forma $(0,\dots,\bar{g}_\omega,\dots,0)$, y los conjuntos Ω_ω^ℓ tienen la representación

$$\Omega_\omega^\ell = \left\{ \mathbf{h}_\omega = (h_p)_{p\in\mathcal{Q}_\omega^\ell} \Big| \sum_{p\in\mathcal{Q}_\omega^\ell} h_p = \bar{g}_\omega, \quad h_p \geq 0 \ \forall p \in \mathcal{Q}_\omega^\ell \right\}.$$

Si se emplea CGPVIP$_E$, entonces se generan varios hipercaminos con flujo positivo por cada par O-D. Más concretamente, se genera un hipercamino por cada modo y por cada nodo de transferencia. Las columnas de $\mathcal{P}_\omega^\ell$ son de la forma

$$\left(0,\dots,0,g_\omega^a,0,\dots,0,g_\omega^b,0,\dots,0,g_{\omega,t_1}^c,0,\dots,0,g_{\omega,t_n}^c,0,\dots,0,g_\omega^d,0,\dots,0\right).$$

Estos tipos de puntos conducen a una región factible para el RMPVIP más compleja que las obtenida por el CGPVIP$_{SD}(\ell)$. Esta dificultad puede ser evitada definiendo el conjunto $\mathcal{Q}_\omega^\ell = \mathcal{Q}_\omega^{\ell-1} \cup \{\{p_\omega^k\}_{k\in\{a,b,c,d\}}, \{p_{\omega,t}^c\}_{t\in T_\omega}\}$, esto es, añadiendo todos los hipercaminos con flujo positivo. Esta modificación incrementa el tamaño del RMPVIP(ℓ) porque se añade una nueva columna por cada hipercamino con flujo positivo, pero la región resultante tiene una estructura más sencilla (la misma que en el caso de CGPVIP$_{SD}(\ell)$).

En esta sección abordamos la resolución de RMPVIP($\bar{\mathbf{C}} - \Lambda, \Omega^\ell$) mediante un método de linealización (Wu y otros [240] y Montero y Barceló [174]). Este método genera una sucesión de problemas de optimización no lineal de la manera siguiente: sea s el contador de la sucesión de estos problemas no lineales. Sea $\hat{\mathbf{h}}^s$ una solución factible para el RMPVIP(ℓ), esto es $\hat{\mathbf{h}}^s \in \Omega^\ell$, entonces la función de coste se aproxima por la aplicación lineal y simétrica

$$\bar{\mathbf{C}}^s(\mathbf{h}) = \bar{\mathbf{C}}(\hat{\mathbf{h}}^s) + \frac{1}{\alpha}\mathbf{B}(\hat{\mathbf{h}}^s)^T(\mathbf{h} - \hat{\mathbf{h}}^s), \tag{1.40}$$

donde $\mathbf{B}(\hat{\mathbf{h}}^s)$ es la matriz diagonal de la matriz Jacobiana de $\bar{\mathbf{C}}$ evaluada en $\hat{\mathbf{h}}^s$ (método linealizado de Jacobi o abreviadamente LJ) o $\mathbf{B}(\hat{\mathbf{h}}^s)$ es alguna matriz fija $\bar{\mathbf{B}}$ que es simétrica y definida positiva (método de proyección, PM abreviadamente). El análisis clásico de la convergencia de este método recae sobre la propiedad contractiva del operador proyección. La elección del parámetro α determina la convergencia del método bajo la suposición de que $\bar{\mathbf{C}}(\mathbf{h})$ es fuertemente monótona y continuamente lipschitziana en Ω^ℓ.

En ambos casos el RMVIP($\bar{\mathbf{C}} - \Lambda, \Omega^\ell$) se aproxima en el punto $\hat{\mathbf{h}}^s$ por RMVIP($\bar{\mathbf{C}}^s - \Lambda, \Omega^\ell$) y este problema simétrico VIP puede ser formulado mediante el siguiente problema de optimización

$$\begin{array}{ll} \text{minimizar} & Z = \bar{\mathbf{C}}(\hat{\mathbf{h}}^s)^T(\mathbf{h} - \hat{\mathbf{h}}^s) + \frac{1}{2\alpha}(\mathbf{h} - \hat{\mathbf{h}}^s)^T\mathbf{B}(\hat{\mathbf{h}}^s)(\mathbf{h} - \hat{\mathbf{h}}^s) + R(\mathbf{g}) \\ \text{sujeto a} & \mathbf{h} \in \Omega^\ell, \\ & \mathbf{g} = \delta^{\mathbf{g}}\mathbf{h}, \end{array} \tag{1.41}$$

donde

$$\begin{aligned} R(\mathbf{g}) &= \sum_{\omega\in W} R_\omega(\mathbf{g}_\omega) = \sum_{\omega\in W}\Bigg[U_\omega^{d*} g_\omega^d + (1/\beta_1) \sum_{k\in\{a,b,c,d\}} g_\omega^k(\ln g_\omega^k - 1 + \alpha^k) \\ &- (1/\beta_2)\sum_{\omega\in W} g_\omega^c(\ln g_\omega^c - 1) + (1/\beta_2)\sum_{t\in T_\omega} g_{\omega,t}^c(\ln g_{\omega,t}^c - 1 + \alpha_t^c)\Bigg] \end{aligned}$$

y Ω^ℓ está definida por (1.39). Si $\mathbf{B}(\hat{\mathbf{h}}^s)$ es una matriz diagonal, (1.41) se descompone en una colección de problemas convexos e independientes, uno por cada par O-D ω.

$$\begin{array}{ll} \text{minimizar} & Z = \sum_{p\in\mathcal{Q}_\omega^\ell}\left[\bar{C}_p(\hat{h}_p^s)(h_p - \hat{h}_p^s) + \frac{1}{2\alpha}B_p(\hat{h}_p^s)(h_p - \hat{h}_p^s)^2\right] + R_\omega(\mathbf{g}_\omega) \\ \text{sujeto a} & \mathbf{h}_\omega \in \Omega_\omega^\ell, \\ & \mathbf{g}_\omega = \delta_\omega^{\mathbf{g}}\mathbf{h}_\omega, \end{array} \qquad [\text{OP}_\omega^\ell(s)]$$

donde $\delta^{\mathbf{g}}_{\omega}$ es la submatriz de $\delta^{\mathbf{g}}$ asociada al par de demanda ω. Este problema puede ser resuelto mediante una algoritmo tipo Evans, donde el cálculo de los hipercaminos mínimos es fácilmente realizable. Este algoritmo está recogido en la tabla 1.2.

Tabla 1.2: Algoritmo de resolución del problema $\mathrm{OP}^{\ell}_{\omega}(\mathrm{s})$

0. (*Inicialización*): Sea ϵ un parámetro de tolerancia. Sea un par $\omega \in W$. Sea $\mathcal{Q}^{\ell}_{\omega}$ un subconjunto de hipercaminos de $\mathcal{G}$. Tomar $i := 0$, $\hat{\mathbf{h}}^i_{\omega} = \mathbf{h}^{s-1}_{\omega}$ donde $\mathbf{h}^{s-1}_{\omega}$ es la solución de $\mathrm{OP}^{\ell}_{\omega}(s-1)$. Tomar $\mathbf{g}^i_{\omega} = \delta^{\mathbf{g}}_{\omega}\hat{\mathbf{h}}^i_{\omega}$.
1. (*Cálculo del hipercamino mínimo*): Calcular

$$U^{k*}_{\omega} = \min\left\{\bar{C}_p(\hat{\mathbf{h}}^i_{\omega}) \mid p \in \mathcal{Q}^{\ell}_{\omega} \cap P^k_{\omega}\right\} \quad k \in \{a,b\},$$

y $p^{k*}_{\omega} = \arg\min\left\{\bar{C}_p(\hat{\mathbf{h}}^i_{\omega}) \mid p \in \mathcal{Q}^{\ell}_{\omega} \cap P^k_{\omega}\right\} \quad k \in \{a,b\}$.
Calcular

$$U^{k*}_{\omega,t} = \min\left\{\bar{C}_p(\hat{\mathbf{h}}^i_{\omega}) \mid p \in \mathcal{Q}^{\ell}_{\omega} \cap P^c_{\omega,t}\right\}, \tag{1.42}$$

y $p^{c*}_{\omega,t} = \arg\min\left\{\bar{C}_p(\hat{\mathbf{h}}^i_{\omega}) \mid p \in \mathcal{Q}^{\ell}_{\omega} \cap P^c_{\omega,t}\right\}$. Empleando (1.42) y (1.3) calcular el coste U^{c*}_{ω}.
2. (*Asignación a hipercaminos*): Empleando las fórmulas (1.1), (1.2),(1.3),(1.4) y (1.5), donde los costes óptimos $\mathbf{U}^*_{\omega}$ son obtenidos en el paso 1 calcular una nueva demanda. Denotarla por $\mathbf{q}^i_{\omega}$. Asignar toda la demanda $\mathbf{q}^i_{\omega}$ al hipercamino óptimo $\{p^{k^*}_{\omega}\}_{k\in\{a,b,d\}}$ y $\{p^{c*}_{\omega,t}\}_{t\in T_{\omega}}$. Denotar este por $\tilde{\mathbf{h}}^i_{\omega}$.
3. (*Búsqueda lineal*): Realizar una búsqueda lineal en la dirección $\mathbf{d}^i := (\tilde{\mathbf{h}}^i_{\omega}, \mathbf{q}^i_{\omega}) - (\hat{\mathbf{h}}^i_{\omega}, \mathbf{g}^i_{\omega})$ en el punto $(\hat{\mathbf{h}}^i_{\omega}, \mathbf{g}^i_{\omega})$ para el problema $\mathrm{OP}^{\ell}_{\omega}(s)$. Tomar α^i una búsqueda inexacta del problema unidimensional.
4. (*Criterio de terminación*): Si $\|\alpha^i\mathbf{d}^i\| \leq \epsilon$ entonces parar, y tomar $\mathbf{h}^s_{\omega} = \hat{\mathbf{h}}^i_{\omega}$. En caso contrario ir al paso 5.
5. (*Actualizar*): Sea $(\hat{\mathbf{h}}^{i+1}_{\omega}, \mathbf{g}^{i+1}_{\omega}) = (\hat{\mathbf{h}}^i_{\omega}, \mathbf{g}^i_{\omega}) + \alpha^i\mathbf{d}^i$. Hacer $i := i+1$ y volver al paso 1.

La convergencia del método de linealización aplicado al RMPVIP($\bar{\mathbf{C}} - \Lambda, \Omega^{\ell}$) puede ser garantizada bajo las suposiciones de que la aplicación de coste $\bar{\mathbf{C}}(\mathbf{h})$ es fuertemente monótona y continuamente lipschitziana, $\mathbf{B}(\hat{\mathbf{h}}^s)$ no varía "demasiado" a lo largo del proceso iterativo, y asumiendo ciertos valores de α. Si el método de proyección fuese empleado, la segunda condición se cumpliría automáticamente y los valores de α suficientemente pequeños, cumplirían la tercera condición.

1.3.2 TAP-MVIP en el espacio de flujo en los arcos

Hemos desarrollado dos algoritmos para el problema formulado en el espacio de flujo en los hipercaminos. Ahora discutimos el caso de que el modelo TAP-M pueda ser formulado únicamente en el espacio de flujo en los arcos. Esta propiedad depende del modelo elegido de TEAP. Un ejemplo de esta situación es el modelo desarrollado en Fernández y otros [73]. Supondremos que el TAP-MVIP($\bar{\mathbf{C}} - \Lambda, \Omega$) tiene la siguiente formulación en el espacio de flujo en los arcos. Encontrar un $(\mathbf{f}^*, \mathbf{g}^*) \in \Omega^{\mathbf{g}}_{\mathbf{f}}$ que verifique

$$\mathbf{c}(\mathbf{f}^*)^T(\mathbf{f}-\mathbf{f}^*) - \Lambda(\mathbf{g}^*)^T(\mathbf{g}-\mathbf{g}^*) \geq 0, \quad \forall(\mathbf{f},\mathbf{g}) \in \Omega^{\mathbf{g}}_{\mathbf{f}}, \qquad [\text{TAP-MVIP}(\mathbf{c}-\Lambda, \Omega^{\mathbf{g}}_{\mathbf{f}})]$$

donde $\Omega^{\mathbf{g}}_{\mathbf{f}}$ es el espacio de flujo en los arcos y demandas. Este conjunto se puede formular por

$$\Omega^{\mathbf{g}}_{\mathbf{f}} = \left\{(\mathbf{f},\mathbf{g}) \mid \mathbf{f} = \delta^{\mathbf{f}}\mathbf{h} \text{ y } \mathbf{g} = \delta^{\mathbf{g}}\mathbf{h} \text{ para algún } \mathbf{h} \in \Omega\right\}.$$

El objetivo de esta sección es especializar los dos algoritmos CG/SD, discutidos en la sección anterior, y mostrar que en este caso son equivalentes a aplicar un algoritmo CG/SD al problema general de asignación de tráfico. Las consideraciones efectuadas para este modelo (ver por ejemplo: Lawphongpanich y Hearn [144], Pang y Yu [192], Marcotte y Guélat [161], Montero y Barceló [174]) también pueden ser aplicadas al TAP-MVIP($\mathbf{c}-\Lambda, \Omega_{\mathbf{f}}^{\mathbf{g}}$).

TAP-MVIP($\mathbf{c}-\Lambda, \Omega_{\mathbf{f}}^{\mathbf{g}}$) es un problema de desigualdades variacionales no lineales restringido linealmente. Podemos aplicar el algoritmo CG/SD a este caso. CGPVIP$_\varphi$ no es diferente del discutido en la sección 1.3, exceptuando que la solución debe estar en el espacio de flujos en arcos y demandas. Este problema puede ser resuelto calculando la solución del CGPVIP$_\varphi(\ell)$, empleando los algoritmos desarrollados en los apéndices I o II, y proyectándola en la región $\Omega_{\mathbf{f}}^{\mathbf{g}}$ mediante las expresiones:

$$\bar{\mathbf{f}}^\ell = \delta^{\mathbf{f}}\bar{\mathbf{h}}^\ell,$$
$$\bar{\mathbf{g}}^\ell = \delta^{\mathbf{g}}\bar{\mathbf{h}}^\ell.$$

donde $\bar{\mathbf{h}}^\ell$ denota la solución obtenida en el espacio de flujos en los hipercaminos.

Consideraremos que la región factible del RMPVIP para la formulación en el espacio de flujos en los arcos será similar a la desarrollada en la descomposición simplicial agregada para el TAP. Supondremos que $\mathcal{P}_s^\ell$ es el conjunto de columnas retenidas al comienzo de la iteración ℓ. Definimos

$$\mathcal{P}^\ell = \mathcal{P}_s^\ell \bigcup \left\{ \begin{pmatrix} \bar{\mathbf{f}}^\ell \\ \bar{\mathbf{g}}^\ell \end{pmatrix} \right\}.$$

Entonces la región factible se define por $\tilde{\Omega}^\ell = \texttt{conv}\,(\mathcal{P}^\ell)$, y el RMPVIP($\ell$) se transforma en el siguiente problema de desigualdades variacionales.

$$\mathbf{c}(\mathbf{f}^\ell)^T(\mathbf{f}-\mathbf{f}^\ell) - \Lambda(\mathbf{g}^\ell)^T(\mathbf{g}-\mathbf{g}^\ell) \geq \epsilon, \quad \forall(\mathbf{f},\mathbf{g}) \in \tilde{\Omega}^\ell. \qquad [\text{RMPVIP}(\mathbf{c}-\Lambda, \tilde{\Omega}^\ell)]$$

Este problema puede ser reformulado empleando la representación simplicial del conjunto $\tilde{\Omega}^\ell$. Denotamos por $[\mathcal{P}^\ell]$ la matriz asociada al conjunto de puntos de $\mathcal{P}^\ell$; $\mathbf{A}_{\mathbf{f}}^\ell$ y $\mathbf{A}_{\mathbf{g}}^\ell$ las submatrices de $[\mathcal{P}^\ell]$ relativas a los flujos en los arcos y a las demandas. La representación simplicial del conjunto $\tilde{\Omega}^\ell$ es

$$\tilde{\Omega}^\ell = \left\{ (\mathbf{f},\mathbf{g})^T \mid \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix} = \begin{bmatrix} A_{\mathbf{f}}^\ell \\ A_{\mathbf{g}}^\ell \end{bmatrix} \beta \ \text{ tal que } \mathbf{1}^T\beta = 1,\ \beta \geq 0 \right\}.$$

Ahora introducimos una nueva función para poder formular el RMPVIP empleando como nuevas variables los coeficientes β. Este cambio de variable conduce a una reducción de las dimensiones del problema y a unas restricciones de símplice que son más sencillas que las de un poliedro general. Definimos $\tilde{\mathbf{c}}(\beta) : \Re^m \mapsto \Re^{|A\cup B|}$ como $\tilde{\mathbf{c}}(\beta) = \mathbf{A}_{\mathbf{f}}^{\ell\,T}\mathbf{c}(\mathbf{A}_{\mathbf{f}}^\ell\beta)$; y $\tilde{\Lambda}(\beta) : \Re^m \mapsto \Re^n$ como $\tilde{\Lambda}(\beta) = \mathbf{A}_{\mathbf{g}}^{\ell\,T}\Lambda(\mathbf{A}_{\mathbf{g}}^\ell\beta)$ donde $m = \left|\mathcal{P}^\ell\right|$, y n es la cantidad de componentes del vector demanda $\mathbf{g}$. El problema formulado en téminos de las nuevas variables consiste en encontrar un $\beta^\ell \in \Theta^\ell$ cumpliendo

$$\mathbf{c}(\mathbf{A}_{\mathbf{f}}^\ell\beta^\ell)^T(\mathbf{A}_{\mathbf{f}}^\ell\beta - \mathbf{A}_{\mathbf{f}}^\ell\beta^\ell) - \Lambda(\mathbf{A}_g^\ell\beta^\ell)^T(\mathbf{A}_g^\ell\beta - \mathbf{A}_g^\ell\beta^\ell) =$$
$$\left[\tilde{\mathbf{c}}(\beta^\ell) - \tilde{\Lambda}(\beta^\ell)\right]^T(\beta-\beta^\ell) \geq 0, \quad \forall\beta \in \Theta^\ell, \qquad [\text{VIP}(\mathbf{c}-\Lambda, \Theta^\ell)]$$

donde $\Theta^\ell = \left\{\beta \mid \mathbf{1}^T\beta = 1,\ \beta \geq 0\right\}$.

Notar que la formulación de RMPVIP($\mathbf{c}-\Lambda, \tilde{\Omega}^\ell$) es equivalente a la formulación RMPVIP($\mathbf{c}-\Lambda, \Theta^\ell$). Por tanto β^* es una solución de RMPVIP($\mathbf{c}-\Lambda, \Theta^\ell$) si y sólo si $(\mathbf{f}^*, \mathbf{g}^*)$, con $\mathbf{f}^* = \mathbf{A}_{\mathbf{f}}^\ell\beta^*$ y $\mathbf{g}^* = \mathbf{A}_{\mathbf{g}}^\ell\beta^*$, es una solución de RMPVIP($\mathbf{c}-\Lambda, \tilde{\Omega}^\ell$).

Bertsekas y Gafni [21] mostraron que si $T(\mathbf{f},\mathbf{g}) = [\mathbf{c}(\mathbf{f}), -\Lambda(\mathbf{g})]^T$ es continuamente lipschitziana y fuertemente monótona sobre $\tilde{\Omega}^\ell$, entonces una solución del problema RMPVIP($\mathbf{c}-\Lambda, \Theta^\ell$) se puede encontrar mediante el método de proyección, definido por

$$\beta^{i+1} = P_{\Theta^\ell}^{\mathbf{S}}[\beta^i - \alpha\mathbf{A}^{\ell\,T}T(\mathbf{A}^\ell\beta^i)], \quad \beta^0 \in \Theta^\ell,$$

supuesto que el tamaño del paso $\alpha > 0$ sea suficientemente pequeño y que $\mathbf{S}$ sea una matriz definida positiva y simétrica. $P^{\mathbf{S}}_{\Theta^\ell}(\mathbf{z})$ denota la única proyección de $\mathbf{z}$ en el conjunto Θ^ℓ con respecto a la norma $\|\cdot\|$ correspondiente a $\mathbf{S}$ (la norma inducida por el producto interior $\mathbf{x}^T\mathbf{S}\mathbf{x}$).

La siguiente proposición muestra que la anterior condición suficiente para la convergencia del método de proyección depende únicamente de la aplicación de costes en los arcos. Notar que este resultado muestra propiedades sobre la convergencia del método empleando las variables β, sin embargo las hipótesis se dan en término de las variables originales $(\mathbf{f}, \mathbf{g})$.

PROPOSICIÓN 1.3.1 *Suponiendo que* $\mathbf{c}(\mathbf{f})$ *es continuamente lipschitziana y fuertemente monótona en*

$$\tilde{\Omega}^\ell_{\mathbf{f}} = \{\mathbf{f} \mid \text{ existe } \mathbf{g} \text{ tal que } (\mathbf{f}, \mathbf{g}) \in \tilde{\Omega}^\ell\},$$

y que $\beta_1 > 0,\ \beta_2 > 0,\ \beta_2 - \beta_1 > 0$ *entonces* $T(\mathbf{f}, \mathbf{g}) = [\mathbf{c}(\mathbf{f}), -\Lambda(\mathbf{g})]^T$ *es continuamente lipschitziana y fuertemente monótona en* $\tilde{\Omega}^\ell$.

DEMOSTRACIÓN. El teorema 5.4.3 de Ortega y Rheinboldt [188] da una caracterización de aplicación fuertemente monótona que puede ser usada para garantizar que la aplicación $-\Lambda(\mathbf{g})$ cumple esta propiedad en

$$\tilde{\Omega}^\ell_{\mathbf{g}} = \left\{\mathbf{g} \mid \text{ existe } \mathbf{f} \text{ tal que } (\mathbf{f}, \mathbf{g}) \in \tilde{\Omega}^\ell\right\}.$$

Esta propiedad se establece empleando la siguiente condición suficiente. Si $-\Lambda(\mathbf{g})$ es una aplicación de clase $\mathcal{C}^1$ en $\tilde{\Omega}_{\mathbf{g}}$ y

$$\mathbf{y}^T(-\nabla\Lambda(\mathbf{g}) - m_\Lambda \mathbf{I})\mathbf{y} \geq 0, \quad \forall \mathbf{g} \in \tilde{\Omega}^\ell_{\mathbf{g}}, \forall \mathbf{y} \in \Re^n, \tag{1.43}$$

entonces $-\Lambda(\mathbf{g})$ es fuertemente monótona sobre $\Omega^\ell_{\mathbf{g}}$.

La aplicación de coste $-\Lambda(\mathbf{g})$ tiene una matriz Jacobiana diagonal. Sus elementos de la diagonal son de la forma $\frac{\psi_i}{g_i}$ donde $\psi_i \in \{1/\beta_1, 1/\beta_2, 1/\beta_1 - 1/\beta_2\}$ y g_i es la $i-$ésima componente del vector $\mathbf{g}$. La constante ψ_i es positiva por hipótesis. Entonces la condición (1.43) es equivalente a mostrar que existe una constante positiva m_Λ cumpliendo que $g_i \leq \frac{m_\Lambda}{\psi_i}$ para todo $\mathbf{g} = (g_i) \in \tilde{\Omega}^\ell_g$. Esta relación se satisface porque el conjunto $\tilde{\Omega}_{\mathbf{g}}$ está acotado, obteniendo que $-\Lambda(\mathbf{g})$ es fuertemente monótona en $\tilde{\Omega}^\ell_{\mathbf{g}}$. Esto es

$$[-\Lambda(\mathbf{g}) + \Lambda(\mathbf{q})]^T (\mathbf{g} - \mathbf{q}) \geq m_\Lambda \|\mathbf{g} - \mathbf{q}\|^2, \quad \forall \mathbf{g}, \mathbf{q} \in \tilde{\Omega}^\ell_{\mathbf{g}}.$$

Por hipótesis,

$$[\mathbf{c}(\mathbf{f}) - \mathbf{c}(\mathbf{v})]^T (\mathbf{f} - \mathbf{v}) \geq m_c \|\mathbf{f} - \mathbf{v}\|^2, \quad \forall \mathbf{f}, \mathbf{v} \in \tilde{\Omega}^\ell_{\mathbf{f}},$$

y sumando ambas relaciones

$$[T(\mathbf{f}, \mathbf{g}) - T(\mathbf{v}, \mathbf{q})]^T \begin{pmatrix} \mathbf{f} - \mathbf{v} \\ \mathbf{g} - \mathbf{q} \end{pmatrix} \geq \min\{m_\Lambda, m_c\} \left\| \begin{pmatrix} \mathbf{f} - \mathbf{v} \\ \mathbf{g} - \mathbf{q} \end{pmatrix} \right\|^2, \ \forall (\mathbf{f}, \mathbf{g}),\ (\mathbf{v}, \mathbf{q}) \in \tilde{\Omega}^\ell,$$

y $T(\mathbf{f}, \mathbf{g})$ es una aplicación fuertemente monótona en $\tilde{\Omega}^\ell$.

La aplicación $-\Lambda$ es continuamente lipschitziana porque es continuamente diferenciable en $\tilde{\Omega}_{\mathbf{g}}$. Por hipótesis $\mathbf{c}$ es continuamente lipschitziana en $\tilde{\Omega}^\ell_{\mathbf{f}}$, por tanto T también lo es en $\tilde{\Omega}^\ell$. □

La principal conclusión de esta sección es que cualquier algoritmo propuesto para resolver el RMP para el problema general de asignación de tráfico también puede ser empleado para resolver el RMPVIP$(\mathbf{c} - \Lambda, \Theta)$ del TAP-M. Para asegurar la convergencia del algoritmo CG/SD debemos probar que los RMPVIPs son resueltos exactamente. La hipótesis de que $\mathbf{c}(\mathbf{f})$ es fuertemente monótona garantiza que el método de proyección es válido para resolver el RMPVIP.

La adaptación de los algoritmos CG/SD aplicados al problema TAP-M definido en el espacio de flujo en los arcos se puede realizar haciendo una pequeña modificación en la fase de generación de columnas. Ésta es la principal diferencia entre el TAP y TAP-M. Estos resultados serán usados en la próxima sección para realizar los experimentos numéricos.

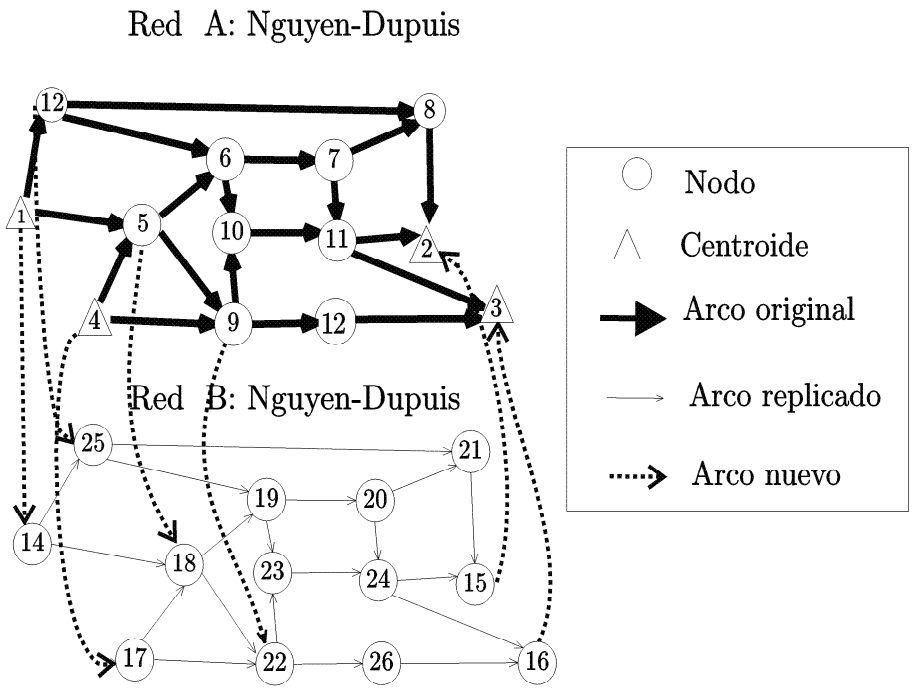


Figura 1.2: Duplicación de la red Nguyen-Dupuis

1.4 Resultados experimentales

En esta sección se desarrollaran experimentos numéricos para el caso de costes simétricos (modelo del apéndice III).

No conocemos redes reales del modelo TAP-M en la literatura. Por esta razón, los problemas de prueba han sido definidos a partir de los problemas de asignación de tráfico de la tabla 1.3.

Tabla 1.3: Redes de tráfico

Nombre	$\|N^a\|$	$\|A\|$	$\|W\|$	# Centroide	Referencia
Nguyen y Dupuis	13	19	4	4	Nguyen y Dupuis [179]
Sioux Falls	24	76	528	24	LeBlanc y otros [149]
Hull	501	798	142	23	Florian [80]

La topología de las redes de pruebas se define duplicando la red de tráfico. La figura 1.2 muestra este procedimiento aplicado a la red de Nguyen y Dupuis [179]. Dos conjuntos de nuevos arcos han sido añadidos. El primero une los centroides con sus réplicas en la red B. La dirección de los arcos depende de si el centroide es origen o destino. El segundo conjunto une los nodos adyacentes desde un origen (nodos de trasferencia) con sus réplicas en la red B.

Las redes de prueba son de pequeño tamaño. Sus dimensiones se muestran en la tabla 1.4. $|\mathcal{N}|$ es el número de nodos, $|A \cup B|$ es el número de arcos, $|W|$ es el número de pares O-D. En los resultados numéricos hemos considerado exclusivamente los modos (a), (b), y (c). El número de variables demanda es la cantidad de variables $\{g^k_\omega\}_{k\in\{a,b,c\}\ \omega\in W}$, $\{g^c_{\omega,t}\}_{t\in T_\omega\ \omega\in W}$. El número total de variables es la suma de variables de demanda más el número de arcos de la red.

La principal motivo para elegir redes de pequeño tamaño es que su duplicación incrementa sustancialmente su tamaño. El número total de variables es aproximadamente seis veces el número de pares O-D más dos veces el número de arcos de la red original.

Hemos duplicado la matriz O-D original. La partición modal depende de los parámetros del modelo

Tabla 1.4: Dimensiones de las redes multimodales de prueba

Problema	$\|\mathcal{N}\|$	$\|A \cup B\|$	$\|W\|$	# Centroides	# Demanda variables	# Variables
NgD2	26	45	4	4	20	65
SiF2	48	224	528	24	3263	3487
Hul2	1002	1678	142	23	757	2435

logit y de los costes de viaje en cada alternativa. Estos parámetros están recogidos en la tabla 1.5. Hemos elegido todos los parámetros logit igual a 1, lo que significa que el atractivo de cada alternativa sólo depende de su coste de viaje. Para simplificar hemos supuesto que la tasa de ocupación vehicular es de una persona por vehículo, esto es $\gamma_\omega = 1$. Hemos considerado diferentes valores de los parámetros β_1, β_2 en cada red.

Tabla 1.5: Parámetros logit para las redes de prueba

Problema	β_1	β_2	θ_a	θ_b	γ_ω	α_t^c	α^k
NgD2	2.00	4.00	1.0	1.0	1.0	1.0	1.0
SiF2	1.00	1.20	1.0	1.0	1.0	1.0	1.0
Hul2	1.00	1.50	1.0	1.0	1.0	1.0	1.0

Los modelos de asignación de tráfico suelen emplear funciones del tipo BPR ([185]) para describir el coste de viaje en los arcos. Esta tiene la expresión

$$c_l(f_l) = c_l^0 \left[1 + \left(\frac{f_l}{k_l}\right)^{m_l}\right],$$

donde c_l^0 es el coste de viaje en el arco l sin flujo, $m_l \geq 1$, y k_l es su *capacidad*. Las funciones de coste en los arcos para los problemas de prueba son

$$c_l(f_l) = c_l^0 \left[\left(\frac{f_l}{k_l}\right)^{m_l}\right],$$

para los arcos de la red original y sus réplicas, y para el resto de los arcos $c_l(f_l) = 0$.

1.4.1 Detalles de la implementación

Los problemas de prueba tienen dos propiedades fundamentales que nos permiten especializar los dos algoritmos CG/SD. La primera propiedad es que la matriz Jacobiana de la función de costes es diagonal, garantizando que el problema VIP($\bar{\mathbf{C}} - \Delta, \Omega$) pueda ser formulado mediante un problema de optimización (ver apéndice III). Los algoritmos aquí propuestos en el contexto de problemas de optimización pertenecen a la clase de descomposición simplicial no lineal (NSD) de Larsson y otros [141]. Esta clase generaliza el algoritmo RSD de Hearn y otros [123, 125] permitiendo usar columnas no lineales tales como las obtenidas de los subproblemas tipo Evans. La segunda propiedad es que el sistema de transporte público (duplicación de la red de tráfico) puede ser formulado en el espacio de flujo en los arcos. Esta propiedad permitirá trabajar con las variables de demanda como si fueran flujos en los arcos.

Estas dos características conducen a:

- ◇ CGPVIP$_\varphi$ se resuelve empleando el algoritmo del apéndice I y II, y los flujos de los caminos se asignarán al espacio de flujos en los arcos y de demanda.

- ◇ RMPVIP de TAP es equivalente al RMPVIP para el problema simétrico de tráfico. Las variables de demanda son consideradas como los flujos en los arcos. El coste de estas variables se puede observar en el problema de optimización del apéndice III.

En el paso 4 del algoritmo CG/SD hemos empleado las reglas de eliminación y adición de columnas de Hearn y otros [123, 125]. Hemos considerado un parámetro r para controlar el tamaño del problema maestro. Cuando r es mayor que el número de variables del problema y se emplea como subproblema de generación de columnas CGPVIP$_{SD}$, el algoritmo CG/SD es el de descomposición simplicial (SD) (ver Holloway [128]); cuando $r = 1$, el método se transforma en el algoritmo de Frank-Wolfe [88]. El algoritmo E es transformado al algoritmo de Evans [72] cuando $r = 1$ y el subproblema de generación de columnas es CGPVIP$_E$.

Para obtener la solución de CGPVIP, estos algoritmos calculan un problema de caminos mínimos multiproducto en las redes $\mathcal{G}^a$ y $\mathcal{G}^b$. En la primera red el conjunto de demandas es W^a y en la segunda es W^b. El subproblema de Frank-Wolfe asigna toda la demanda al camino con menor *coste extendido*, donde este valor tiene en cuenta el coste del viaje, la partición modal y la distribución por nodos de transferencia. El subproblema de Evans emplea los caminos mínimos dentro de cada alternativa de viaje (sólo coche, sólo transporte público, modo combinado a través del intercambiador t_1,..., modo combinado a través del intercambiador t_s) y asigna el flujo en estos caminos de acuerdo al modelo logit anidado y de sus costes. Hemos empleado el algoritmo L2QUE de Gallo y Pallotino [92] para resolver el problema de caminos mínimos. El RMP lo hemos resuelto mediante el método de Newton proyectado de Bertsekas [18]. Este algoritmo tiene convergencia superlineal.

Los programas fuentes han sido escritos en *FORTRAN Visual Workbench* y se ha empleado la precisión simple en operaciones de representación y precisión doble cuando pudieran aparecer errores de redondeo. Por ejemplo, la precisión simple se emplea para calcular los caminos mínimos en las redes y precisión doble en las operaciones involucradas en la resolución del problema maestro. Los códigos han sido ejecutados en un PC con 384 "megabytes" de memoria RAM a 400 MHz.

La tabla 1.6 muestra los resultados experimentales para los algoritmos RSD y E. Los experimentos computacionales se han realizado para tres niveles de precisión y tres valores del parámetro r. Como ya hemos mencionado el valor $r = 1$ transforma el RSD y el algoritmo E en los algoritmos de Frank-Wolfe y Evans respectivamente, y $r = \infty$ genera el SD en un caso y un algoritmo nuevo en el otro.

Tabla 1.6: Resultados experimentales

Problema	ϵ	RSD			E		
		$r = 1$	$r = 10$	$r = \infty$	$r = 1$	$r = 10$	$r = \infty$
Hul2	0.1	1.64† (5)††	1.53 (4)	1.59 (4)	0.82 (2)	0.77 (2)	0.72 (2)
	0.01	4.44 (12)	7.85 (12)	8.40 (12)	2.91 (7)	0.77 (2)	0.72 (2)
	0.001	105.89 (228)	67.28 (79)	50.86 (33)	20.92 (48)	14.72 (21)	17.91 (20)
SiF2	0.1	8.57 (22)	11.69 (22)	11.04 (19)	2.91 (11)	3.13 (8)	3.24 (8)
	0.01	221.79 (454)	268.14 (340)	126.66 (55)	9.83 (34)	11.37 (21)	11.92 (21)
	0.001	(>2000)	(>2000)	(>100)	38.28 (123)	40.70 (63)	36.14 (42)
NgD2	0.001	7.36 (461)	7.75 (21)	7.80 (20)	2.75 (174)	4.61 (16)	3.51 (14)
	0.0001	(>3000)	209.71 (247)	14.50 (26)	(>3000)	17.69 (32)	13.01 (25)
	0.00001	(>3000)	463.57 (527)	38.12 (39)	(>3000)	65.09 (86)	25.92 (38)

† Tiempo de CPU.
†† Numero de iteraciones.

1.5 Aplicación del TAP-M al diseño paramétrico de intercambiadores

En el desarrollo y evaluación de facilidades para el intercambio de pasajeros, es necesario desarrollar herramientas adecuadas que fundamenten las decisiones adoptadas. Una de estas facilidades son los denominados intercambiadores multimodales urbanos, o simplemente intercambiadores.

Dos perspectivas pueden ser consideradas a la hora de abordar el problema: *micro* y *macro*. La metodología micro analiza los intercambiadores considerando aisladamente el tráfico en cada uno de ellos. Este valor se calcula mediante métodos directos, como son las encuestas. Estas medidas son consideradas como independientes del resto de la red de transporte. El objetivo para estos estudios es definir los "estándares" de calidad y consideran aspectos del problema relacionados con la operación del sistema, tales como garantías de conexiones, facilidades de espera, venta de billetes, información dinámica a los usuarios, evaluación de distancias dentro del intercambiador, seguridad, etc. La metodología macro analiza las influencias mutuas entre la asignación de tráfico multimodal y las facilidades diseñadas. En una metodología macro se analizan los tiempos de transferencia, capacidades y precios de las áreas de aparcamientos, conexiones, etc. en un contexto de rutas multimodales, caracterización de la demanda, asignación de recursos y otras características del sistema de transporte.

El diseño de estos intercambiadores puede ser considerado a largo plazo (planificación estratégica) o a medio plazo (planificación táctica). En la planificación estratégica se desea determinar el número de intercambiadores, su localización, así como la topología de la red. En la planificación táctica es asumida que la topología de la red es conocida y fija. En este contexto se desea analizar sus características. No se considera la planificación a corto plazo para los problemas de diseño de intercambiadores.

En esta sección ilustramos el uso del modelo TAP-M para diseñar intercambiadores multimodales urbanos. Para ello hemos considerado la red de pruebas de la figura 1.3 y aplicamos el modelo TAP-M para estimar la demanda en los intercambiadores en función de su diseño. Hemos considerado los siguientes parámetros

- ◇ Localización de los intercambiadores.
- ◇ Capacidad de los aparcamientos.
- ◇ Tiempo medio de transferencia en el intercambiador.
- ◇ Tarifas de los aparcamientos.

Hemos considerado una red de metro no congestionada con tres líneas de transporte. La primera une los centroides A-B, la segunda C-B y la tercera los intercambiadores $12 - 13$. Hemos supuesto que el coste en los arcos de la red de metro es independiente del flujo y hemos considerado un coste generalizado de la forma

$$c_l(f_l) = a_1 t_l^v + a_2 t_l^w + a_3 t_l^t + a_4 t_l^n + a_1 \delta + a_5 F_l, \quad l \in B$$

donde t_l^v es el tiempo de viaje en metro en el arco l, t_l^w es el tiempo andando hasta la parada, t_l^t es el tiempo de espera, t_l^n es el tiempo de transferencia, δ es una penalización a cambiar de línea de transporte público (5 minutos) y F_l es la tarifa del billete asociada al arco l. Hemos tomado los valores $a_1 = a_5 = 1$, $a_2 = a_3 = a_4 = 2$.

Hemos considerado una red de tráfico donde los costes en los arcos vienen descritos por funciones del tipo BPR siendo éstas de la forma

$$c_l(f_l) = t_l(1 + \nu_l(f_l/K_l)^4), \quad l \in A \tag{1.44}$$

donde c_l es el coste de viaje en el arco l para un flujo f_l, t_l es el coste de viaje en el arco l cuando está libre de flujo, ν_l es un parámetro para dar el nivel de congestión y K_l es la capacidad del arco l. Hemos considerado $\nu_l = \frac{15}{4.5}$ y $K_l = 4.5$.

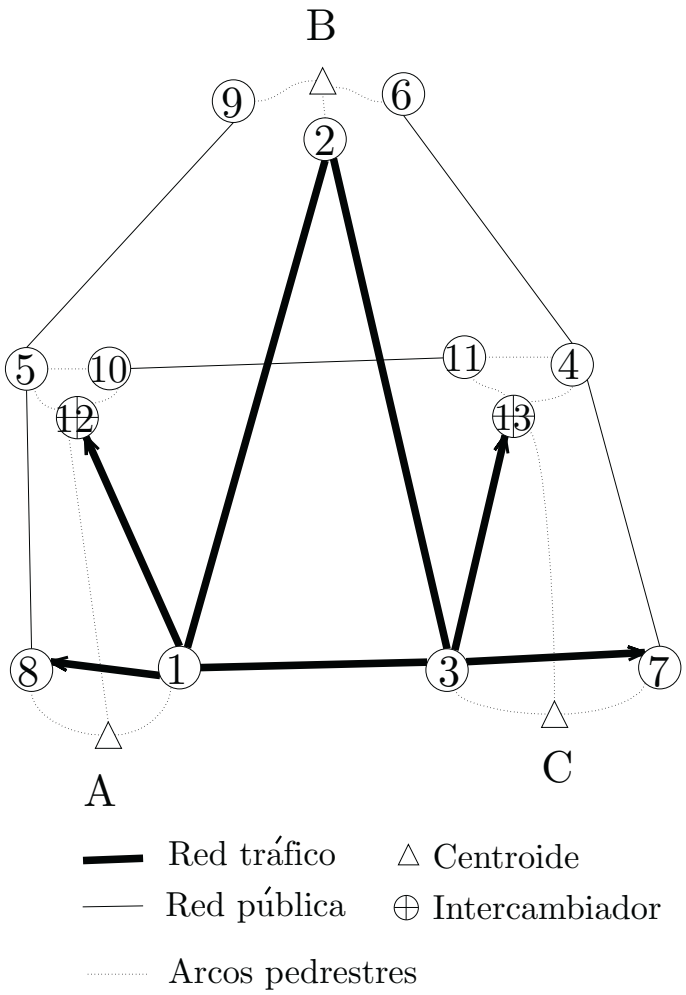


Figura 1.3: Red de pruebas GaM

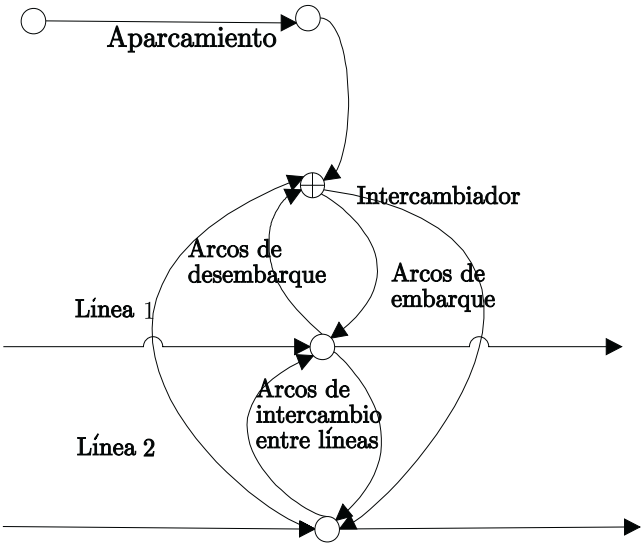


Figura 1.4: Representación de un intercambiador mediante un grafo.

Un intercambiador multimodal se representa mediante un grafo, ver figura 1.4, en el que el intercambiador es servido por una o varias líneas. Una línea se identifica por distintos pares de arcos de embarque y desembarque. Inicialmente hemos considerado cuatro potenciales intercambiadores situados en los nodos 12, 13, 7 y 8. En los arcos de tráfico, que unen los centroides con las paradas de las líneas de transporte, hemos representado el coste de viaje y el coste del aparcamiento. Este último coste representa el tiempo de aparcar (búsqueda de una plaza libre de aparcamiento), precio del billete y tiempo caminando hasta la parada de metro. La forma funcional de este coste también es del tipo BPR (5.20), donde K_l es la capacidad del aparcamiento. Una descripción completa de los parámetros de la red de pruebas se dan en la tabla 1.7

La tabla 1.8 muestra la demanda total entre los centroides y la tasa de ocupación vehicular. Los parámetros empleados en el modelo logit son $\beta_1 = 0.01$, $\beta_2 = 0.05$, $\alpha^a = 2.0$, $\alpha^b = 3.0$, $\alpha^c = 1.0$ y $\alpha_t = 1.0$. Los parámetros de ponderación de los costes en las redes son $\theta_a = 1.0$ y $\theta_b = 1$.

1.5.1 Ilustración de la solución del modelo TAP-M

En esta sección ilustramos las salidas del modelo TAP-M. Hemos considerado dos niveles de congestión en la red que vienen definidos mediante dos matrices O-D (ver tabla 1.8). Las salidas básicas del modelo son la partición modal de la demanda, los costes en equilibrio por modos y por intercambiadores (nodos de transferencia) y el nivel de servicio de cada arco de la red de transporte. Notar que el número de usuarios en los arcos asociados con los aparcamientos nos proporciona el nivel de servicio de los aparcamientos. Esta información, exceptuando los flujos en los arcos, se muestra en la tabla 1.9. Este modelo, por tanto, permite evaluar la partición modal de la demanda en función de la demanda total y de la red de transporte. En las próximas secciones veremos como el modelo puede ser utilizado en el diseño de intercambiadores, permitiendo evaluar nuevas políticas, definidas mediante cambios en la parametrización de la red de transporte.

1.5.2 Demanda de aparcamientos frente a la capacidad ofertada

Una posibilidad del modelo es evaluar la demanda de aparcamientos en función de la capacidad de los mismos. El coste de aparcamiento tiene en cuenta el tiempo empleado para buscar un espacio libre donde aparcar y la distancia media a la salida. Cuando el número de usuarios se acerca a la capacidad del aparcamiento estos costes se incrementan. Este modelo no condiera explícitamente la

Tabla 1.7: Parámetros de los costes en los arcos de la red GaM

Tipo arco	Arco	t_l or c_l	K_l	Arc	t_l or c_l	K_l
Tráfico	(2,1)	51.0	4.5	(1,2)	58.5	4.5
	(3,2)	58.9	4.5	(2,3)	56.7	4.5
	(1,3)	65.8	4.5	(3,1)	60.6	4.5
Aparcamiento	(1,12)	7.8	2.5	(1,8)	14.6	2.5
	(3,7)	13.1	2.5	(3,13)	10.7	2.5
Metro	(5,8)	16.5	–	(8,5)	16.5	–
	(9,5)	66.3	–	(5,9)	66.3	–
	(4,6)	27.3	–	(6,4)	27.3	–
	(7,4)	14.5	–	(4,7)	14.5	–
	(10,11)	50.1	–	(11,10)	50.1	–
Andando	(7,3)	24.8	–	(3,7)	24.8	–
	(3,13)	64.7	–	(13,3)	64.7	–
	(2,6)	34.8	–	(6,2)	34.8	–
	(2,9)	14.9	–	(9,2)	14.9	–
	(1,12)	47.3	–	(12,1)	47.3	–
	(1,8)	24.8	–	(8,1)	24.8	–
Embarque/	(10,12)	7.5	—	(12,10)	15.0	–
Desembarque	(13,11)	15.0	–	(11,13)	7.5	–
	(10,5)	14.25	–	(5,10)	16.7	–
	(12,5)	7.5	–	(5,12)	5.0	–
	(13,4)	7.5	–	(4,13)	5.0	–
	(11,4)	14.25	—	(4,11)	16.7	–

Tabla 1.8: Demanda de viajeros (expresada en unidades de millar) y tasa de ocupación vehicular para cada par O-D

O–D pair	Nivel de congestión		γ_ω	O–D pair	Nivel de congestión		γ_ω
	Bajo	Alto			Bajo	Alto	
$\omega_1 = (A, B)$	4.5	5.5	1.1	$\omega_2 = (A, C)$	3.5	6.0	1.1
$\omega_3 = (C, A)$	4.0	7.5	1.1	$\omega_4 = (C, B)$	3.0	8.0	1.1

Tabla 1.9: Solución numérica de la red GaM

Nivel de congestión	Par O–D	Coche		Metro		Park'n-Ride		Intercambiador $t = 12$		Intercambiador $t = 13$	
		g_ω^a	U_ω^{a*}	g_ω^b	U_ω^{b*}	g_ω^c	L_ω^{c*}	$g_{\omega,t}^c$	$U_{\omega,t}^{c*}$	$g_{\omega,t}^c$	$U_{\omega,t}^{c*}$
Bajo	$\omega_1 = (A, B)$	1.730	55.83	2.414	122.50	.355	114.11	.3240	95.96	.0314	112.66
	$\omega_1 = (A, B)$	1.586	63.24	1.612	161.60	.303	128.77	.1289	125.86	.1740	119.86
	$\omega_3 = (C, A)$	1.829	60.66	1.825	160.90	.346	127.32	.2327	115.22	.1128	129.72
	$\omega_4 = (C, B)$	1.026	53.88	1.734	101.40	.240	99.01	.0050	156.62	.2354	79.42
Alto	$\omega_1 = (A, B)$	2.070	59.46	2.995	122.50	.435	115.47	.4137	96.48	.0212	155.73
	$\omega_1 = (A, B)$	2.572	73.96	2.917	161.60	.512	135.52	.3002	126.38	.2115	132.93
	$\omega_3 = (C, A)$	3.022	84.91	3.847	160.90	.630	141.78	.2509	139.99	.3793	132.07
	$\omega_4 = (C, B)$	2.510	67.21	4.837	101.40	.653	101.63	.0044	181.39	.6489	81.77

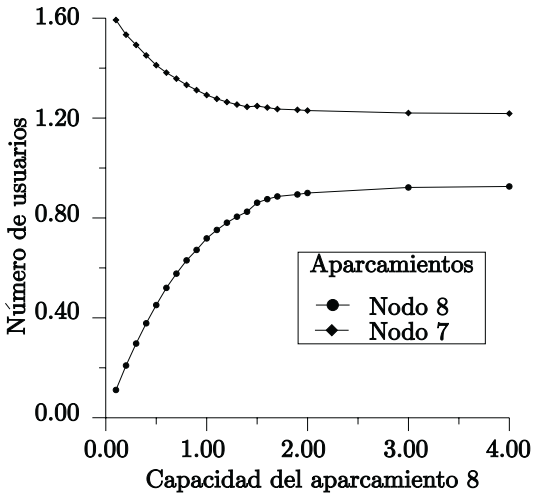


Figura 1.5: Demanda de aparcamiento frente a la capacidad ofertada de aparcamiento

restricción de que la demanda de aparcamiento no puede superar la capacidad ofertada, sin embargo si está implícitamente considerada por el hecho de que cuando se sobrepasa la capacidad, el coste de aparcamiento es tan grande que los usuarios se ven desincentivados a emplearlo. El modelo genera niveles de servicio de los aparcamientos inferiores o ligeramente superiores a su capacidad.

En este ejemplo, los costes de aparcamiento son de la forma BPR (5.20) y se estudia el efecto de la variación del parámetro K_l. Este ejemplo consta de dos aparcamientos localizados en los nodos 7 y 8. La capacidad del aparcamiento 8 es incrementada y entonces se evalúa la demanda de aparcamiento en 7 y 8 en función de la capacidad ofrecida en 8.

Los resultados son mostrados en la figura 1.5. Inicialmente un incremento de la capacidad del aparcamiento 8 produce un incremento en la demanda de este aparcamiento. Este efecto se produce porque la demanda satura la capacidad ofertada y muchos de los usuarios eligen utilizar el aparcamiento 7 que puede ser considerado como el competidor del aparcamiento 8. Cuando la oferta de aparcamiento empieza a superar a la demanda de éste, el efecto de generación de demanda en el aparcamiento 8 empieza a reducirse. Notar que el modelo TAP-M considera la competencia entre aparcamientos. El gráfico muestra, no sólo como la mejora de la capacidad del aparcamiento afecta al propio aparcamiento 8, sino a los otros aparcamientos como en este caso al 7.

Otra observación importante es que para evaluar cada par de puntos del gráfico hay que resolver el modelo TAP-M y por tanto este procedimiento requiere gran cantidad de cálculos.

Tabla 1.10: Número de usuarios de aparcamiento frente a la localización

Localización de los aparcamientos				
8	12	13	7	Demanda total
–	0.968	1.260	–	2.228
0.224	0.826	1.231	–	2.281
0.222	0.779	0.9638	0.418	2.383
1.210	–	–	0.905	2.115
–	0.921	0.998	0.419	2.338

1.5.3 Influencia de la localización de los intercambiadores en la demanda de modos combinados

TAP-M puede ser usado para evaluar la puesta en funcionamiento de un nuevo aparcamiento. Esta estimación tiene en cuenta los aparcamientos existentes, la competencia entre ellos y con el resto de alternativas de transporte. Para ilustrar este hecho hemos considerado la posibilidad de abrir cuatro aparcamientos localizados en los nodos 8, 12, 13 y 7. En la tabla 1.10 se muestra la demanda de aparcamiento en función de varias combinaciones de aparcamientos. Por ejemplo, la primera fila de la tabla representa abrir los aparcamientos 12 y 13 (el símbolo − indica que el correspondiente aparcamiento no está operativo). Esto produciría una demanda de aparcamiento de 2.228 plazas. Para el aparcamiento 12 la demanda sería de 0.968 y de 1.260 para el aparcamiento 13. La tabla 1.10 evidencia que el nivel de servicio depende de la competencia entre aparcamientos. Por ejemplo, en la segunda situación el hecho de abrir el aparcamiento 8 produce una merma en la demanda del aparcamiento 12 de 0.142 unidades.

Este ejemplo muestra (ligeramente) el efecto que tiene la apertura de nuevas facilidades en la generación de viajes combinados. La mayor demanda de viajes combinados se obtiene abriendo los cuatro aparcamientos.

1.5.4 Influencia de las tarifas en el nivel de servicio de los aparcamientos

La parametrización de la red recoge múltiples aspectos del proceso de planificación. Una posibilidad es introducir las políticas de tarifas de los aparcamientos. Para ilustrarlo hemos considerado que están abiertos los aparcamientos situados en los nodos 12, 13 y 7. Hemos introducido cambios en la tarifa del aparcamiento 7 mediante el incremento del parámetro asociado con el coste de viaje en el arco sin flujo. La tarifa es transformada en tiempo de viaje. Los resultados son mostrados en la figura 1.6. En la figura solamente se muestra la demanda en el aparcamiento 7 y 13, que son los que compiten más directamente. Un incremento de las tarifas del aparcamiento 7 reduce la cantidad de usuarios del aparcamiento. Parte de esta demanda es derivada al aparcamiento 13.

1.5.5 Influencia del tiempo medio de transferencia en el nivel de servicio de los intercambiadores

Otra posibilidad que permite la parametrización de la red es evaluar el efecto que tiene el tiempo medio de transferencia en los intercambiadores sobre su nivel de servicio. Este tiempo medio recoge distancias de transferencia y tiempos de espera (por tanto efecto de las frecuencias del servicio). En este ejemplo hemos considerado el tiempo medio de transferencia en el intercambiador 12. Hemos supuesto que los aparcamientos 12, 13, 7 y 8 están operativos. Este tiempo es la media entre el coste de los arcos $(5, 10)$ y $(10, 5)$. Hemos cambiado estos costes y hemos evaluado la partición modal en toda la red de transporte. Las soluciones obtenidas se muestran en la figura 1.7.

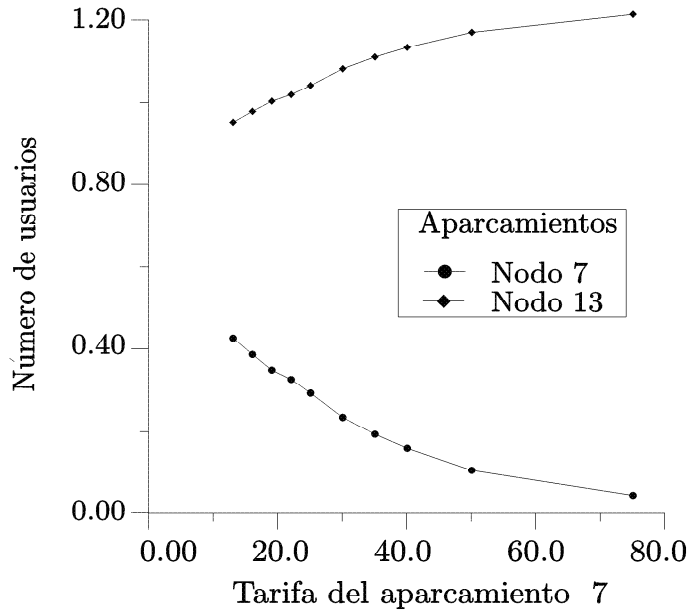


Figura 1.6: Demanda de aparcamiento frente tarifas de aparcamiento

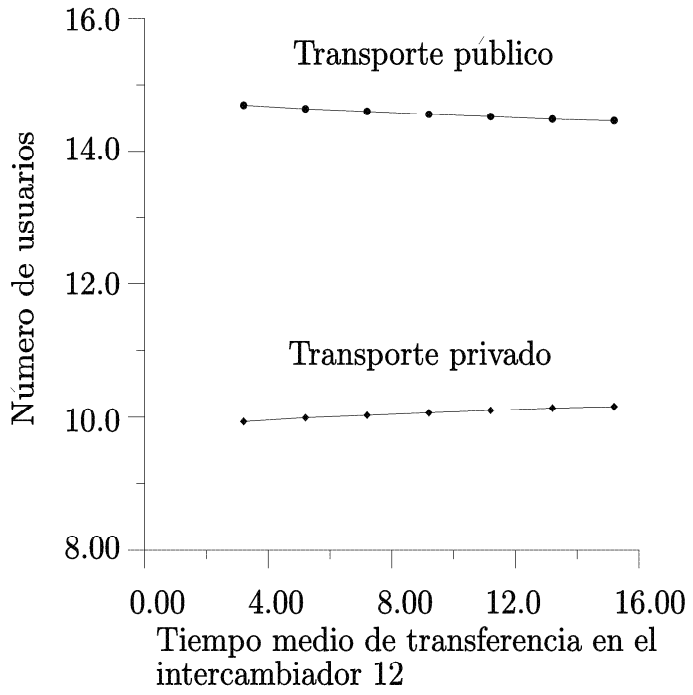


Figura 1.7: Partición modal frente al tiempo de transferencia del intercambiador 12

Apéndice I: resolución del CGPVIP$_{SD}(\ell)$

La tabla 1.11 muestra el algoritmo para resolver el subproblema $CGPVIP_{SD}(\ell)$.

Tabla 1.11: Resolución del $CGPVIP_{SD}(\ell)$

1.0. (*Inicialización*): Sea $\mathbf{h}^{\ell-1} \in \Omega$, calcular $\mathbf{f}^{\ell-1} = \delta^{\mathbf{f}}\mathbf{h}^{\ell-1}$, $\mathbf{g}^{\ell-1} = \delta^{\mathbf{g}}\mathbf{h}^{\ell-1}$, y actualizar $\mathbf{c}(\mathbf{f}^{\ell-1})$.

1.1. (*Cálculo de caminos mínimos en* $\mathcal{G}^a$):

(a) Para cada $\omega \in W$, basándose en el actual coste $\mathbf{c}^a(\mathbf{f}^{\ell-1})$, calcular un camino mínimo p^a_ω. Denotar $U^{a*}_\omega = \bar{C}_{p^a_\omega}(\mathbf{h}^{\ell-1})$.

(b) Para cada $(i,t) \in W^a - W$, basándose en el actual coste $\mathbf{c}^a(\mathbf{f}^{\ell-1})$, calcular el camino de mínimo coste $p^a_{it'}$. Denotar $U^{a*}_{it'} = \theta_a C_{U^a_{it'}}(\mathbf{h}^{\ell-1})$.

1.2. (*Cálculo de hipercaminos de mínimo coste en* $\mathcal{G}^b$): Para cada $\omega \in W^b$, basándose en el actual coste $\mathbf{c}^b(\mathbf{f}^{\ell-1})$, calcular el hipercamino de mínimo coste p^b_ω. Denotar $U^{b*}_\omega = \bar{C}_{p^b_\omega}(\mathbf{h}^{\ell-1})$.

1.3. (*Cálculo del hipercamino combinado de mínimo coste en* $\mathcal{G}$): Para cada $\omega \in W$, calcular

$$U^{c*}_{\omega,t'} = \frac{1}{\gamma_\omega} U^{a*}_{it'} + U^{b*}_{t'j}, \quad \forall t' \in T_\omega.$$

1.4. Para todo $\omega \in W$ calcular los costes extendidos

$$\lambda^k_\omega = U^{k*}_\omega + \frac{\ln\left(g^k_\omega\right)^{\ell-1} + \alpha^k}{\beta_1} \quad k \in \{a,b,d\},$$

$$\lambda^c_\omega = \min_{t' \in T_\omega} \left\{ U^{c*}_{\omega,t'} + \frac{\left(\ln\left(g^c_{\omega,t'}\right)^{\ell-1} + \alpha^c_{t'}\right)}{\beta_2} \right\} + \frac{\ln\left(g^c_\omega\right)^{\ell-1} + \alpha^c}{\beta_1} - \frac{\ln\left(g^c_\omega\right)^{\ell-1}}{\beta_2}. \tag{1.45}$$

Denotar por t^* el nodo de transferencia que minimice la expresión (1.45). Denotar por p^c_ω el hipercamino combinado de mínimo coste en la red multimodal para el par ω a través del nodo t^*.

1.5. Calcular la solución $\bar{\mathbf{h}}^\ell$ como

$$\begin{aligned} k_\omega &= \arg\min_{k \in \{a,b,c,d\}} \left\{\lambda^k_\omega\right\}, \quad \forall \omega \in W, \\ \bar{h}^\ell_{p^{k_\omega}_\omega} &= \bar{g}_\omega, \quad \forall \omega \in W. \end{aligned}$$

y poner el resto de componentes a cero.

Apéndice II: resolución del subproblema CGPVIP$_E(\ell)$

En el algoritmo de Evans se linealiza exclusivamente los términos involucrados en el coste de los arcos, mientras que los términos asociados con la demanda se consideran explícitamente.

La solución del subproblema $CGPVIP_E$ se basa en el hecho de que este problema es un caso particular del problema TAP-MVIP($\bar{\mathbf{C}} - \Lambda, \Omega$), donde el coste de los hipercaminos son constantes. Esto implica que este modelo debe satisfacer las condiciones de equilibrio derivadas de emplear los costes $\bar{C}_p(\mathbf{h}) = \bar{C}_p(\mathbf{h}^{\ell-1})$ para todo p. Hay que destacar dos observaciones:

- ◇ Los costes de los hipercaminos $\bar{C}_p$ son independientes del nivel de servicio. Por esta razón los caminos óptimos son independientes de las variables de demanda.
- ◇ Los valores de las variables de demanda deben satisfacer el modelo de demanda logit anidado. Esto permite resolver el cálculo de estas variables. Estas demandas serán asignadas a los hipercaminos encontrados de mínimo coste.

El siguiente algoritmo resuelve el subproblema CGPVIP$_E(\ell)$.

Tabla 1.12: Resolución del subproblema $CGPVIP_E(\ell)$

1.0.-1.3. Los mismos pasos que en el algoritmo para resolver el CGPVIP$_{SD}(\ell)$. Ver tabla 1.11.

1.4. (*Cálculo de las variables de demanda*): Encontrar un nuevo conjunto de variables $\mathbf{g}_\omega$ para cada $\omega \in W$ usando las formulas (1.1), (1.2),(1.3),(1.4) y (1.5); donde los costes óptimos $\mathbf{U}_\omega$ son obtenidos en las etapas anteriores.

1.5. (*Asignación*): Para todo $\omega \in W$ hacer

$$\begin{aligned} \bar{h}^{\ell}_{p^k_\omega} &= g^k_\omega \quad k \in \{a,b,c,d\}, \\ \bar{h}^{\ell}_{p^c_{\omega,t}} &= g^c_{\omega,t} \quad t \in T_\omega. \end{aligned}$$

Apéndice III: modelo de optimización para costes simétricos

En este modelo la red de transporte público es equivalente a la red de tráfico y los costes en los arcos tienen una matriz Jacobiana simétrica. Las condiciones de equilibrio de este modelo son expresadas como solución del siguiente problema de optimización convexa diferenciable bajo la hipótesis de

$$\beta_1,\ \beta_2 > 0,\ \text{y}\ \frac{\beta_2 - \beta_1}{\beta_1 \beta_2} > 0. \tag{1.46}$$

[TAP–M]

$$\begin{aligned}
\min Z(\mathbf{f},\mathbf{g}) &= \theta_a \sum_{l\in A} \int_0^{f_l} c_l(x)dx + \theta_b \sum_{l\in B} \int_0^{f_l} c_l(x)dx + \sum_{\omega\in W} U_\omega^{d*} g_\omega^d \\
&+ (1/\beta_1) \sum_{k\in\{a,b,c,d\}} \sum_{\omega\in W} g_\omega^k (\ln g_\omega^k - 1 + \alpha^k) - (1/\beta_2) \sum_{\omega\in W} g_\omega^c (\ln g_\omega^c - 1) \\
&+ (1/\beta_2) \sum_{\omega\in W} \sum_{t\in T_\omega} g_{\omega,t}^c (\ln g_{\omega,t}^c - 1 + \alpha_t^c),
\end{aligned} \tag{1.47}$$

sujeto a:

$$\sum_{k\in\{a,b,c,d\}} g_\omega^k = \bar{g}_\omega, \quad \forall \omega \in W, \tag{1.48}$$

$$g_\omega^k = \sum_{p\in P_\omega^k} h_p, \quad \forall k \in \{a,b,d\}, \quad \forall \omega \in W, \tag{1.49}$$

$$g_\omega^c = \sum_{t\in T_\omega} g_{\omega,t}^c, \quad \forall \omega \in W, \tag{1.50}$$

$$g_{\omega,t}^c = \sum_{p\in P_{\omega,t}^c} h_p, \quad \forall t \in T_\omega,\ \forall \omega \in W; \tag{1.51}$$

$$f_l = \sum_{\omega\in W} \frac{1}{\gamma_\omega} \left(\sum_{t\in T_\omega} \sum_{p\in P_{it}^a} \delta_{lp} h_p + \sum_{p\in P_\omega^a} \delta_{lp} h_p \right), \quad \forall l \in A, \tag{1.52}$$

$$f_l = \sum_{\omega\in W} \left(\sum_{t\in T_\omega} \sum_{p\in P_{tj}^b} \delta_{lp} h_p + \sum_{p\in P_\omega^b} \delta_{lp} h_p \right), \quad \forall l \in B, \tag{1.53}$$

$$h_p \geq 0, \quad \forall p \in P_\omega^a \cup P_\omega^b \cup P_{it}^a \cup P_{tj}^b; \tag{1.54}$$

$$\delta_{lp} = \begin{cases} 1, & \text{si } p \text{ emplea el arco } l; \\ 0, & \text{en otro caso.} \end{cases} \qquad \forall l \in A \cup B,\ \forall p \in \{P_\omega^a, P_\omega^b, P_{it}^a, P_{tj}^b\}. \tag{1.55}$$

Abreviadamente lo denotaremos

$$\begin{aligned} &\text{minimizar} && Z(\mathbf{f},\mathbf{g}) = S(\mathbf{f}) + R(\mathbf{g}) \\ &\text{sujeto a} && (\mathbf{f},\mathbf{g}) \in \tilde{\Omega}, \end{aligned} \qquad \text{[TAP-M]}$$

donde $S(\mathbf{f})$ es el coste asociado con el flujo en los arcos y $R(\mathbf{g})$ es el coste asociado con la demanda y denotamos por $\tilde{\Omega}$ la región factible definida por las restricciones (1.48)-(1.55).

Ahora probaremos que la solución del anterior modelo de optimización satisface las condiciones de equilibrio. Bajo la hipótesis (1.46) el TAP-M es un problema convexo y las condiciones de optimalidad se formulan: encontrar $(\mathbf{f}^*,\mathbf{g}^*) \in \tilde{\Omega}$ tal que

$$\nabla S(\mathbf{f}^*)^T (\mathbf{f} - \mathbf{f}^*) + \nabla R(\mathbf{g}^*)^T (\mathbf{g} - \mathbf{g}^*) \geq 0, \quad \forall (\mathbf{f},\mathbf{g}) \in \tilde{\Omega}. \tag{1.56}$$

Consideremos el flujo en los hipercaminos $\mathbf{h}^* \in \Omega$ cumpliendo $\mathbf{f}^* = \delta^{\mathbf{f}}\mathbf{h}^*$ y $\mathbf{g}^* = \delta^{\mathbf{g}}\mathbf{h}^*$. Daremos la formulación del anterior problema de desigualdades variacionales en términos de las variables h_p. La función objetivo está constituida por

$$\begin{aligned}
\nabla S(\mathbf{f}^*)^T(\mathbf{f}-\mathbf{f}^*) &= \theta_a \sum_{l\in A} c_l(f_l^*)(f_l - f_l^*) + \theta_b \sum_{l\in B} c_l(f_l^*)(f_l - f_l^*), \\
\nabla R(\mathbf{g}^*)^T(\mathbf{g}-\mathbf{g}^*) &= \sum_{\omega\in W} U_\omega^{d*}(g_\omega^d - g_\omega^{d*}) + (1/\beta_1)\sum_{k\in\{a,b,c,d\}}\sum_{\omega\in W}(\ln g_\omega^{k*} + \alpha^k)(g_\omega^k - g_\omega^{k*}) \qquad (1.57)\\
&- (1/\beta_2)\sum_{\omega\in W}(\ln g_\omega^{c*})(g_\omega^c - g_\omega^{c*}) + (1/\beta_2)\sum_{\omega\in W}\sum_{t\in T_\omega}(\ln g_{\omega,t}^{c*} + \alpha_t^c)(g_{\omega,t}^c - g_{\omega,t}^{c*}).
\end{aligned}$$

Empleando la restricción (1.52) podemos expresar el término del coste en los arcos, el cual depende del flujo en los arcos, en función del flujo en los hipercaminos h_p y h_p^*.

$$\begin{aligned}
\theta_a \sum_{l\in A} c_l(f_l^*)(f_l - f_l^*) &= \theta_a \sum_{l\in A} c_l(f_l^*)\left[\sum_{\omega\in W}\frac{1}{\gamma_\omega}\left(\sum_{t\in T_\omega}\sum_{p\in P_{it}^a}\delta_{lp}(h_p - h_p^*) + \sum_{p\in P_\omega^a}\delta_{lp}(h_p - h_p^*)\right)\right] \\
&= \sum_{\omega\in W}\left[\sum_{t\in T_\omega}\sum_{p\in P_{it}^a}\left[\sum_{l\in A}\frac{\theta_a c_l(f_l^*)}{\gamma_\omega}\delta_{lp}\right](h_p - h_p^*) + \sum_{p\in P_\omega^a}\left(\sum_{l\in A}\frac{\theta_a c_l(f_l^*)}{\gamma_\omega}\delta_{lp}\right)(h_p - h_p^*)\right]. \qquad (1.58)
\end{aligned}$$

Empleando (1.11) y (1.12) podemos escribir la ecuación (1.58) como

$$\theta_a \sum_{l\in A} c_l(f_l^*)(f_l - f_l^*) = \sum_{\omega\in W}\left[\sum_{p\in P_{it}^a}\sum_{t\in T_\omega}\bar{C}_p^*(h_p - h_p^*) + \sum_{p\in P_\omega^a}\bar{C}_p^*(h_p - h_p^*)\right].$$

donde $\bar{C}_p^* = \bar{C}_p(\mathbf{h}^*)$. Análogamente obtenemos la siguiente expresión para los arcos en la red pública

$$\theta_b \sum_{l\in B} c_l(f_l^*)(f_l - f_l^*) = \sum_{\omega\in W}\left[\sum_{p\in P_\omega^b}\bar{C}_p^*(h_p - h_p^*) + \sum_{t\in T_\omega}\sum_{p\in P_{tj}^b}\bar{C}_p^*(h_p - h_p^*)\right].$$

Para usar una notación abreviada consideramos que el modo (d) tiene un único camino óptimo para cada demanda, esto es, $P_\omega^d = \{h_p\}$, y $g_\omega^d = h_p$. El coste del camino $p \in P_\omega^d$ lo denotamos por U_ω^{d*} y es independiente del flujo del camino. Para unificar la notación tomamos $\bar{C}_p^* = U_\omega^{d*}$ con $p \in P_\omega^d$.

Ahora expresamos el término $\nabla R(\mathbf{g}^*)^T(\mathbf{g}-\mathbf{g}^*)$ en función de las variables h_p. Empleando la ecuación (1.49)

$$\begin{aligned}
(1/\beta_1)\sum_{\omega}\sum_{k\in\{a,b,c\}}(\ln g_\omega^{k*} + \alpha^k)(g_\omega^k - g_\omega^{k*}) &= \sum_{\omega}\sum_{k\in\{a,b,c\}}\sum_{p\in P_\omega^k}\frac{(\ln g_\omega^{k*} + \alpha^k)}{\beta_1}(h_p - h_p^*), \\
(-1/\beta_2)\sum_{\omega}(\ln g_\omega^{c*})(g_\omega^c - g_\omega^{c*}) &= \sum_{\omega}\sum_{p\in P_\omega^c} -\frac{(\ln g_\omega^{c*})}{\beta_2}(h_p - h_p^*).
\end{aligned}$$

Usando la expresión (1.51) podemos reemplazar las variables $g_{\omega,t}^c$ $g_{\omega,t}^{c*}$ de la función objetivo por las variables h_p y h_p^*.

$$\sum_{\omega}\sum_{t\in T_\omega}\frac{(\ln g_{\omega,t}^{c*} + \alpha_t^c)}{\beta_2}(g_{\omega,t}^c - g_{\omega,t}^{c*}) = \sum_{\omega}\sum_{t\in T_\omega}\sum_{p\in P_{\omega,t}^c}\frac{(\ln g_{\omega,t}^{c*} + \alpha_t^c)}{\beta_2}(h_p - h_p^*).$$

Sustituyendo las anteriores expresiones en (1.56) obtenemos:

$$\sum_{\omega}\left[\sum_{k\in\{a,b,d\}}\sum_{p\in P_{\omega}^{k}}\left(\bar{C}_p+\frac{\ln g_{\omega}^{k*}+\alpha^{k}}{\beta_1}\right)(h_p-h_p^*)\right.$$
$$\left.+\sum_{t\in T_{\omega}}\sum_{p\in P_{\omega,t}^{c}}\left(\bar{C}_p+\frac{\ln g_{\omega}^{c*}+\alpha^{c}}{\beta_1}-\frac{\ln g_{\omega}^{c*}}{\beta_2}+\frac{\ln g_{\omega,t}^{c*}+\alpha_t^{c}}{\beta_2}\right)(h_p-h_p^*)\right]\geq 0,\forall\mathbf{h}\in\Omega,$$

$\mathbf{h}^*$ es una solución de TAP-MVIP($\bar{\mathbf{C}}-\Lambda,\Omega$), y por tanto es un flujo en equilibrio.

Alternativamente al método aquí seguido, Fernández y otros [73] emplean las condiciones de optimalidad de Karush-Khun-Tucker para demostrar que la solución del TAP-M satisface las condiciones de equilibrio **C1**, **C2** y **C3**.

Capítulo 2

La clase de algoritmos CG/SD en optimización convexa diferenciable: análisis de la convergencia

Resumen

En este capítulo se introduce una clase de algoritmos de generación de columnas (CG)/ descomposición simplicial (SD) para problemas de optimización no lineales, convexos y diferenciables. Este desarrollo estuvo motivado por los resultados numéricos de la descomposición simplicial no lineal (NSD), obtenidos en Larsson y otros [154, 141], aplicados a problemas no lineales de flujos en redes de grandes dimensiones. La clase NSD se construyó sobre la idea intuitiva de que la generación de columnas no lineales tiene ventajas computacionales. La clase aquí desarrollada es más general y permite la generación de columnas obtenidas como una solución truncada del problema original. Esto permite obtener columnas de gran calidad que conducen a una reducción sustancial de los tamaños de los problemas maestros restringidos, en comparación con el clásico RSD (Hearn y otros [123, 125]). La convergencia global de esta clase se demuestra para problemas de minimización de una función (pseudo-)convexa sobre un conjunto compacto y convexo. Propiedades sobre la convergencia finita de los algoritmos CG/SD son también analizadas. El estudio teórico desarrollado en este capítulo se completará con un estudio numérico sobre problemas de flujos en redes uniproducto y multiproducto, que se realizará en el siguiente capítulo, mostrando una comparativa entre estos nuevos algoritmos y los algoritmos RSD y NSD.

Palabras clave:

Descomposición simplicial, generación y eliminación de columnas, aproximación interior, símplices, cara óptima, no degeneración, minimización pseudoconvexa, mínimo débilmente puntiagudo, convergencia finita.

2.1 Introducción y motivación

El algoritmo de *descomposición simplicial* (SD), tuvo su origen en los trabajos de Holloway [128] y Von Hohenbalken [127], lo que constituye una metodología para la construcción de algoritmos para problemas no lineales de grandes dimensiones. Hay dos características principales en esta clase de métodos:

(i) Estos métodos construyen y resuelven una aproximación al problema original, obtenida reemplazando la región factible por un conjunto poliedral que es una *aproximación interior* (un subconjunto) de dicha región. Esta aproximación se le conoce con el nombre de *problema maestro restringido* (RMP).

(ii) Esta aproximación interior es mejorada (aumentada) mediante la generación de un vector (o columna) en el conjunto factible, a través de la resolución de otra aproximación al problema de optimización original (el *subproblema de generación de columnas*, CGP) donde la función objetivo es aproximada (a menudo por una función *lineal*).

Por tanto, la clase de métodos de descomposición simplicial puede ser situada en el marco de métodos de *generación de columnas*. Como la sucesión de valores de la función objetivo de los problemas maestros restringidos es decreciente, este marco también pertenece a los algoritmos *iterativos factibles de descenso*.

Los problemas para los cuales los métodos SD han sido aplicados satisfactoriamente son los que poseen restricciones con estructura especial, como son los problemas con restricciones de red. (Hearn y otros [125], Mulvey y otros [176]) o estructura de producto cartesiano (Larsson y Patriksson [140]). Estas propiedades se explotan para resolver eficientemente los problemas lineales de generación de columnas mediante el empleo de métodos especializados. También ha sido constatado que estos subproblemas lineales son la causa de una convergencia lenta cerca de la solución óptima, debido a un deterioro creciente en la mejora de la contribución de cada columna. Esto fue experimentado en los trabajos sobre el RSD de Hearn y otros [123, 125] y ha conducido al desarrollo de métodos SD con generación de columnas no lineales en Larsson y otros [154, 141] (ver también Patriksson [197]). Los algoritmos desarrollados en esos trabajos estaban basados en la sustitución del subproblema lineal por un problema estrictamente convexo tal como los emplean los algoritmos de Goldstein–Levitin–Polyak Goldstein [112], Levitin y Polyak [150] o del método de Newton. Los subproblemas convexos no requieren ser resueltos exactamente para asegurar la convergencia de los métodos. Estos algoritmos truncados, dentro del contexto de algoritmos CG/SD, son convergentes y mejoran (a veces sustancialmente) a los esquemas clásicos.

Los experimentos numéricos realizados en redes multiproducto de mediano y gran tamaño han demostrado que estos subproblemas estrictamente convexos, conducen a generación de columnas de gran calidad, en el sentido de que conducen a grandes mejoras. Consecuentemente reducen, comparado con el esquema clásico del algoritmo SD, el tamaño y el número necesario de los RMP para alcanzar una determinada precisión.

A continuación presentamos el problema en estudio y discutimos algunos detalles de la clase de algoritmos CG/SD.

2.1.1 Métodos de generación de columnas / descomposición simplicial

Consideramos el siguiente problema restringido de optimización convexa diferenciable:

$$\begin{aligned} \text{minimizar} \quad & f(\mathbf{x}), \\ \text{sujeto a} \quad & \mathbf{x} \in X, \end{aligned} \qquad [\text{CDP}(f, X)]$$

donde $X \subset \Re^n$ es no vacío, convexo y compacto, y $f : X \mapsto \Re$ es continuamente diferenciable y pseudoconvexa en X. Bajo estas suposiciones, el conjunto de soluciones óptimas, $\text{SOL}(f, X)$, es no

vacío, convexo, compacto y caracterizado por la siguiente desigualdad variacional

$$-\nabla f(\mathbf{x}^*) \in N_X(\mathbf{x}^*), \qquad [\mathrm{VIP}(\nabla f, X)]$$

donde

$$N_X(\mathbf{x}) := \begin{cases} \{\, \mathbf{z} \in \Re^n \mid \mathbf{z}^{\mathrm{T}}(\mathbf{y} - \mathbf{x}) \leq 0, \quad \forall \mathbf{y} \in X \,\}, & \mathbf{x} \in X, \\ \emptyset, & \mathbf{x} \notin X \end{cases} \tag{2.1}$$

denota el cono normal de X en $\mathbf{x} \in \Re^n$.

La obtención del método SD se basa en los dos resultados clásicos de representación de conjuntos convexos y de puntos en tales conjuntos. El primer resultado es el *teorema de representación* (ver por ejemplo, Lasdon [143], Bazaraa y otros [12]), que establece que un vector $\mathbf{x} \in \Re^n$ pertenece a un conjunto *poliedral* X si y sólo si se puede representar como combinación convexa de sus puntos extremos ($\mathbf{p}_i$, $i \in \mathcal{P}$) más una combinación no negativa de sus direcciones extremas ($\mathbf{d}_j$, $j \in \mathcal{D}$), esto es, para algún par de vectores λ y μ,

$$\mathbf{x} = \sum_{i \in \mathcal{P}} \lambda_i \mathbf{p}_i + \sum_{j \in \mathcal{D}} \mu_j \mathbf{d}_j, \tag{2.2}$$

$$\sum_{i \in \mathcal{P}} \lambda_i = 1, \tag{2.3}$$

$$\lambda_i,\, \mu_j \geq 0, \qquad i \in \mathcal{P}, \quad j \in \mathcal{D}. \tag{2.4}$$

Entonces, en principio, el conjunto poliédrico X puede ser representado internamente en función de sus puntos y direcciones extremas, y el problema $\mathrm{CDP}(f, X)$ puede ser formulado en las variables λ_i y μ_j en lugar de $\mathbf{x}$. La ventaja de hacer esta transformación es que la representación interna de X es más simple que la representación original usando igualdades y desigualdades lineales. Descartando las restricciones operacionales (2.2), el conjunto descrito por (2.3) y (2.4) es el producto cartesiano de un *símplice* y el octante no negativo. Un problema de optimización sobre dicho conjunto puede ser resuelto con casi el mismo esfuerzo que el empleado en un problema no restringido (ver por ejemplo, Bertsekas y otros [17]).

La desventaja de esta transformación es que el número de puntos y direcciones extremas crecen exponencialmente con la dimensión del problema, y esto introduce un número impracticable de variables. La práctica de la descomposición simplicial recae sobre el segundo resultado básico de representación de conjuntos convexos, el *teorema de Carathéodory* (ver por ejemplo, teorema 17.1 de Rockafellar [204]). Este resultado establece que un punto $\mathbf{x}$ en la envoltura convexa de cualquier subconjunto X de $\Re^n$ puede ser representado como una combinación convexa de a lo sumo tantos elementos de X como su dimensión, $\mathtt{dim}\, X$ (que es definido como la dimensión de su envoltura afin), más uno. El teorema de Carathéodory no se formula en término de direcciones y puntos extremos; y es por tanto aplicable a conjuntos convexos generales. En el contexto de la descomposición simplicial (clásica) se aplica este resultado a conjuntos poliédricos acotados, obteniendo que cualquier punto puede ser descrito por una combinación convexa de puntos extremos. El número total de estos puntos extremos no puede exceder de la dimensión del poliedro más uno. Este resultado obviamente refina el teorema de representación.

La forma clásica del método SD fue primeramente descrita por Von Hohenbalken [127]. El algoritmo alterna entre la solución de dos subproblemas. Conocida un subconjunto de puntos extremos $\hat{\mathcal{P}}$ y de direcciones extremas $\hat{\mathcal{D}}$ de $\mathcal{P}$ y $\mathcal{D}$ respectivamente, f se minimiza sobre la aproximación interna de X definida reemplazando los conjuntos $\mathcal{P}$ y $\mathcal{D}$ en (2.2), en término de las variables $\hat{\lambda}_i$, $i \in \hat{\mathcal{P}}$ y $\hat{\mu}_j$, $j \in \hat{\mathcal{D}}$. [Este es el denominado *problema maestro restringido* (*RMP*); en algunas ocasiones a este problema se le denomina *paso de coordinación*.]. Notar que empleamos la notación $\hat{\lambda}$ y $\hat{\mu}$ para distinguir los vectores en el RMP de los vectores (de mayores dimensiones) λ y μ en el *problema maestro completo*, el cual es equivalente al $\mathrm{CDP}(f, X)$ y está definido por el sistema (2.2)–(2.4). Denotando por $\hat{\Lambda}$ el conjunto de vectores $(\hat{\lambda}, \hat{\mu})$ cumpliendo las restricciones del sistema (2.3)–(2.4) para los subconjuntos conocidos $\hat{\mathcal{P}}$ y $\hat{\mathcal{D}}$ y empleando (2.3) para sustituir $\mathbf{x}$ en función de $(\hat{\lambda}, \hat{\mu})$ [escribimos $\mathbf{x} = \mathbf{x}(\hat{\lambda}, \hat{\mu})$]. El

RMP puede entonces ser formulado por

$$\begin{array}{ll} \text{minimizar} & f(\mathbf{x}(\hat{\lambda},\hat{\mu})) \\ \text{sujeto a} & (\hat{\lambda},\hat{\mu}) \in \hat{\Lambda} \end{array} \qquad [\mathrm{RMP}(f,\hat{\Lambda})]$$

Alternativamente, un punto o dirección extrema de X es generado a través de la solución de una aproximación a $\mathrm{CDP}(f,X)$, en la que la función f es reemplazada por su aproximación lineal, $\mathbf{y} \mapsto f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x})$, definida en la solución, $\mathbf{x}$, de $\mathrm{RMP}(f,\hat{\Lambda})$, esto es, por el problema

$$\begin{array}{ll} \text{minimizar} & \nabla f(\mathbf{x})^T\mathbf{y} \\ \text{sujeto a} & \mathbf{y} \in X \end{array} \qquad (2.5)$$

Este es el llamado *subproblema de generación de columnas, CGP*, y corresponde al paso de *descomposición* en la descripción de algunos métodos de generación de columnas. Si la solución a este problema pertenece a la actual aproximación interior, la actual solución $\mathbf{x}$, es óptima para el problema $\mathrm{CDP}(f,X)$, entonces $-\nabla f(\mathbf{x})$ es un elemento de $N_X(\mathbf{x})$. En caso contrario, $\hat{\mathcal{P}}$ o $\hat{\mathcal{D}}$ es aumentado por un nuevo elemento y la aproximación interior resultante conduce a un nuevo RMP que tiene un valor óptimo estrictamente menor que el anterior; el último resultado es derivado del hecho de que $\nabla f(\mathbf{x})^T\mathbf{d} < 0$ (esto es, $\mathbf{d}$ define una dirección de descenso con respecto a f en $\mathbf{x}$), donde $\mathbf{d}$ denota la dirección $\mathbf{d} := \mathbf{y} - \mathbf{x}$ hacia el nuevo punto extremo $\mathbf{y}$, o una nueva dirección extrema. Este proceso de resolver un CGP y un RMP es repetido iterativamente.

En el método de Von Hohenbalken [127], el teorema de Carathéodory es empleado en la validación de una regla de *eliminación de columnas*, de acuerdo a la cual, cualquier punto o dirección extrema cuyo peso sea cero en la expresión de la solución $\mathbf{x}$ del RMP, será eliminado/a; gracias a la finitud de los conjuntos $\mathcal{P}$ y $\mathcal{D}$ y al hecho de que la sucesión de valores óptimos de la función objetivo del RMP es estrictamente decreciente, la convergencia del algoritmo SD a una solución óptima se realiza en un número finito de iteraciones.

La denominada descomposición simplicial restringida (RSD) fue desarrollada por Hearn y otros [123, 125]. Este algoritmo constituye una mejora del SD para los problemas donde el conjunto X es un poliedro acotado. La base para tal mejora está en la observación de que cualquier solución factible, por ejemplo las soluciones óptimas, debe poderse representar como una combinación convexa de a lo sumo $\dim X + 1$ puntos extremos, como afirma el teorema de Carathéodory. Realmente, la máxima dimensión de puntos extremos necesarios para describir la solución óptima $\mathbf{x}^*$ es $\dim F^* + 1$, donde F^* es la *cara óptima* de X. En el contexto de minimización convexa diferenciable, este conjunto está descrito por

$$F^* := \{\, \mathbf{y} \in X \mid \nabla f(\mathbf{x}^*)^T(\mathbf{y}-\mathbf{x}^*) = 0 \,\}.$$

Este conjunto es la envoltura convexa de los puntos extremos de X que resuelven el problema de aproximación lineal (2.5) de $\mathrm{CDP}(f,X)$, basándose en esta observación, Hearn y otros introdujeron un parámetro r en el esquema original SD para limitar el número de puntos extremos almacenados en el RMP. Cuando esta cantidad es alcanzada, cualquier nuevo punto extremo generado reemplaza la columna de $\hat{\mathcal{P}}$ que posee un menor peso en la solución del RMP. Para asegurar la convergencia del algoritmo, la solución del RMP debe ser retenida como una columna individual (no obstante, no es contada entre las r columnas activas).

El valor de r es crucial para la eficiencia del algoritmo. Si $r \geq \dim F^* + 1$, entonces el número de los RMP a resolver es finito, y la velocidad de convergencia es gobernada por la velocidad de convergencia del método empleado para resolver el RMP; entonces, el algoritmo RSD posee una convergencia superlineal si se emplea (por ejemplo) el método de Newton proyectado (Hearn y otros [125]). Si $r < \dim F^* + 1$, entonces el RSD posee sólo convergencia asintótica, y la velocidad de convergencia es la misma que el algoritmo de *Frank–Wolfe* (que es obtenido como caso especial del RSD cuando $r = 1$), esto es, la tasa de convergencia es sublineal. La dimensión de F^* no puede ser estimado de los datos del problema y es desconocido a priori, el valor adecuado de r debe ser basado en la experiencia computacional.

El algoritmo RSD ha sido satisfactoriamente aplicado a problemas no lineales de grandes dimensiones con estructura especial, en particular, en modelos de programación matemática de flujos no lineales en redes, donde el subproblema de generación de columnas se reduce a un problema de flujos lineales en redes que pueden ser eficientemente resueltos (ver por ejemplo, Hearn y otros [125], Mulvey [176], Larsson y Patriksson [140]). La experiencia con el método RSD ha mostrado que hace rápidos progresos inicialmente, y rápidamente alcanza una solución casi óptima, especialmente cuando es empleado un gran valor de r y cuando un método de segundo orden es utilizado en la solución de RMP, pero esta velocidad se reduce cerca de la solución óptima. El algoritmo RSD es menos eficiente para problemas que poseen grandes valores de la $\mathtt{dim}\, F^*$, que hace que el valor de r, el número y el tamaño de los RMP lleguen a ser excesivamente grandes.

La explicación de esta conducta se encuentra en la construcción del problema de generación de columnas que emplea una aproximación de primer orden de f. El subproblema de generación de columnas en el RSD es el mismo que en el algoritmo de Frank–Wolfe mencionado anteriormente. Es sabido que la calidad de éstas direcciones de búsqueda se deteriora rápidamente. La razón es que la sucesión de derivadas direccionales $\{\nabla f(\mathbf{x}^t)^T\mathbf{d}^t\}$ en las direcciones de búsqueda $\mathbf{d}^t := \mathbf{y}^t - \mathbf{x}^t$ tienden a cero pero la sucesión $\{\mathbf{d}^t\}$ no converge a $\mathbf{0}$; lo que implica que las direcciones de búsqueda tienden rápidamente a ser ortogonales al gradiente de f y por tanto la calidad de las columnas generadas (entendida como la mejora inducida en el RMP) será rápidamente deteriorada. Una conclusión natural es emplear en la fase de generación de columnas del método SD una mejor aproximación de f, de ello se podría esperar una mejor calidad en las columnas y por tanto una mejor aproximación interior en el RMP. Larsson y otros [154, 141], basándose en esta observación, extendieron el algoritmo RSD a métodos de generación de columnas no lineales. Esta clase se denomina *descomposición simplical no lineal* (*NSD*). Estos métodos poseen menor sensibilidad a la dimensión de la cara óptima debido a que emplean un menor número de columnas para describir la solución óptima. Esto hace que los métodos requieran un menor número de iteraciones y permite elegir un valor del parámetro r menor.

El método NSD se obtiene reemplazando en el RSD el subproblema de generación de columnas lineal (2.5) por el más general

$$\begin{aligned} \text{minimizar} \quad & \nabla f(\mathbf{x})^T\mathbf{y} + \varphi(\mathbf{y},\mathbf{x}), \\ \text{sujeto a} \quad & \mathbf{y} \in X \end{aligned} \qquad [\mathrm{CDP}(\varphi(\cdot,\mathbf{x}), \nabla f, X, \mathbf{x})]$$

donde $\varphi : X \times X \mapsto \Re$ es una función continua de la forma $\varphi(\mathbf{y},\mathbf{x})$; y para todo $\mathbf{x} \in X$ es convexa y continuamente diferenciable respecto a la variable $\mathbf{y}$. Además, se asume que satisface las propiedades de $\varphi(\mathbf{x},\mathbf{x}) = 0$ y $\nabla_{\mathbf{y}}\varphi(\mathbf{x},\mathbf{x}) = \mathbf{0}$ para todo $\mathbf{x} \in X$.

Entre las posibles elecciones para φ mencionamos las siguientes, donde $\mathbf{x}^t$ denota el punto en la iteración t, `diag` denota la parte diagonal de una matriz y donde $\gamma > 0$:

$\varphi(\mathbf{y},\mathbf{x}^t)$	Subproblema
0	Frank–Wolfe
$(1/2)(\mathbf{y}-\mathbf{x}^t)^T\nabla^2 f(\mathbf{x}^t)(\mathbf{y}-\mathbf{x}^t)$	Newton
$(1/2)(\mathbf{y}-\mathbf{x}^t)^T[\mathtt{diag}\,\nabla^2 f(\mathbf{x}^t)](\mathbf{y}-\mathbf{x}^t)$	Diag. Newton
$\gamma/2\|\mathbf{y}-\mathbf{x}^t\|^2$	Proyección

Supongamos que en la iteración t el algorimos ha generado la columna $\hat{\mathbf{y}}^t$. El método NSD no almacena $\hat{\mathbf{y}}^t$ sino su extensión a la frontera (relativa) de X, esto es,

$$\mathbf{y}^t := \mathbf{x}^t + \ell_t(\hat{\mathbf{y}}^t - \mathbf{x}^t), \qquad \text{donde} \qquad \ell_t := \max\{\, \ell \mid \mathbf{x}^t + \ell(\hat{\mathbf{y}}^t - \mathbf{x}^t) \in X \,\}. \tag{2.6}$$

La motivación de esta operación es aumentar tanto como sea posible la aproximación interior $X^t \subset X$. Notar que esta operación en los métodos RSD y SD es inútil ya que no se puede prolongar las columnas obtenidas como solución de (2.5), por ser puntos extremos de X. Por otro lado, la solución de $\mathrm{CDP}(\varphi(\cdot,\mathbf{x}), \nabla f, X, \mathbf{x})$ puede ser un punto del interior (relativo) de X y esta operación proyectaría la columna en la frontera relativa.

NOTA 2.1.1 (Extensión de los métodos de direcciones factibles a búsquedas multidimensionales). Varios trabajos anteriores han empleado el subproblema $\mathrm{CDP}(\varphi(\cdot,\mathbf{x}), \nabla f, X, \mathbf{x})$ con métodos de búsquedas

lineales para CDP(f, X) (ver por ejemplo, Tseng [229], Patriksson [193, 194], Migdalas [172], Zhu y Marcotte [250], Patriksson [197]). El algoritmo NSD puede ser interpretado como una generalización de estos métodos de búsquedas unidimensionales a métodos con búsquedas multidimensionales. La clase planteada en este capítulo contiene los subproblemas de tipo CDP$(\varphi(\cdot, \mathbf{x}), \nabla f, X, \mathbf{x})$ y muchas otras alternativas.

Esta interpretación posible proporciona un nueva motivación para la clase aquí descrita. Para problemas altamente no lineales, los métodos de búsquedas unidimensionales pueden llegar a ser ineficientes debido a que consideran longitudes de paso demasiado pequeñas. Varias alternativas han sido desarrolladas para subsanar esta deficiencia tales como búsquedas sobre curvas y esquemas de tipo "*trust region*". El marco propuesto reemplaza la búsqueda unidimensional por búsqueda sobre conjuntos de mayores dimensiones que podrían beneficiarse de la posibilidad de emplear conjuntos no poliedrales. □

Muchas de las ventajas del método RSD recaen sobre las propiedades del conjunto X. Éstas están todavía presentes en los métodos NSD debido a que puede ser explotadas para resolver los problemas CDP$(\varphi(\cdot, \mathbf{x}), \nabla f, X, \mathbf{x})$. La convergencia finita se perderá en general, debido a que son generados puntos no extremos (ver ejemplo 2.4.17). Cabe esperar que sea más rápida la convergencia del método NSD que la del RSD, en función del número de iteraciones y del tiempo requerido, ya que los subproblemas no lineales pueden ser resueltos eficientemente. En los experimentos numéricos de redes no lineales de grandes dimensiones tales conclusiones fueron obtenidas en Larsson y otros [154, 141] quienes observaron que el NSD es relativamente menos sensible a los valores de $\mathtt{dim}\, F^*$ que el RSD, esto permite emplear un menor valor de r en el NSD.

Finalmente remarcaremos que los resultados para la convergencia del NSD permiten que tanto el CGP y RMP puedan ser resueltos aproximadamente. Esto facilita las aplicaciones prácticas de la clase. En Hearn y otros [125], la convergencia se establece para el RSD cuando el RMP se resuelve empleando únicamente una iteración del método de Newton. El esquema de generación de columnas / descomposición simplicial de Larsson y otros [138] es validado para soluciones truncada en ambos subproblemas, empleando el concepto de aplicaciones cerradas. Varios desarrollos sobre estas líneas han sido realizados en el capítulo 9 de Patriksson [197] para el NSD, donde la propiedad de clausura de la aplicación multievaluada definida por el algoritmo no es necesariamente exigida y donde las reglas para la introducción o elimación de columnas son más generales. Estos desarrollos han sido empleados en la construcción de la versión conceptual de la clase de algoritmos CG/SD.

2.1.2 Motivación para una nueva clase de algoritmos de generación de columnas / descomposición simplicial

El algoritmo NSD de Larsson y otros [154, 141] considera solamente una parte de los posibles modos de generar columnas de alta calidad a través de la exploración de problemas de generación de columnas no lineales. Una clase de algoritmos que no ha sido explorada en el marco del NSD es la que obtiene columnas mediante la resolución aproximada del problema *original* aplicando un número limitado de iteraciones de algún algoritmo. En el NSD este número siempre es uno, y el clásico SD efectúa una única iteración del algoritmo de Frank-Wolfe sobre el problema original. Parece razonable que estos esquemas mejorarán los anteriores.

No obstante, una importante motivación para el desarrollo de la presente clase de algoritmos, se encuentra en la experiencia computacional de Larsson y otros [154, 141], en la que se estableció que la resolución de mejores aproximaciones al problema original tenía ventajas computacionales. El limitado rango de métodos para la generación de columnas en Larsson y otros [154, 141] permite mejoras sustanciales para la clase anterior. Algunas cuestiones teóricas importantes no fueron abordadas en Larsson y otros [154, 141], y algunas de ellas son tratadas en este capítulo. Primeramente, ninguna indagación fue realizada sobre las propiedades de la aproximación interior X^t. En ese capítulo demostraremos que, en el caso general de los algoritmos CG/SD, estos conjuntos son símplices bajo las hipótesis naturales de resolución exacta del RMP y bajo las mismas reglas de eliminación de columnas que en los trabajos de (Von Hohenbalken [127], Hearn y otros [123, 125]. En segundo lugar,

Larsson y otros probaron que en el NSD las columnas generadas identifican finitamente la cara óptima, pero nada fue probado respecto a la sucesión $\{\mathbf{x}^t\}$. Probaremos que la aproximación interior X^t en el CG/SD identifica la cara óptima de X en un número finito de iteraciones, por tanto, la sucesión $\{\mathbf{x}^t\}$ también lo hace. Una posible aplicación de este resultado es la construcción de algoritmos que combinen inicialmente un algoritmo CG/SD y finalmente un algoritmo de conjuntos activos donde la búsqueda es reducida al subespacio de restricciones activas. En tercer y último lugar, es sabido que el algoritmo RSD es finitamente convergente bajo ciertas condiciones en su especificación. Larsson y otros [154, 141] no estudiaron esta propiedad para el algoritmo NSD. Demostraremos para el CG/SD, en el caso general de conjuntos convexos y bajo la hipótesis de óptimos débilmente puntiagudos, que posee la propiedad de convergencia finita.

En la próxima sección se desarrolla el marco propuesto y se analiza su convergencia asintótica. El algoritmo es dado en forma conceptual para permitir una gran flexibilidad en sus realizaciones. En la sección 2.3, estudiamos varias formas de definir la aproximación interior del conjunto factible y establecemos las condiciones bajo las cuales este conjunto es un símplice. Desarrollamos condiciones sobre la geometría del problema de optimización y sobre los algoritmos usados en el CGP y en el RMP de modo que las restricciones activas, o incluso las soluciones, puedan ser identificadas en un número finito de iteraciones.

2.2 El algoritmo conceptual CG/SD y su convergencia

2.2.1 El algoritmo CG/SD

Comenzamos estableciendo y validando el esquema conceptual del algoritmo CG/SD, que permite la solución inexacta tanto del RMP como del CGP y además una mayor generalidad en las reglas para la adaptación de la aproximación interior. El algoritmo está descrito por medio de unas aplicaciones (posiblemente punto–conjunto) algorítmicas que permite asumir condiciones similares a las empleadas en el análisis de la convergencia de Zangwill [248].

Supondremos que el problema de generación de columnas se resuelve empleando un procedimiento iterativo, denotado por $\mathcal{A}_{\mathrm{c}}^k$ y perteneciente a una colección finita de algoritmos $\mathcal{K}_{\mathrm{c}}$, de dos tipos posibles. Para la primera alternativa, el algoritmo se aplica una vez en cada iteración principal y proporciona una dirección de descenso. Decimos que este algoritmo es de tipo 1. La segunda alternativa recae sobre la existencia de una función de mérito, $\Pi : X \times X \mapsto \Re^n$, para el problema de generación de columnas. En este caso, el algoritmo es aplicado al menos una vez (posiblemente varias veces) comenzando en el punto $\mathbf{x}$. Asumiremos que cada vez que se aplica el algoritmo se produce una reducción del valor de la función de mérito $\Pi(\cdot, \mathbf{x})$, a no ser se haya obtenido una solución del problema de generación de columnas; decimos que estos algoritmos son de tipo 2.

Supondremos que los problemas maestros restringidos se resuelven empleando una clase finita de algoritmos iterativos con similares propiedades que los algoritmos del tipo 2, pero en lugar de reducir una función de mérito genérica reducen el valor de f. Esta clase se denota $\mathcal{K}_{\mathrm{r}}$. En cada iteración se aplica uno de estos métodos que es denotado por $\mathcal{A}_{\mathrm{r}}^k$ con $k \in \mathcal{K}_{\mathrm{r}}$.

Los algoritmos empleados en el RMP y en el CGP pertenecen a la clase de algoritmos cerrados de descenso:

HIPÓTESIS 2.2.1 (Algoritmos cerrados de descenso). *Sea $\widehat{X} \subseteq X$ un conjunto no vacío, compacto y convexo, y sea $A : \widehat{X} \mapsto 2^{\widehat{X}}$ una aplicación multievaluada sobre $\widehat{X}$. Sea $\Pi : X \times X \mapsto \Re^n$ una función continua en $X \times X$ cumpliendo que $\nabla\Pi(\cdot, \mathbf{x}) = \nabla f(\cdot)$ y la función $\Pi(\cdot, \mathbf{x})$ es pseudoconvexa para todo $\mathbf{x} \in X$. La aplicación A satisface las siguientes tres condiciones:*

(a) (Cerrada). *La aplicación A es cerrada en $\widehat{X}$, esto es, para cada $\mathbf{x} \in \widehat{X}$,*

$$\left.\begin{array}{r} \{\mathbf{x}^t\} \to \mathbf{x} \\ \mathbf{y}^t \in A(\mathbf{x}^t),\ \{\mathbf{y}^t\} \to \mathbf{y} \end{array}\right\} \quad \Longrightarrow \quad \mathbf{y} \in A(\mathbf{x}).$$

(b) (Punto fijo). $\mathbf{x} \in \mathrm{SOL}(f, \widehat{X}) \iff \mathbf{x} \in A(\mathbf{x}) \iff \mathbf{x} \in \arg\min_{\mathbf{y} \in X} \Pi(\mathbf{y}, \mathbf{x})$.

(c) (Descenso). *Sea* $\mathbf{x} \in \widehat{X} \setminus \mathrm{SOL}(f, \widehat{X})$. *Si* A *es de tipo* 1 *entonces*

$$\mathbf{y} \in A(\mathbf{x}) \qquad \Longrightarrow \qquad \exists \delta > 0 : f(\mathbf{x} + \ell(\mathbf{y} - \mathbf{x})) < f(\mathbf{x}), \quad \forall \ell \in (0, \delta];$$

si A *es de tipo* 2 *entonces*

$$\mathbf{y} \in A(\mathbf{x}) \qquad \Longrightarrow \qquad \begin{cases} \Pi(\mathbf{y}, \mathbf{x}) < \Pi(\mathbf{x}, \mathbf{x}) & \textit{si es empleada en el problema CGP}, \\ f(\mathbf{y}) < f(\mathbf{x}) & \textit{si es usada en el problema RMP.} \end{cases}$$

Notar que cuando un algoritmo de tipo 2 es aplicado al problema CGP también se produce un descenso en la función objetivo f debido a la propiedad de convexidad de $\Pi(\cdot, \mathbf{x})$ y la igualdad $\nabla \Pi(\cdot, \mathbf{x}) = \nabla f(\cdot)$.

En la tabla 3.1 se recoge los diferentes pasos del algoritmo CG/SD. El algoritmo descrito en esta tabla es conceptual, pero describe importantes casos particulares. El ejemplo 2.2.3 muestra algunos métodos conocidos que pueden situarse en este marco.

Tabla 2.1: El algoritmo CG/SD

0. (*Inicialización*): Elegir un punto inicial $\mathbf{x}^0 \in X$, y tomar $t := 0$.
1. (*Problema de generación de columnas*): Elegir un algoritmo $\mathcal{A}_{\mathrm{c}}^{k_t}$, $k_t \in \mathcal{K}_{\mathrm{c}}$. Si es de tipo 1, aplicar una iteración; en caso contrario aplicar por lo menos una iteración sobre $\mathrm{CDP}(f, X)$, comenzando en $\mathbf{x}^t$. Sea $\hat{\mathbf{y}}^t$ el punto resultante.
2. (*Criterio de terminación*): Si $\mathbf{x}^t$ resuelve $\mathrm{CDP}(f, X) \rightarrow$ parar. En caso contrario, continuar
3. (*Aumento del conjunto*): Sea $X^{t+1} \subset X$ un conjunto compacto no vacío y convexo tal que el segmento $[\mathbf{x}^t, \hat{\mathbf{y}}^t] \subseteq X^{t+1}$.
4. (*Problema maestro restringido*): Elegir un algoritmo $\mathcal{A}_{\mathrm{r}}^{k_t}$, $k_t \in \mathcal{K}_{\mathrm{r}}$. Aplicar al menos una iteración de este algoritmo sobre $\mathrm{CDP}(f, X^{t+1})$ comenzando en el punto $\mathbf{x}^t$. Sea el punto resultante $\mathbf{x}^{t+1}$.
5. (*Actualización*): Tomar $t := t + 1$. Ir al paso 1.

2.2.2 El resultado básico de convergencia

El siguiente resultado de convergencia global extiende el dado en Larsson y otros [138] para el NSD a la clase de generación de columnas / descomposición simplicial. Ésta, respecto a la dada en Larsson y otros [138], permite una definición más general de la aproximación interior y una mayor flexibilidad en la elección de los algoritmos usados en los pasos 1 y 4. La demostración aquí planteada combina la dada en Larsson y otros [138] con las realizadas en Patriksson [194, 197] para algoritmos truncados de descenso con aplicaciones multievaluadas cerradas.

Teorema 2.2.2 (Convergencia global). *Suponiendo las hipótesis* 2.2.1 *entonces se cumple que la sucesión* $\{\mathbf{x}^t\}$ *de iteraciones converge a* $\mathrm{SOL}(f, X)$ *en el sentido que*

$$\left\{ d_{\mathrm{SOL}(f,X)}(\mathbf{x}^t) \right\} := \left\{ \underset{\mathbf{x} \in \mathrm{SOL}(f,X)}{\text{minimizar}} \ \|\mathbf{x}^t - \mathbf{x}\| \right\} \rightarrow 0. \tag{2.7}$$

Demostración. Si la convergencia es finita entonces la última iteración es óptima. Supondremos que la sucesión $\{\mathbf{x}^t\}$ es infinita.

La sucesión $\{X^t\}$ está formada por conjuntos compactos, convexos y no vacíos de X por tanto posee al menos un punto de acumulación, $\widetilde{X} \subseteq X$ (en el sentido de convergencia de conjuntos, ver Salinetti y

Wets [208] y el capítulo 4 de Rockafellar, y Wets [206]), que también es no vacío (de $[\mathbf{x}^{t-1}, \mathbf{y}^{t-1}] \subseteq X^t$ para todo t, y del teorema 4.18 Rockafellar y Wets [206]), compacto (de X^t es compacto para todo t y de la proposición 4.4 Rockafellar y Wets [206]), y convexo (de X^t es convexo para todo t, y la proposición 4.15 Rockafellar y Wets [206]). La demostración se centra en las subsucesiones que definen el conjunto límite $\widetilde{X}$, el cual (por simplicidad en la notación) no será formulado explícitamente.

Del hecho de que existen infinitas iteraciones y de que los conjuntos $\mathcal{K}_{\rm c}$ y $\mathcal{K}_{\rm r}$ son finitos, existirá al menos un par de estos elementos, digamos $(k_{\rm c}, k_{\rm r})$, que aparecerán un número infinito de veces en la subsucesión que definen el conjunto $\widetilde{X}$.

Como la sucesión $\{\mathbf{x}^t\}$ pertenece al conjunto compacto X, ésta tiene un conjunto no vacío y acotado de puntos de acumulación que denotamos por $\overline{X} \subseteq X$, el cual, es un conjunto cerrado (ver por ejemplo, teorema 3.7 de Rudin [207]). Por ser f continua, podemos encontrar una subsucesión convergente, denotada $\{\mathbf{x}^t\}_{t\in\mathcal{T}}$, donde $\mathcal{T} \subseteq \{0, 1, 2, \dots\}$, cuyo límite $\mathbf{x}^{\mathcal{T}} \in \arg\max_{\mathbf{x}\in\overline{X}} f(\mathbf{x})$. Entonces, $\mathbf{x}^{\mathcal{T}} \in \widetilde{X}$ (ver por ejemplo, la proposición 1.1.2 de Aubin y Frankowska [8]).

Denotamos por $\mathbf{z}^{t-1} \in X^t$, $t \in \mathcal{T}$, la primera iteración del algoritmo $\mathcal{A}_{\rm r}^{k_{\rm r}}$ aplicada al problema maestro restringido CDP(f, X^t), comenzando desde $\mathbf{x}^{t-1} \in X^t$. En cada iteración de este algoritmo se produce un descenso con respecto a f, al menos que el punto inicial sea una solución del RMP, CDP(f, X^t). Entonces para todo $t \in \mathcal{T}$, $f(\mathbf{x}^t) \leq f(\mathbf{z}^{t-1}) < f(\mathbf{x}^{t-1})$. Sea $\mathbf{x}^{\mathcal{T}-1} \in \overline{X}$ el límite de una subsucesión convergente de la sucesión $\{\mathbf{x}^{t-1}\}_{t\in\mathcal{T}}$ y sea $\mathbf{z}^{\mathcal{T}-1} \in \widetilde{X}$ un punto de acumulación de la correspondiente subsucesión $\{\mathbf{z}^{t-1}\}_{t\in\mathcal{T}}$. Tomando el límite correspondiente a este punto de acumulación, la continuidad de f conduce a $f(\mathbf{x}^{\mathcal{T}}) \leq f(\mathbf{z}^{\mathcal{T}-1}) \leq f(\mathbf{x}^{\mathcal{T}-1})$.

Debido a que $\mathbf{x}^{\mathcal{T}-1} \in \overline{X}$ y a la definición de $\mathbf{x}^{\mathcal{T}}$ obtenemos que $f(\mathbf{x}^{\mathcal{T}}) \geq f(\mathbf{x}^{\mathcal{T}-1})$ y concluimos que $f(\mathbf{x}^{\mathcal{T}}) = f(\mathbf{z}^{\mathcal{T}-1}) = f(\mathbf{x}^{\mathcal{T}-1})$. La última igualdad, junto con la propiedad de clausura y de descenso de la aplicación definida por el algoritmo $\mathcal{A}_{\rm r}^{k_{\rm r}}$ conducen a que $\mathbf{x}^{\mathcal{T}-1} \in \mathrm{SOL}\,(f, \widetilde{X})$. Entonces, usando la relación $f(\mathbf{x}^{\mathcal{T}}) = f(\mathbf{x}^{\mathcal{T}-1})$ y la definición de $\mathbf{x}^{\mathcal{T}}$, obtenemos que para todo $\mathbf{x} \in \widetilde{X}$, $f(\mathbf{x}) \geq f(\mathbf{x}^{\mathcal{T}})$, y que para todo $\mathbf{x} \in \overline{X}$, $f(\mathbf{x}) = f(\mathbf{x}^{\mathcal{T}})$. De ahí, $\overline{X} \subseteq \mathrm{SOL}\,(f, \widetilde{X})$.

Ahora, sea $\varepsilon \geq 0$, existe un número infinito de iteraciones $\mathbf{x}^{t-1}$ con $d_{\mathrm{SOL}(f,X)}(\mathbf{x}^{t-1}) \geq \varepsilon$. Esta sucesión infinita de iteraciones tiene algún punto de acumulación, denotémoslo por $\widetilde{\mathbf{x}}$, que es el límite de alguna sucesión convergente $\{\mathbf{x}^{t-1}\}_{t\in\widetilde{\mathcal{T}}}$, donde $\widetilde{\mathcal{T}} \subseteq \{0, 1, 2, \dots\}$. De lo anterior sabemos que $\widetilde{\mathbf{x}} \in \mathrm{SOL}\,(f, \widetilde{X})$.

Primero supondremos que el algoritmo $\mathcal{A}_{\rm c}^{k_{\rm c}}$ es de tipo 1. Sea $\widetilde{\mathbf{y}}$ un punto arbitrario de acumulación de la sucesión $\{\mathbf{y}^{t-1}\}_{t\in\widetilde{\mathcal{T}}}$. Del hecho de que $\mathbf{y}^{t-1} \in X^t$ para todo $t \in \widetilde{\mathcal{T}}$, se cumple que $\widetilde{\mathbf{y}} \in \widetilde{X}$ (ver por ejemplo, la proposición 1.1.2 Aubin y Frankowska [8]). De $\widetilde{\mathbf{x}} \in \widetilde{X}^*$, obtenemos que $\nabla f(\widetilde{\mathbf{x}})^{\rm T}(\widetilde{\mathbf{y}} - \widetilde{\mathbf{x}}) \geq 0$. Sin embargo $\widetilde{\mathbf{x}} \notin X^*$, entonces por cumplirse que la aplicación multievaluada definida por el algoritmo $\mathcal{A}_{\rm c}^{k_{\rm c}}$ es cerrada y de descenso, obtenemos que $\nabla f(\widetilde{\mathbf{x}})^{\rm T}(\widetilde{\mathbf{y}} - \widetilde{\mathbf{x}}) < 0$ lo que conduce a una contradicción. Obteniendo $\widetilde{\mathbf{x}} \in \mathrm{SOL}\,(f, X)$.

Supondremos ahora que el algoritmo $\mathcal{A}_{\rm c}^{k_{\rm c}}$ es de tipo 2. Para todo $t \in \widetilde{\mathcal{T}}$, sea $\mathbf{v}^{t-1} \in X$ que denota el punto obtenido por la realización de una iteración del algoritmo $\mathcal{A}_{\rm c}^{k_{\rm c}}$ sobre el problema CDP(f, X), comenzando en $\mathbf{x}^{t-1}$. En cada iteración del algoritmo $\mathcal{A}_{\rm c}^{k_{\rm c}}$ se produce un descenso con respecto a $\Pi(\cdot, \mathbf{x}^t)$, al menos que la actual iteración sea una solución al problema de generación de columnas, ello conduce a que $t \in \widetilde{\mathcal{T}}$, $\Pi(\mathbf{y}^{t-1}, \mathbf{x}^{t-1}) \leq \Pi(\mathbf{v}^{t-1}, \mathbf{x}^{t-1}) < \Pi(\mathbf{x}^{t-1}, \mathbf{x}^{t-1})$. Tomando límites correspondientes a una apropiada subsucesión infinita, la continuidad de Π nos lleva a que $\Pi(\widetilde{\mathbf{y}}, \widetilde{\mathbf{x}}) \leq \Pi(\widetilde{\mathbf{x}}, \widetilde{\mathbf{x}})$, donde $\widetilde{\mathbf{y}} \in X$ denota un punto de acumulación de la sucesión $\{\mathbf{y}^{t-1}\}_{t\in\widetilde{\mathcal{T}}}$. Como $\hat{\mathbf{y}}^{t-1} \in X^t$ para todo $t \in \widetilde{\mathcal{T}}$, se cumple que $\widetilde{\mathbf{y}} \in \widetilde{X}$. Del hecho de que $\widetilde{\mathbf{x}} \in \mathrm{SOL}\,(f, \widetilde{X})$ concluimos que $\Pi(\widetilde{\mathbf{y}}, \widetilde{\mathbf{x}}) \geq \Pi(\widetilde{\mathbf{x}}, \widetilde{\mathbf{x}})$, y entonces, $\Pi(\widetilde{\mathbf{y}}, \widetilde{\mathbf{x}}) = \Pi(\widetilde{\mathbf{x}}, \widetilde{\mathbf{x}})$. La igualdad, junto con las propiedades de descenso y de cerradura de la aplicación algorítmica $\mathcal{A}_{\rm c}^{k_{\rm c}}$, implican que $\widetilde{\mathbf{x}} \in \mathrm{SOL}\,(f, X)$. Obteniendo $\varepsilon = 0$.

Hemos identificado un par de algoritmos $k_{\rm c}$ y $k_{\rm r}$ de las colecciones $\mathcal{K}_{\rm c}$ y $\mathcal{K}_{\rm r}$, y la aplicación que define el resto de iteraciones es de la forma $C(\mathbf{x}) := \{\, \mathbf{y} \in \hat{X} \mid f(\mathbf{y}) \leq f(\mathbf{x}) \,\}$, $\mathbf{x} \in \hat{X}$, para cualquier conjunto compacto convexo no vacío $\hat{X} \subseteq X$. Podemos invocar el *teorema del paso espaciador* (ver por ejemplo, p. 231 de Luenberger [152]), que garantiza que el resultado sigue siendo cierto para toda la sucesión, gracias a las propiedades de los algoritmos $k_{\rm c}$ y $k_{\rm r}$ establecidas anteriormente. □

Ejemplo 2.2.3 Ejemplos de algoritmos CG/SD.

(1) (SD). La fase de generación de columnas en el método SD es un algoritmo de tipo 1 y está

definido por el subproblema del algoritmo de Frank–Wolfe; el hecho de que $\hat{\mathbf{y}}^t$ es un punto extremo hace que la regla (2.6) produzca $\ell_t = 1$. Entonces $X^{t+1} := \texttt{conv}\,(X^t \cup \{\hat{\mathbf{y}}^t\})$. Podemos no emplear ninguna regla de eliminación de columnas o la misma que en Von Hohenbalken [127], bajo la cual se eliminarían todos los puntos extremos de X^t que no son necesarios para representar $\mathbf{x}^t$ como combinación convexa, es decir, aquellos cuyos pesos valiesen cero. En ambos casos, X^{t+1} es un subconjunto compacto, convexo y no vacío de X y la condición establecida en el paso 3 de la tabla 3.1.

(2) (RSD). Analizado el ejemplo anterior sólo queda justificar que para cualquier elección de r se cumple la exigencia del paso 3 de la tabla 3.1 y esto es cierto debido a que las columnas son introducidas como en un esquema SD o en otro caso se almacenan explícitamente la solución $\mathbf{x}^t$ del RMP. Cuando $r = 1$ el algoritmo colapsa en el de Frank–Wolfe y $X^{t+1} := [\mathbf{x}^t, \mathbf{y}^t]$.

Notar que el algoritmo empleado por Hearn y otros [125] para el RMP se toma una iteración del método de Newton como $\mathcal{A}_{\mathrm{r}}^k$ (que es un algoritmo cerrado con la propiedad de punto fijo y que produce un descenso para f bajo las hipótesis de pseudo-convexidad para f. Por tanto se cumple las hipótesis 2.2.1).

(3) (NSD, versión exacta). Una versión exacta del método NSD de Larsson y otros [154, 141] puede ser obtenida tomando $\mathcal{A}_{\mathrm{c}}^k$ como el problema de generación de columnas $\mathrm{CDP}(\varphi(\cdot, \mathbf{x}), \nabla f, X, \mathbf{x})$, seguido por la expansión (2.6). La hipótesis 2.2.1 se cumple para esta clase de métodos, por el resultado establecido en el capítulo 2 de Patriksson [197].

(4) (NSD, versión truncada). El método NSD de Larsson y otros [154, 141] aplicado al TAP emplea un algoritmo truncado de Frank–Wolfe para (aproximadamente) resolver el problema de generación de columnas $\mathrm{CDP}(\varphi(\cdot, \mathbf{x}), \nabla f, X, \mathbf{x})$. Este tipo de algoritmo también se sitúa en este marco definiendo $\mathcal{A}_{\mathrm{c}}^k$ mediante la construcción del problema $\mathrm{CDP}(\varphi(\cdot, \mathbf{x}), \nabla f, X, \mathbf{x})$ y empleando k iteraciones del algoritmo de Frank–Wolfe para su solución (el valor de k_t en cada iteración t no es necesario darlo a priori, puede ser el resultado de la estrategia empleada de truncamiento). Su aplicabilidad está asegurada por los mismos argumentos empleados en la versión exacta del NSD, por el resultado obtenido en el capítulo 2 de Patriksson [197]. En este caso, la función de mérito $\Pi(\cdot, \mathbf{x})$ es idéntica a la función objetivo del subproblema $\mathrm{CDP}(\varphi(\cdot, \mathbf{x}), \nabla f, X, \mathbf{x})$.

(5) (Evans multidimensional). El algoritmo con subproblemas de Evans desarrollado en el capítulo 1 para el TAP-M con costes simétricos es un ejemplo de algoritmo CG/SD. El algoritmo de generación de columnas (de tipo 1) está definido por el subproblema de Evans y el algoritmo $\mathcal{A}_{\mathrm{r}}^k$ es un método de Newton proyectado Bertsekas [17]. □

2.3 Propiedades del del problema maestro restringido

La aproximación interior del conjunto X empleada en el SD es un conjunto poliedral cuyos puntos extremos también lo son de X. En el RSD la aproximación interior es ligeramente modificada, de modo que cuando una columna con peso positivo es eliminada entonces también se almacena la actual solución $\mathbf{x}$. En los métodos CG/SD podemos considerar reglas mucho más generales.

Para garantizar que la región factible de los RMP en la clase CG/SD sigue siendo un símplice como ocurre con los métodos RSD y SD, es necesario introducir ciertas propiedades de la actualización de los conjuntos X^t. En esta sección estableceremos reglas para la eliminación e introducción de columnas en la aproximación interior con el fin de garantizar que los conjuntos X^t siguen siendo símplices.

2.3.1 Aproximación interior de X

Cuando consideramos la actualización de la aproximación interior de una iteración a la siguiente debemos considerar dos fases. La primera fase se decide qué columnas deben ser eliminadas. Este criterio se puede basar en las coordenadas baricéntricas (pesos) de la representación de $\mathbf{x}^t$. Por ejemplo Hearn y otros [123, 125] eliminan las columnas no activas (peso nulo) o alternativamente se pueden eliminar todas las que sus pesos sean insignificantes. En dicho caso se debe introducir el vector $\mathbf{x}^t$

como una columna individual. En la segunda fase debemos decir qué columna se tiene que almacenar. La diferencia esencial entre el método SD formulado por Von Hohenbalken y el de sus sucesores, con el método aquí desarrollado, es que las columnas obtenidas no son necesariamente puntos extremos de X. Si empleamos la estrategia (2.6) de Larsson y otros [141] entonces la columna introducida pertenece a la frontera (relativa) de X.

En lo siguiente, usaremos $\mathcal{P}_s^t$ para denotar el conjunto de columnas generadas y retenidas en la iteración t; además, $\mathcal{P}_x^t$ es vacío o contiene una columna que corresponde a una solución de algún RMP anterior. Definimos el conjunto $\mathcal{P}^t = \mathcal{P}_s^t \cup \mathcal{P}_x^t$.

La tabla 2.2 resume varias reglas empleadas en estas dos fases, las cuales constituyen realizaciones del paso 3 en el algoritmo CG/SD de la tabla 3.1. (Ha sido incluida la inicialización necesaria de los conjuntos).

Tabla 2.2: Definición del conjunto X^t

0. (*Inicialización*): Elegir un punto inicial $\mathbf{x}^0 \in X$, tomar $t := 0$, $\mathcal{P}_s^0 = \emptyset$, $\mathcal{P}_x^0 = \{\mathbf{x}^0\}$, $\mathcal{P}^0 = \mathcal{P}_s^0 \cup \mathcal{P}_x^0$ y $X^0 = \texttt{conv}\,(\mathcal{P}^0)$. Sea r un entero positivo, y sea $\Re_+ \supset \{\varepsilon_1^t\} \to 0$.

3.1 (*Regla de eliminación de columnas*): Sea $\mathbf{x}^t = \sum_{i=1}^m \lambda_i \mathbf{p}_i$, donde $m = |\mathcal{P}^t|$ y $\mathbf{p}_i \in \mathcal{P}^t$.

3.1.a (Solución exacta para el RMP). Descartar todos los elementos $\mathbf{p}_i$ cuyo peso sea $\lambda_i = 0$.

3.1.b (Solución truncada del RMP). Descartar todos los elementos $\mathbf{p}_i$ cumpliendo

$$\nabla f(\mathbf{x}^t)^T(\mathbf{p}_i - \mathbf{x}^t) \geq \varepsilon_1^t > 0. \tag{2.8}$$

3.2 (*Extensión a la frontera relativa de X*): Sea $\hat{\mathbf{y}}^t$ el vector generado por el algoritmo de generación de columnas, y sea $\mathbf{y}^t$ definida por (2.6).

3.3 (*Regla de aumento del conjunto de columnas*):

3.3.a (Esquema de descomposición simplicial). $\mathcal{P}^{t+1} = \mathcal{P}^t \cup \{\mathbf{y}^t\}$. Tomar $X^{t+1} = \texttt{conv}\,(\mathcal{P}^{t+1})$.

3.3.b (Esquema de descomposición simplicial restringida). Si $|\mathcal{P}_s^t| < r$, entonces el conjunto $\mathcal{P}_s^{t+1} = \mathcal{P}_s^t \cup \{\mathbf{y}^t\}$, y $\mathcal{P}_x^{t+1} = \mathcal{P}_x^t$; en caso contrario, reemplazar el elemento de $\mathcal{P}_s^t$ con mínimo peso en la expresión de $\mathbf{x}^t$ por $\mathbf{y}^t$ para obtener $\mathcal{P}_s^{t+1}$, y tomar $\mathcal{P}_x^{t+1} = \{\mathbf{x}^t\}$. Finalmente, tomar $\mathcal{P}^{t+1} = \mathcal{P}_s^{t+1} \cup \mathcal{P}_x^{t+1}$ y $X^{t+1} = \texttt{conv}\,(\mathcal{P}^{t+1})$.

La regla de eliminación de columnas 3.1.a es aplicada en la descomposición simplial original (Holloway [128], Von Hohenbalken [127]), y en los posteriores desarrollos de Hearn y otros [123, 125], Larsson y otros [154, 141]. La regla 3.1.b se emplea cuando el RMP se resuelve aproximadamente. El siguiente resultado muestra que cuando el RMP se resuelve de forma exacta ambas reglas coinciden. Primeramente recordaremos la definición de soluciones ε-óptimas.

Definición 2.3.1 (ε-optimalidad). El vector $\mathbf{x} \in X$ es una solución *ε-optima* ($\varepsilon > 0$) para CDP(f, X) si se cumple

$$\nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) \geq -\varepsilon, \qquad \forall \mathbf{y} \in X. \tag{2.9}$$

□

Proposición 2.3.2 *Sea $\bar{\mathbf{x}}^t$ una solución ε-óptima para el RMP en la iteración t con $\bar{\mathbf{x}}^t = \sum_{i=1}^m \lambda_i \mathbf{p}_i$, donde $\sum_{i=1}^m \lambda_i = 1$, $\lambda_i \geq 0$, $\mathbf{p}_i \in \mathcal{P}^t$ para todo $i \in \{1, \ldots, m\}$, y $m = |\mathcal{P}^t|$. Entonces para cualquier $j \in \{1, \ldots, m\}$, se cumple que*

$$\textit{si } \nabla f(\bar{\mathbf{x}}^t)^T(\mathbf{p}_j - \bar{\mathbf{x}}^t) \geq \varepsilon_1^t > 0 \qquad \Longrightarrow \qquad \lambda_j \leq \frac{\varepsilon}{\varepsilon + \varepsilon_1^t}. \tag{2.10}$$

Demostración. Sea $\mathbf{z} = \bar{\mathbf{x}}^t + \frac{\lambda_j}{(1-\lambda_j)}(\bar{\mathbf{x}}^t - \mathbf{p}_j) = \sum_{i \neq j}^m \frac{\lambda_i}{(1-\lambda_j)}\mathbf{p}_i$. El elemento $\mathbf{z}$ pertenece a X^t porque es una combinación convexa de puntos de $\mathcal{P}^t \subset X^t$ y X^t es un conjunto convexo.

Empleando la propiedad de ε-optimalidad de $\bar{\mathbf{x}}^t$ sobre X^t, entonces se cumple

$$-\varepsilon \leq \nabla f(\bar{\mathbf{x}}^t)^T(\mathbf{z} - \bar{\mathbf{x}}^t) = -\frac{\lambda_j}{(1-\lambda_j)}\nabla f(\bar{\mathbf{x}}^t)^T(\mathbf{p}_j - \bar{\mathbf{x}}^t).$$

Por hipótesis, obtenemos que $-\frac{\lambda_j}{(1-\lambda_j)}\nabla f(\bar{\mathbf{x}}^t)^T(\mathbf{p}_j - \bar{\mathbf{x}}^t) \leq -\frac{\lambda_j}{(1-\lambda_j)}\varepsilon_1^t$. Combinando estas desigualdades obtenemos el resultado deseado. □

NOTA 2.3.3 (Equivalencia entre las reglas de eliminación de columnas). Este resultado implica que si el RMP se resuelve de forma exacta (es decir, esto significa $\varepsilon = 0$), entonces $\lambda_j = 0$ en (2.10), aquí las dos reglas coinciden. (Ver también el lema 1 de Hearn y otros [123] para un resultado similar en el contexto del método RSD.) En general, la regla 3.1.a de eliminación de columnas es más agresiva que la proporcionada por la regla 3.1.b. □

2.3.2 Las aproximaciones interiores son símplices

La aproximación interior $X^t = \texttt{conv}\,(\mathcal{P}^t)$ empleada por los métodos SD y RSD son símplices bajo la hipótesis de que los RMP son resueltos de forma exacta (teorema 3 de Hearn y otros [123]). Ahora estableceremos que esta propiedad también se cumple en los métodos CG/SD, incluso sin la suposición de que X sea un conjunto poliedral. Introducimos las siguientes definiciones y propiedades para los símplices tomadas de Rockafellar [204].

DEFINICIÓN 2.3.4 (Símplice). Sea $\{\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_m\}$ son $m+1$ puntos distintos de $\Re^n$ con $m \leq n$ donde los vectores $\mathbf{z}_1 - \mathbf{z}_0, \mathbf{z}_2 - \mathbf{z}_0, \ldots, \mathbf{z}_m - \mathbf{z}_0$ son linealmente independientes, entonces el conjunto $C = \texttt{conv}\,(\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_m)$, que es la envoltura convexa de $\{\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_m\}$, es un *m-símplice* en $\Re^n$. Además C está contenido en subespacio de dimensión m y se dice que C tiene dimensión m, o $\texttt{dim}\ C = m$. □

PROPOSICIÓN 2.3.5 (Propiedades sobre símplices).

(a) *Si* $\mathbf{x}$ *es un elemento de un m-símplice* C*, entonces* $\mathbf{x}$ *está unívocamente determinado como una combinación convexa de puntos* $\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_m$*, definiendo* C*, es decir,*

$$\mathbf{x} = \sum_{i=1}^{m} \lambda_i \mathbf{z}_i, \qquad \sum_{i=0}^{m} \lambda_i = 1, \qquad \lambda_i \geq 0, \qquad i = 0, 1, \ldots, m,$$

y los pesos $\lambda_0, \lambda_1, \ldots, \lambda_m$ *son únicos.*

(b) *Sea* $\mathbf{x}$ *un elemento de un m-símplice* C *y sea* $\lambda_i > 0$ *un peso positivo en la (única) expresión de* $\mathbf{x}$ *como combinación convexa de* $\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_m$*, entonces el conjunto*

$$\texttt{conv}\,(\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_{i-1}, \mathbf{x}, \mathbf{z}_{i+1}, \ldots, \mathbf{z}_m)$$

es un m-símplice.

(c) *Si* $\texttt{conv}\,(\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_m)$ *es un m-símplice entonces* $\texttt{conv}\,(\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_{i-1}, \mathbf{z}_{i+1}, \ldots, \mathbf{z}_m)$ *para cualquier* $i = 0, 1, \ldots, m$ *es un* $(m-1)$*-símplice.* □

El resultado principal de esta sección es el siguiente.

TEOREMA 2.3.6 (La aproximación interior es un símplice).

Suponemos que los RMP son resueltos exactamente. Entonces, el conjunto X^t *es un símplice para todo* $t \geq 0$.

DEMOSTRACIÓN. Probaremos por inducción que X^t es un símplice al inicio del paso 3.3. Cuando $t = 0$, $X^t = \{\mathbf{x}^0\}$; entonces, X^0 es un 0-símplice. Suponer ahora que X^t es un símplice para $t \geq 0$. Los elementos con pesos nulos han sido eliminados al comienzo del paso 3.3; entonces, los elementos que se mantienen en $\mathcal{P}^t$ deben tener pesos positivos. Por la hipótesis de inducción y la proposición 2.3.5.c los puntos no eliminados definen un símplice. Sin pérdida de generalidad asumimos que al principio del paso 3.3, $\mathcal{P}^t = \{\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_m\}$ y por hipótesis de inducción, $\mathcal{P}^t$ define un m-símplice. Denotamos la envoltura convexa de este conjunto por $\bar{X}^t$.

El elemento $\mathbf{x}^t$ es expresado como

$$\mathbf{x}^t = \sum_{i=0}^{m} \lambda_i \mathbf{p}_i, \qquad \text{con } \lambda_i > 0 \text{ y } \mathbf{p}_i \in \mathcal{P}^t.$$

Se sigue que $\mathbf{x}^t \in \texttt{rint}\,(\bar{X}^t)$. Probaremos que si $\mathbf{x}^t$ no es una solución óptima para el CGP, entonces $\texttt{conv}\,(\bar{X}^t \cup \{\mathbf{y}^t\})$ es un símplice, donde $\mathbf{y}^t$ es la columna añadida en la iteración $t+1$. Primeramente demostraremos que $\mathbf{x}^t$ es también solución óptima para el problema de minimización de f sobre $\texttt{aff}\,(\bar{X}^t) \cap X$, donde $\texttt{aff}\,(\bar{X}^t)$ es la envoltura afín de $\bar{X}^t$. Como f es pseudoconvexa, $\mathbf{x}^t$ es solución del problema si y sólo si

$$\nabla f(\mathbf{x}^t)^T(\mathbf{y} - \mathbf{x}^t) \geq 0, \qquad \text{para todo } \mathbf{y} \in \texttt{aff}\,(\bar{X}^t) \cap X, \tag{2.11}$$

cuestión que procedemos a probar.

Sea $\mathbf{y}$ un elemento arbitrario de $\texttt{aff}\,(\bar{X}^t) \cap X$. Si $\mathbf{y} \in \bar{X}^t \subset X^t$ entonces el punto $\mathbf{y}$ satisface la desigualdad en (2.11) porque $\mathbf{x}^t$ resuelve el RMP sobre X^t. En caso contrario $\mathbf{y} \in \texttt{aff}\,(\bar{X}^t) - \bar{X}^t$. Como $\mathbf{x}^t$ está en el interior relativo de $\bar{X}^t$, existe un único elemento, $\mathbf{z}$, en el conjunto $[\mathbf{x}^t, \mathbf{y}] \cap \texttt{rfro}\,(\bar{X}^t)$, donde $\texttt{rfro}\,(\bar{X}^t)$ es la frontera relativa de $\bar{X}^t$. Este punto cumple que $\mathbf{y} = \mathbf{x}^t + \lambda(\mathbf{z} - \mathbf{x}^t)$ para algún $\lambda > 1$. Por la optimalidad de $\mathbf{x}^t$ sobre X^t y el hecho de que $\mathbf{z} \in X^t$, obtenemos que $\nabla f(\mathbf{x}^t)^T(\mathbf{z} - \mathbf{x}^t) \geq 0$, esto conduce a $\nabla f(\mathbf{x}^t)^T(\mathbf{y} - \mathbf{x}^t) = \lambda[\nabla f(\mathbf{x}^t)^T(\mathbf{z} - \mathbf{x}^t)] \geq 0$. Esto completa la demostración de (2.11).

Si $\mathbf{x}^t$ resolviese el CGP en la iteración t, el algoritmo termina sin generar el conjunto X^{t+1}. En caso contrario, por la hipótesis 2.2.1.c, por la relación $\nabla\Pi(\cdot, \mathbf{x}^t) = \nabla f(\cdot)$, siendo $\Pi(\cdot, \mathbf{x}^t)$ pseudoconvexa y del uso de la regla (2.6), se sigue que la columna $\mathbf{y}^t$ generada en el paso 3.2 satisface $\nabla f(\mathbf{x}^t)^T(\mathbf{y}^t - \mathbf{x}^t) < 0$. Esta relación junto con la optimalidad de $\mathbf{x}^t$ sobre $\texttt{aff}\,(\bar{X}^t) \cap X$, implica que $\mathbf{y}^t \notin \texttt{aff}\,(\bar{X}^t)$. Como $\bar{X}^t$ es un m-símplice por la hipótesis de inducción $\texttt{conv}\,(\bar{X}^t \cup \{\mathbf{y}^t\})$ es entonces un $(m+1)$-símplice.

En el caso de que la relación $m = |\mathcal{P}_s^t| < r$ se cumpla, entonces el conjunto es generado por el paso 3.3.a. es $X^{t+1} = \texttt{conv}\,(\mathcal{P}^{t+1}) = \texttt{conv}\,(\bar{X}^t \cup \{\mathbf{y}^t\})$. En caso contrario, cuando se cumple que $m = |\mathcal{P}_s^t| = r$, se emplea el paso 3.3.b para definir la siguiente aproximación interior. Supondremos sin pérdida de generalidad que $\mathcal{P}_s^t = \{\mathbf{p}_0, \ldots, \mathbf{p}_{m-1}\}$, y sea $\mathcal{P}_x^t = \{\mathbf{x}'\}$. Por hipótesis, $\mathcal{P}^t$ define un m-símplice. En esta caso $X^{t+1} = \texttt{conv}\,(\mathcal{P}^{t+1})$ donde $\mathcal{P}^{t+1} = \{\mathbf{p}_0, \ldots, \mathbf{p}_{i-1}, \mathbf{p}_{i+1}, \ldots, \mathbf{p}_{m-1}, \mathbf{x}^t, \mathbf{y}^t\}$, para algún i. Este conjunto define un m-símplice porque $\texttt{conv}\,(\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_{m-1}, \mathbf{x}', \mathbf{y}^t)$ es un $(m+1)$-símplice por lo anterior, por la proposición 2.3.5.b $\texttt{conv}\,(\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_{m-1}, \mathbf{x}^t, \mathbf{y}^t)$ es un $(m+1)$-símplice y $X^{t+1} := \texttt{conv}\,(\mathbf{p}_0, \ldots, \mathbf{p}_{i-1}, \mathbf{p}_{i+1}, \ldots, \mathbf{p}_{m-1}, \mathbf{x}^t, \mathbf{y}^t)$ es un m-símplice por la proposición 2.3.5.c. Entonces, en cualquier caso, X^{t+1} es un símplice. Esto completa la demostración. □

2.4 Convergencia finita en los algoritmos CG/SD

En esta sección estableceremos algunas propiedades sobre la convergencia finita de la clase de algoritmos CG/SD. La investigación se divide en dos partes. En la primera parte, establecemos condiciones sobre el tipo de problemas y sobre las características de los algoritmos empleados en la resolución del CGP y RMP, de modo que la cara óptima sea alcanzada en un número finito de iteraciones. Cuando X es un poliedro este resultado implica que se identifican las restricciones activas en un número finito de iteraciones. En la segunda parte estudiaremos propiedades más fuertes, de modo que el anterior resultado implique que se alcance la solución óptima. Esta condición está dada bajo la hipótesis de que $\text{SOL}(f, X)$ es un conjunto de óptimos puntiagudos débiles.

Ahora introducimos algunas nociones de la geometría de los conjuntos convexos.

2.4.1 Geometría de las caras y no degeneración

Comenzaremos con algunas propiedades de las caras de los conjuntos convexos.

DEFINICIÓN 2.4.1 (Cara). Sea X un conjunto convexo en $\Re^n$. Un conjunto convexo F es una *cara* de X si los extremos de cualquier segmento en X, cuyo interior relativo tiene intersección no vacía con F, están contenidos en F. Entonces, si $\mathbf{x}$ e $\mathbf{y}$ están en X y $\lambda\mathbf{x} + (1-\lambda)\mathbf{y}$ pertenece a F para algún $0 < \lambda < 1$, entonces $\mathbf{x}$ e $\mathbf{y}$ también deben pertenecer a F. □

El siguiente resultado coincide con los teoremas 18.1–2 de Rockafellar [204].

TEOREMA 2.4.2 *Sea F una cara de un conjunto convexo X. Si Ω es un subconjunto de X cumpliendo que $\mathtt{rint}\,\Omega$ corta a F, entonces $\Omega \subset F$.* □

Un corolario de este resultado es que si el interior relativo de dos caras F_1 y F_2 tienen una intersección no vacía entonces $F_1 = F_2$. El siguiente resultado complementa al anterior, estableciendo que cada punto de un conjunto convexo pertenece al interior relativo de una única cara.

TEOREMA 2.4.3 *La colección de los interiores relativos de las caras de un conjunto convexo X es una partición de X.* □

Emplearemos la notación $F(\mathbf{x})$ para denotar la única cara F de X en la que $\mathbf{x} \in \mathtt{rint}\,F$. Notar que esta es la cara minimal conteniendo al punto $\mathbf{x}$. Caracterizaremos estas caras minimales, para lo que introduciremos las siguientes definiciones.

DEFINICIÓN 2.4.4 (El cono k-tangente $K_X(\mathbf{x})$). Un vector $\mathbf{v}$ se dice que es *k-tangente* al conjunto X en el punto $\mathbf{x}$ en X si para algún $\varepsilon > 0$, $\mathbf{x} + t\mathbf{v} \in X$ para todo valor $t \in (-\varepsilon, \varepsilon)$. El conjunto de los vectores $k-$tangentes $\mathbf{v}$ en $\mathbf{x}$ es un cono, que es denotado por $K_X(\mathbf{x})$. □

DEFINICIÓN 2.4.5 (Linealidad de K). Para cualquier cono K, llamaremos *linealidad* de K, y lo denotaremos por $\mathtt{lin}\,K$, el mayor subespacio conotenido en K, esto es, $\mathtt{lin}\,K = K \cap (-K)$. □

LEMA 2.4.6 (Caracterización de $F(\mathbf{x})$). *Sea $\mathbf{x} \in X$. Se cumple que $F(\mathbf{x}) = (\mathbf{x} + \mathtt{lin}\,K_X(\mathbf{x})) \cap X$.*

DEMOSTRACIÓN. Es fácil de ver que $(\mathbf{x} + \mathtt{lin}\,K_X(\mathbf{x})) \cap X$ es una cara de X cumpliendo que $\mathbf{x} \in \mathtt{rint}\,((\mathbf{x} + \mathtt{lin}\,K(\mathbf{x})) \cap X) \cap \mathtt{rint}\,F(\mathbf{x})$. Empleando el teorema 2.4.3 estas caras son idénticas. □

Recordaremos la definición (2.1) de cono normal $N_X(\mathbf{x})$ al conjunto X en $\mathbf{x}$. Notar que si F es una cara de X, entonces el cono normal es independiente de $\mathbf{x} \in \mathtt{rint}\,F$, por tanto denotaremos $N_X(F)$ el cono normal a la cara F. El *cono tangente* a X en $\mathbf{x}$, $T_X(\mathbf{x})$, es el cono polar de $N_X(\mathbf{x})$.

La cara de X que está expuesta por el vector $\mathbf{d} \in \Re^n$ (la *cara expuesta*) es el conjunto

$$E_X(\mathbf{d}) = \arg\max_{\mathbf{y} \in X} \mathbf{d}^T\mathbf{y}.$$

Para un conjunto convexo general, $\mathbf{x} \in E_X(\mathbf{d})$ si y sólo si se cumple que $\mathbf{d} \in N_X(\mathbf{x})$ (ver por ejemplo, Burke y Moré [36]). Para conjuntos poliedrales, la cara expuesta es independiente de la elección del vector $\mathbf{d} \in \mathtt{rint}\,N_X(\mathbf{x})$. Además, la cara de cualquier conjunto poliedral está expuesta por algún vector $\mathbf{d}$. Estos resultados se cumplen para conjuntos convexos más generales. En el análisis de propiedades de identificación de los algoritmos CG/SD se centrarán en las propiedades de las caras de X.

DEFINICIÓN 2.4.7 (Caras casi-poliedrales). Una cara F de X es *casi-poliedral* sii

$$\mathtt{aff}\, F = \mathbf{x} + \mathtt{lin}\, T_X(\mathbf{x}), \qquad \mathbf{x} \in \mathtt{rint}\, F. \tag{2.12}$$

□

El interior relativo de una cara casi-poliedral es equivalente al concepto de *cara abierta* definido en Dunn [68]. Cada cara de un conjunto poliedral X es casi-poliedral, pero no es cierto para conjuntos convexos generales, como muestra el ejemplo dado en Burke y Moré [35]. Además, una cara casi-poliedral no necesariamente es un conjunto poliedral. Las caras casi-poliedrales están expuestas por algún vector de $\mathtt{rint}\, N_X(F)$, y tienen otras propiedades comunes con las caras de los conjuntos poliedrales. Ver Burke y Moré [35, 36] para propiedades de las caras casi-poliedrales.

Ahora volveremos a estudiar las propiedades de la cara óptima de X. La siguiente definición extiende la definición de cara óptima dada para problemas con solución única (ver por ejemplo, Wolfe [238]) a problemas con mútiples soluciones.

DEFINICIÓN 2.4.8 (Cara óptima). La *cara óptima* de CDP(f, X), es

$$F^* = \bigcap_{\mathbf{x}^* \in \mathrm{SOL}(f,X)} F_{\mathbf{x}^*},$$

donde $F_{\mathbf{x}^*} = \{\, \mathbf{y} \in X \mid \nabla f(\mathbf{x}^*)^T(\mathbf{y} - \mathbf{x}^*) = 0 \,\}$. □

El conjunto F^* es la intersección de caras minimales. Es elemental mostrar que si la función f es pseudoconvexa, entonces se cumple que $F^* \supset \mathrm{SOL}(f, X)$. Notar que la cara $F_{\mathbf{x}^*}$ es la cara expuesta $E_X(-\nabla f(\mathbf{x}^*))$.

En el caso donde f es convexa, entonces la función gradiente ∇f es constante en $\mathrm{SOL}(f, X)$, por un resultado dado en Burke y Ferris Burke y Ferris [33]. Entonces en este caso, $F^* = F_{\mathbf{x}^*} = E_X(-\nabla f(\mathbf{x}^*))$ para cualquier $\mathbf{x}^* \in \mathrm{SOL}(f, X)$, simplificando la anterior definición.

Bajo la siguiente condición de regularidad sobre las soluciones óptimas se ha demostrado la identificación de la cara óptima para varios algoritmos (ver por ejemplo, Dunn [68], Burke y Moré [35], Patriksson [197]):

DEFINICIÓN 2.4.9 (Solución no degenerada). Una solución óptima $\mathbf{x}^*$ de CDP(f, X), es *no degenerada* si cumple

$$-\nabla f(\mathbf{x}^*) \in \mathtt{rint}\, N_X(\mathbf{x}^*) \tag{2.13}$$

□

Notar que esta condición de regularidad es independiente de la representación del conjunto X y en el caso de que esté descrito explícitamente mediante restricciones, entonces esta relación es más débil que la condición de *estrictamente complementariedad* (Burke y Moré [35]). Antes de establecer el resultado de identificación finita para los algoritmos CG/SD, introduciremos dos condiciones de regularidad que han sido empleadas en la literatura y las relaciones de una con otra.

DEFINICIÓN 2.4.10 (Condiciones de regularidad).

(1) (Estabilidad geométrica Marcotte y Dussault [160]). Una solución óptima $\mathbf{x}^*$ es *geométricamente estable* sii

$$\text{si } \nabla f(\mathbf{x}^*)^T(\mathbf{x} - \mathbf{x}^*) = 0 \qquad \Longrightarrow \qquad \mathbf{x} \in F^*. \tag{2.14}$$

(2) (Regularidad geométrica, Dussault y Marcotte [69]). La cara óptima F^* es *geométricamente regular* si

$$\mathrm{SOL}(f, X) \ \subset \mathtt{rint}\, F^*, \tag{2.15}$$

y el conjunto $\mathrm{SOL}(f, X)$ es no degenerado en el sentido de la definición 2.4.9. □

Una condición suficiente para la estabilidad geométrica es la convexidad de f sobre X, como remarcamos anteriormente.

Las nociones de estabilidad y regularidad geométrica son equivalentes cuando X es un conjunto poliedral acotado (ver el corolario 2.4 de Dussault y Marcotte [69]). El siguiente resultado extiende esta caracterización al caso general de conjuntos convexos bajo la suposición de no degeneración. La cualificación de restricciones de Guignard [113] utilizada implica que $N_X(\mathbf{x})$ es un cono poliedral para cada $\mathbf{x}$. Los conjuntos poliedrales X satisfacen automáticamente esta cualificación de restricciones.

TEOREMA 2.4.11 (Equivalencia entre las condiciones de regularidad). *Suponer que se cumple la cualificación de restricciones de Guignard. Además se supone que las soluciones óptimas son no degeneradas. Entonces, las siguientes tres afirmaciones son equivalentes.*

(a) *Cada* $\mathbf{x}^* \in \mathrm{SOL}(f, X)$ *es geométricamente estable.*

(b) F^* *es geométricamente regular.*

(c) F^* *es casi-poliedral y se cumple* $F^* = F(\mathbf{x}^*)$ *para todo* $\mathbf{x}^* \in \mathrm{SOL}(f, X)$.

DEMOSTRACIÓN. [(a) $\Longrightarrow$ (b)]. Sea $\mathbf{x}^* \in \mathrm{SOL}(f, X)$. Por definición, $\mathbf{x}^* \in \mathtt{rint}\, F(\mathbf{x}^*)$ para una única cara $F(\mathbf{x}^*)$ de X. También es claro que $F(\mathbf{x}^*) = F^*$, por el hecho de que F^* es la cara minimal que contiene a $\mathbf{x}^*$. Además, por la estabilidad geométrica se cumple $F^* = F_{\mathbf{x}^*}$. Entonces, $F^* = F_{\mathbf{x}^*} = F(\mathbf{x}^*)$, y se sigue que $\mathbf{x}^* \in \mathtt{rint}\, F^*$.

[(b) $\Longrightarrow$ (c)]. La siguiente relación se cumple: $\mathbf{x}^* \in (\mathtt{rint}\, F^*) \cap (\mathtt{rint}\, F(\mathbf{x}^*))$ para todo $\mathbf{x}^* \in \mathrm{SOL}(f, X)$. Como la colección del interior de las caras de X es una partición de X, entonces $F^* = F(\mathbf{x}^*)$ para todo $\mathbf{x}^* \in \mathrm{SOL}(f, X)$.

Ahora probaremos que F^* es una cara casi-poliedral. Como $\mathbf{x}^* \in \mathtt{rint}\, F^*$, demostraremos que $F^* = (\mathbf{x}^* + \mathtt{lin}\,(T_X(\mathbf{x}^*)) \cap X$. Comenzaremos mostrando que $F^* \subset (\mathbf{x}^* + \mathtt{lin}\,(T_X(\mathbf{x}^*)) \cap X$. Empleando el lema 2.4.6, obtenemos que $F^* = (\mathbf{x}^* + K_X(\mathbf{x}^*)) \cap X$. Además, $K(\mathbf{x}^*) \subset T_X(\mathbf{x}^*)$ y $\mathtt{lin}\, K(\mathbf{x}) = K(\mathbf{x})$, que establece la inclusión. Ahora probaremos la otra inclusión. Sea $\mathbf{x}^* + \mathbf{v} \in (\mathbf{x}^* + \mathtt{lin}\,(T_X(\mathbf{x}^*)) \cap X$. Empleando el lema 2.7 de Burke y Moré [35], se sigue que $\mathbf{v} \in N^{\perp}(\mathbf{x}^*)$, donde $\perp$ denota el complemento ortogonal. Por otro lado se cumple $N^{\perp}(\mathbf{x}^*) = N^{\perp}(\mathbf{z})$ para todo $\mathbf{z} \in \mathtt{rint}\, F^*$ (ver el teorema 2.3 de Burke y Moré [35]). Esto implica que $\mathbf{v} \in N^{\perp}(\mathbf{y}^*)$ para cada $\mathbf{y}^* \in \mathrm{SOL}(f, X)$. Como se cumple $-\nabla f(\mathbf{y}^*) \in N_X(\mathbf{y}^*)$, $\nabla f(\mathbf{y}^*)^T \mathbf{v} = 0$ para cada $\mathbf{y}^* \in \mathrm{SOL}(f, X)$. Esta relación establece que $\nabla f(\mathbf{y}^*)^T(\mathbf{x}^* + \mathbf{v} - \mathbf{y}^*) = \nabla f(\mathbf{y}^*)^T \mathbf{v} + \nabla f(\mathbf{y}^*)^T(\mathbf{x}^* - \mathbf{y}^*) = 0$, y $\mathbf{x}^* + \mathbf{v} \in F_{\mathbf{y}^*}$ para todo $\mathbf{y}^* \in \mathrm{SOL}(f, X)$. Por definición de F^*, obtenemos que $\mathbf{x}^* + \mathbf{v} \in F^*$.

[(c) $\Longrightarrow$ (a)]. Sea $\mathbf{x}^* \in \mathrm{SOL}(f, X)$. Probaremos que si $\nabla f(\mathbf{x}^*)^T(\mathbf{z} - \mathbf{x}^*) = 0$ para $\mathbf{z} \in X$, entonces $\mathbf{z} \in F^*$. Como $N_X(\mathbf{x}^*)$ es un cono poliedral y $-\nabla f(\mathbf{x}^*) \in N_X(\mathbf{x}^*)$, existe un conjunto de vectores y de escalares que cumplen $-\nabla f(\mathbf{x}^*) = \sum \lambda_i \mathbf{v}_i$, donde $\lambda_i \geq 0$. El punto $\mathbf{x}^*$ es no degenerado; empleando el lema 3.2 de Burke y Moré [35] estos coeficientes deben ser positivos. La relación $0 = -\nabla f(\mathbf{x}^*)^T(\mathbf{z} - \mathbf{x}^*) = \sum \lambda_i \mathbf{v}_i^T(\mathbf{z} - \mathbf{x}^*)$ implica que $\mathbf{v}_i^T(\mathbf{z} - \mathbf{x}^*) = 0$ para todo i y de aquí $(\mathbf{z} - \mathbf{x}^*) \in N^{\perp}(\mathbf{x}^*)$. Empleando el lema 2.7 de Burke y Moré [35], $N^{\perp}(\mathbf{x}^*) = \mathtt{lin}\, T_X(\mathbf{x}^*)$. Por hipótesis, $F^* = (\mathbf{x}^* + \mathtt{lin}\,(T_X(\mathbf{x}^*)) \cap X$ y $\mathbf{z} = \mathbf{x}^* + (\mathbf{z} - \mathbf{x}^*) \in (\mathbf{x}^* + \mathtt{lin}\,(T_X(\mathbf{x}^*)) \cap X = F^*$. Esto completa la demostración. □

2.4.2 Identificación finita de la cara óptima

Los resultados de identificación de la cara óptima en un número finito de iteraciones serán establecidos bajo las siguientes hipótesis en la construcción y resolución de la sucesión del RMP.

HIPÓTESIS 2.4.12 (Condiciones del RMP). *Asumiremos que se cumple una de las dos siguientes condiciones*

(1) $r \geq \mathtt{dim}\, F^* + 1$, *y los RMP son resueltos exactamente.*

(2) $r = \infty$ *y los RMP son resueltos de modo que* $\mathbf{x}^t$ *es* ε^t*-óptima con* $\Re_+ \supset \{\varepsilon^t\} \to 0$.

TEOREMA 2.4.13 (Resultados de identificación). *Sea $\{\mathbf{x}^t\}$ e $\{\hat{\mathbf{y}}^t\}$ dos sucesiones de puntos generadas por un algoritmo CG/SD, la primera es la sucesión de iteraciones generadas en el RMP y la segunda es la sucesión de columnas generadas en el CGP. Asumimos que la sucesión $\{\mathbf{x}^t\}$ converge a $\mathbf{x}^* \in \mathrm{SOL}(f, X)$.*

(a) *Supongamos que los RMP son resueltos de forma exacta. Si existe un número entero τ_1 tal que $\mathbf{x}^t \in \mathtt{rint}\, F^*$ para todo $t \geq \tau_1$, entonces existe un número entero postivo τ_2 cumpliendo*

$$\hat{\mathbf{y}}^t \in F^*, \qquad t \geq \tau_2.$$

(b) *Supongamos que se cumplen las hipótesis 2.4.12 y $SOL(f, X)$ es geométricamente estable. Si existe un entero positivo τ_1 cumpliendo que $\hat{\mathbf{y}}^t \in F^*$ para todo $t \geq \tau_1$, entonces existe un número entero positivo τ_2 cumpliendo que*

$$\mathbf{x}^t \in \mathtt{rint}\, F^*, \qquad t \geq \tau_2.$$

DEMOSTRACIÓN. (a) Sea $t \geq \tau_1$, entonces existe $\mathbf{x}^t \in \mathtt{rint}\, F^*$. Primeramente demostraremos que la columna $\mathbf{y}^{t+1}$ generada por la regla (2.6) pertenece a la cara óptima F^*. Del hecho de que en cada iteración los RMP son resueltos exactamente tenemos, $\mathbf{x}^{t+1} \in X^{t+1} - X^t$, por tanto $\mathbf{x}^{t+1} = \lambda \mathbf{y}^{t+1} + (1-\lambda)\mathbf{z}$, donde $\mathbf{z} \in X^t$ y $0 < \lambda \leq 1$. Si $\lambda = 1$ entonces el resultado se obtiene trivialmente . En caso contrario, empleando el hecho de que el conjunto F^* es una cara, y que $\mathbf{x}^{t+1} \in (\mathbf{z}, \mathbf{y}^{t+1}) \cap \mathtt{rint}\, F^*$, tenemos que $[\mathbf{z}, \mathbf{y}^{t+1}] \subset F^*$, y por tanto $\mathbf{y}^{t+1}$ pertenece a la cara óptima. Ahora también mostraremos que $\hat{\mathbf{y}}^{t+1}$ pertenece a la cara F^*. Como $\mathbf{x}^{t+1} \in \mathtt{rint}\, F^* \subset F^*$, llegamos a que $[\mathbf{x}^{t+1}, \mathbf{y}^{t+1}] \subset F^*$ y del hecho de que $\hat{\mathbf{y}}^{t+1} \in [\mathbf{x}^{t+1}, \mathbf{y}^{t+1}]$, se obtiene el resultado.

(b) Sea $t \geq \tau_1$, entonces $\hat{\mathbf{y}}^t \in F^*$. Si $\hat{\mathbf{y}}^t \in \mathtt{rint}\, F^*$, como F^* es una cara de X, se cumple $\mathbf{y}^t$ pertenece F^*. En caso contrario $\hat{\mathbf{y}}^t \in \mathtt{rfro}\, F^*$, donde la estrategia (2.6) produce $\mathbf{y}^t = \hat{\mathbf{y}}^t \in F^*$. Esto garantiza que $\{\mathbf{y}^t\}_{t \geq \tau_1} \subset F^*$.

Probaremos que si existe un elemento $\mathbf{z}$ que nunca es eliminado del conjunto $\mathcal{P}^t$ para $t \geq \tau$, entonces $\mathbf{z}$ está en la cara óptima. Esto es cierto si $\mathbf{z}$ no cumple el criterio de eliminación de columnas en ninguna iteración $t \geq \tau$.

Si el RMP se resuelve de forma exacta, entonces se debe satisfacer que para $t \geq \tau$ la solución del RMP en la iteración t se expresa por

$$\mathbf{x}^{t+1} = \lambda_{\mathbf{z}}^t \mathbf{z} + \sum_{i=1}^{n_t} \lambda_i^t \mathbf{p}_i, \qquad 0 < \lambda_{\mathbf{z}}^t,\ 0 \leq \lambda_i^t,\ i = 1, \ldots, n_t \text{ y } \lambda_{\mathbf{z}}^t + \sum_{i=1}^{n_t} \lambda_i^t = 1,\ \mathbf{p}_i \in \mathcal{P}^t.$$

En esta situación obtenemos que $\nabla f(\mathbf{x}^{t+1})^T(\mathbf{z} - \mathbf{x}^{t+1}) = 0$, porque en caso contrario el elemento $\mathbf{x}$ debería tener un peso positivo por la optimalidad de $\mathbf{x}^{t+1}$, lo que implicaría, por la proposición 2.3.2, que $\lambda_{\mathbf{z}}^t = 0$, contradiciendo la suposición que $\lambda_{\mathbf{z}}^t > 0$. Tomando límites en la anterior igualdad obtenemos

$$\nabla f(\mathbf{x}^*)^T(\mathbf{z} - \mathbf{x}^*) = \lim_{t \to \infty} \nabla f(\mathbf{x}^{t+1})^T(\mathbf{z} - \mathbf{x}^{t+1}) = 0.$$

Por otro lado si el RMP se resolviera aproximadamente, el hecho de que $\mathbf{z}$ no cumple el criterio de eliminación de columnas en ninguna de la iteraciones implica que $\nabla f(\mathbf{x}^{t+1})^T(\mathbf{z} - \mathbf{x}^{t+1}) < \varepsilon_1^{t+1}$. Empleando la continuidad de la función $\nabla f(\mathbf{x})$ y tomando límites en la desigualdad, obtenemos

$$\nabla f(\mathbf{x}^*)^T(\mathbf{z} - \mathbf{x}^*) = \lim_{t \to \infty} \nabla f(\mathbf{x}^{t+1})^T(\mathbf{z} - \mathbf{x}^{t+1}) \leq \lim_{t \to \infty} \varepsilon_1^t = 0.$$

Por la optimalidad de $\mathbf{x}^*$, obtenemos que $\nabla f(\mathbf{x}^*)^T(\mathbf{z} - \mathbf{x}^*) \geq 0$, lo que implicaría $\nabla f(\mathbf{x}^*)^T(\mathbf{z} - \mathbf{x}^*) = 0$.

En ambos casos obtenemos que $\nabla f(\mathbf{x}^*)^T(\mathbf{z} - \mathbf{x}^*) = 0$ y del hecho de que $\mathbf{x}^*$ es geométricamente estable se deduce que $\mathbf{z} \in F^*$.

Esto también prueba que cualquier elemento del conjunto $\cup_{t > \tau} \mathcal{P}^t$, que no esté en la cara óptima, debe ser eliminado en alguna iteración. Primero consideramos el caso de que se tome $r = \infty$. Por lo anterior sabemos que $\mathbf{y}^t \in F^*$ para $t \geq \tau_1$. Por la construcción de la aproximación interior, existirá un número entero τ_2 tal que $\mathcal{P}^t \subset F^*$, par $t \geq \tau_2$. Entonces obtenemos que $\mathbf{x}^t \in X^t = \mathtt{conv}\,(\mathcal{P}^t) \subset F^*$, $t \geq \tau_2$. Como los pesos son positivos implica que el actual $\mathbf{x}^t \in \mathtt{rint}\, F^*$.

En el caso de que $r < \infty$, es concebible que exista una iteración t en la que un elemento $\mathbf{x}^t$ sea introducido en $\mathcal{P}^t$. Demostraremos que bajo la hipótesis de que $r \geq \dim F^* + 1$ y de que los RMP son resueltos exactamente, ningún $\mathbf{x}^t$ se introduce en $\mathcal{P}^t$. La conclusión es entonces la misma que para el caso $r = \infty$.

Empleando el resultado anterior, $\mathcal{P}_s^t \subset F^*$ para todo $t \geq \tau_2$. Esto implica que $\dim(\mathtt{conv}(\mathcal{P}_s^t)) \leq \dim F^*$. Denotamos $\dim(\mathtt{conv}(\mathcal{P}_s^t)) = m$ para usarlo posteriormente. Como los RMP están resueltos exactamente, por el teorema 2.3.6, X^t es un símplice para cualquier $t \geq 0$, entonces $\mathtt{conv}(\mathcal{P}_s^t)$ es un m-símplice por la proposición 2.3.5.c.

Supondremos que $\mathbf{x}^t \notin \mathrm{SOL}(f, X)$ para $t \geq \tau_2$. De acuerdo a la demostración del teorema 2.3.6, $\mathtt{conv}(\mathcal{P}_s^t \cup \{\mathbf{y}^t\})$ es entonces un $(m+1)$-símplice; además se cumple para $t \geq \tau_2$, $\mathcal{P}_s^t \cup \{\mathbf{y}^t\} \subset F^*$. Ello conduce a

$$|\mathcal{P}_s^t| = \dim(\mathtt{conv}(\mathcal{P}_s^t)) + 1 = \dim(\mathtt{conv}(\mathcal{P}_s^t \cup \{\mathbf{y}^t\})) \leq \dim F^* \leq r - 1,$$

lo que implica $|\mathcal{P}_s^t| < r$. Esto implica que se cumple $\mathcal{P}_x^{t+1} = \mathcal{P}_x^t$ para todo $t \geq \tau_2$ (paso 3.3.b). Esto completa la demostración. □

Finalmente estableceremos una condición suficiente bajo la cual se cumple $\hat{\mathbf{y}}^t \in F^*$ para todo $t \geq \tau_1$. Con esta finalidad, introducimos el siguiente concepto.

Definición 2.4.14 (Gradiente proyectado). Sea $\mathbf{x} \in X$. El *gradiente* de f en $\mathbf{x}$ *proyectado* en X se define

$$\nabla^X f(\mathbf{x}) := \arg \min_{\nu \in T_X(\mathbf{x})} \|\nabla f(\mathbf{x}) + \nu\|. \tag{2.16}$$

□

Aquí el gradiente proyectado en $\mathbf{x}$ coincide con $P_{T_X(\mathbf{x})}[-\nabla f(\mathbf{x})]$, donde $P_S[\cdot]$ es la proyección euclídea dentro del conjunto convexo S. Notar que por su definición, se cumple $-\nabla f(\mathbf{x}) \in N_X(\mathbf{x})$ si y y sólo si $P_{T_X(\mathbf{x})}[-\nabla f(\mathbf{x})] = \mathbf{0}$. El siguiente resultado muestra que los algoritmos que fuerzan la proyección del gradiente a cero identifican la cara óptima en un número finito de iteraciones. En las aplicaciones a conjuntos poliedrales asumiremos que todas las restricciones son desigualdades. $\mathcal{I}(\mathbf{x})$ y λ_i^* denotan respectivamente el subconjunto de restricciones activas en $\mathbf{x}$ y sus multiplicadores de Lagrange óptimos.

Teorema 2.4.15 (Caracterización de la identificación, Burke y Moré [35, 36]). *Supongamos que* $\{\mathbf{x}^t\} \subset X$ *converge a* $\mathbf{x}^* \in \mathrm{SOL}(f, X)$.

(a) *Supongamos que X es un conjunto poliedral. Entonces, existe un entero positivo τ cumpliendo que*

$$\{\nabla^X f(\mathbf{x}^t)\} \to \mathbf{0}$$
$$\Longleftrightarrow$$
$$\mathbf{x}^t \in E_X[-\nabla f(\mathbf{x}^*)], \qquad t \geq \tau$$
$$\Longleftrightarrow$$
$$\mathcal{I}(\mathbf{x}^t) = \{\, i \in \mathcal{I}(\mathbf{x}^*) \mid \lambda_i^* > 0 \,\}, \qquad t \geq \tau.$$

(b) *Supongamos que $\mathbf{x}^*$ es no degenerado. Además, supongamos que se cumple $\mathbf{x}^* \in \mathtt{rint}\, F^*$, donde la cara F^* de X es casi-poliedral. Entonces, existe un entero positivo τ cumpliendo*

$$\{\nabla^X f(\mathbf{x}^t)\} \to \mathbf{0}$$
$$\Longleftrightarrow$$
$$\mathbf{x}^t \in \mathtt{rint}\, F^*, \qquad t \geq \tau.$$

Si además X es un poliedro. Entonces, a la anterior equivalencia puede ser añadida la relación:

$$\mathcal{I}(\mathbf{x}^t) = \mathcal{I}(\mathbf{x}^*), \qquad t \geq \tau.$$

La aplicación inmediata de este resultado a la clase de algoritmos CG/SD es la siguiente.

TEOREMA 2.4.16 (Identificación finita de la cara óptima). *Supongamos que se cumple la cualificación de restricciones de Guignard. Supongamos que* $\mathrm{SOL}(f, X)$ *es un conjunto de soluciones óptimas no degeneradas, que se cumple la hipótesis 2.4.12 y que* F^* *es geométricamente regular. Si la sucesión* $\{\hat{\mathbf{y}}^t\}$ *es tal que cumple* $\{\nabla^X f(\hat{\mathbf{y}}^t)\} \to \mathbf{0}$*, entonces existe un entero positivo* τ *cumpliendo que* $\mathbf{x}^t \in \mathtt{rint}\, F^*$ *para cada* $t \geq \tau$*.*

DEMOSTRACIÓN. El resultado se obtiene aplicando los teoremas 2.4.11, 2.4.13.b y 2.4.15.b. □

Son varios los algoritmos que fuerzan la proyección del gradiente a cero como lo es el gradiente proyectado o el algoritmo de programación cuadrática secuencial (Burke y Moré [35]). Patriksson [197] en los teorema 7.11 y en 7.19 establece un resultado general para los algoritmos de generación de columnas derivados de la resolución de los subproblemas $\mathrm{CDP}(\varphi(\cdot, \mathbf{x}), \nabla f, X, \mathbf{x})$ definido en la sección 2.1.1. Estos algoritmos fuerzan el gradiente a cero bajo la hipótesis que la función $\varphi(\cdot, \mathbf{x})$ es estrictamente convexa.

2.4.3 Identificación de la solución óptima en un número finito de iteraciones

La propiedad de la convergencia finita de los algoritmos SD y RSD está basada en el hecho de que el número de columnas candidatas (que son los puntos extremos de un poliedro) es finito. Esta propiedad se pierde en la generalización de los algoritmos CG/SD debido a la no linealidad del conjunto factible y/o del principio de generación de columnas. Un ejemplo ilustrativo de este hecho se da a continuación.

EJEMPLO 2.4.17 (Convergencia asintótica de los algoritmos CG/SD). Considerar el siguiente ejemplo de $\mathrm{CDP}(f, X)$:

$$\begin{aligned} \text{minimizar} \quad & f(x_1, x_2) := \left(x_1 - \frac{1}{2}\right)^2 + x_2, \\ \text{subjeto a} \quad & -2x_1 - x_2 \leq -1, \\ & 2x_1 - x_2 \leq 1, \\ & x_2 \leq 1. \end{aligned}$$

Se define el principio generador de columnas del siguiente modo: sea el punto factible $\mathbf{x}$, entonces la columna generada es $\mathbf{y} = (-1/2 + \sqrt{1 + f(\mathbf{x})/2}, -2 + 2\sqrt{1 + f(\mathbf{x})/2})$. Es trivial mostrar que para cualquier punto factible $\mathbf{x} \neq \mathbf{x}^* = (\frac{1}{2}, 0)^{\mathrm{T}}$ se cumple que $f(\mathbf{y}) = \frac{1}{2} f(\mathbf{x}) < f(\mathbf{x})$. Claramente la condición para la convergencia asintótica hacia la única solución $\mathbf{x}^*$ se cumple. Si para alguna restricción X^t del conjunto factible se cumple que $\mathbf{x}^* \in X^t$, entonces $\mathbf{x}^*$ es un punto extremos de X^t porque $\mathbf{x}^*$ es un punto extremo de X y $X^t \subset X$. Supondremos que la regla usada para definir la aproximación interior es la dada en el paso 3.3.a. Por tanto $\mathbf{x}^* \in X^t$ si y sólo si $\mathbf{y}^t = \mathbf{x}^*$. Estableceremos por inducción que $\mathbf{x}^* \notin X^t$ para cualquier t, de lo que se deriva que la convergencia tiene que ser únicamente asintótica. Para $t = 0$, suponer que $X^0 = \{\mathbf{x}^0\} \neq \{\mathbf{x}^*\}$ y que $\mathbf{x}^* \notin X^t$ para algun $t \geq 0$. Empleando que el RMP está resuelto exactamente, entonces $\mathbf{x}^{t+1} \neq \mathbf{x}^*$, conduciendo a $f(\mathbf{x}^{t+1}) > 0$, y por tanto a que se cumpla $f(\mathbf{y}^{t+1}) > 0$. Esto implica que $\mathbf{y}^{t+1} \neq \mathbf{x}^*$, y empleando el argumento anterior $\mathbf{x}^* \notin X^{t+1}$. Esto completa la prueba. □

Para establecer la convergencia finita de un algoritmo CG/SD, debemos imponer una propiedad al conjunto de soluciones óptima $\mathrm{SOL}(f, X)$ que es más fuerte que la no degeneración y las condiciones de regularidad consideradas anteriormente en el teorema 2.4.11. Como veremos, esto implica que el número de columnas necesarias para expandir la cara óptima es finita, de hecho la cara óptima es igual al conjunto formado por la solución óptima, y por tanto el resultado del teorema 2.4.16 implica que la convergencia es finita.

La condición de regularidad utilizada es la siguiente.

DEFINICIÓN 2.4.18 (Mínimo débilmente puntiagudo, Polyak [201]). El conjunto $\mathrm{SOL}(f, X)$ es un conjunto de *mínimos débilmente puntiagudos* si para algún $\alpha > 0$,

$$f(\mathbf{x}) - f(P_{\mathrm{SOL}(f,X)}(\mathbf{x})) \geq \alpha \|\mathbf{x} - P_{\mathrm{SOL}(f,X)}(\mathbf{x})\|, \qquad \mathbf{x} \in X. \tag{2.17}$$

□

Polyak [201] establecieron que el método del gradiente proyectado converge finitamente bajo la hipótesis de la propiedad de mínimo débilmente puntiagudo. Burke y Ferris [34] extendieron este resultado para caracterizar los algoritmos que convergen en un número finito de iteraciones. Extendieron la caraterización del teorema 2.4.15.b a los algoritmos que alcanzan la cara óptima en un número finito de iteraciones del siguiente modo:

TEOREMA 2.4.19 (Caracterización de la convergencia finita, teorema 4.7 de Burke y Ferris [34]). *Suponer que f es convexa y que el conjunto $\mathrm{SOL}(f, X)$ es de mínimos débilmente puntiagudos para $\mathrm{CDP}(f, X)$. Suponer que $\{\mathbf{x}^t\} \subset X$ converge a $\mathrm{SOL}(f, X)$. Entonces, existe un entero τ cumpliendo*

$$\begin{gathered} \{\nabla^X f(\mathbf{x}^t)\} \to \mathbf{0} \\ \Longleftrightarrow \\ \mathbf{x}^t \in \mathrm{SOL}(f, X), \qquad t \geq \tau. \end{gathered}$$

□

Emplearemos este teorema como sigue.

TEOREMA 2.4.20 (Convergencia finita de los algoritmos CG/SD). *Suponer que f es convexa y que $\mathrm{SOL}(f, X)$ es un conjunto de mínimos débilmente puntiagudos para $\mathrm{CDP}(f, X)$. Suponer que se cumple la hipótesis 2.4.12. Si la sucesión $\{\hat{\mathbf{y}}^t\}$ es tal que se cumple $\{\nabla^X f(\hat{\mathbf{y}}^t)\} \to \mathbf{0}$, entonces existe un entero positivo τ cumpliendo $\mathbf{x}^t \in \mathtt{rint}\, \mathrm{SOL}(f, X)$ para cada $t \geq \tau$.*

DEMOSTRACIÓN. Por la convexidad de f, el teorema 4.1 de Burke y Ferris [34] muestra que la cara óptima F^* es igual al conjunto de soluciones óptimas $\mathrm{SOL}(f, X)$, que además es la cara expuesta por el vector $-\nabla f(\mathbf{x}^*)$ para cualquier $\mathbf{x}^* \in \mathrm{SOL}(f, X)$. Como ya se vió en el contexto del teorema 2.4.11, esto implica que $\mathrm{SOL}(f, X)$ es una cara geométricamente regular.

Por hipótesis se cumple $\{\nabla^X f(\hat{\mathbf{y}}^t)\} \to \mathbf{0}$. Bajo la hipótesis de mínimo débilmente puntiagudo, teorema 2.4.20 establece que existe un número entero τ_1 cumpliendo que $\hat{\mathbf{y}}^t \in \mathrm{SOL}(f, X)$ para cada $t \geq \tau_1$.

Estamos en las hipótesis del teorema 2.4.13.b, entonces se sigue que existe un entero τ_2 cumpliendo $\mathbf{x}^t \in \mathtt{rint}\,(\mathrm{SOL}(f, X))$ para todo $t \geq \tau_2$. □

Capítulo 3

La clase de algoritmos CG/SD: estudio computacional

Resumen

En el capítulo anterior se presentó una clase de algoritmos de generación de columnas (CG) / descomposición simplicial (SD) para problemas de optimización convexa diferenciable. Este capítulo completa el estudio teórico del capítulo anterior, analizando numéricamente la eficiencia de estos métodos.

Primeramente se generaliza la clase CG/SD a problemas de optimización (no necesariamente convexos ni diferenciables) que permite interpretar la clase CG/SD como un procedimiento para acelerar la convergencia del algoritmo empleado para generar las columnas en el CGP. Estos algoritmos deben ser iterativos, convergentes y la sucesión de puntos generada debe estar contenida en la región factible.

Se han abordado cinco tipos de experimentos numéricos. El primer grupo ha sido diseñado para estudiar los parámetros involucrados en los métodos CG/SD. El segundo grupo investiga el papel de la prolongación a la frontera relativa de la columna generada en la eficiencia de los métodos CG/SD. El tercer experimento es una indagación en el campo de la elaboración de métodos CG/SD con regiones factibles no poliedrales. El cuarto tipo es una nota teórica en el campo de la elaboración de algoritmos CG/SD con los RMP no restringidos linealmente. El último experimento es una comparativa entre los métodos de descomposición simplicial con generación de columnas lineales con los de columnas no lineales.

Para realizar esta investigación, hemos considerado dos tipos de problemas test: el primero es el problema de flujo en redes no lineales uniproducto con restricciones de capacidad (SNFP), del que se han considerado varias redes de grandes dimensiones con diferentes grados de no linealidad y capacidad total; el segundo problema es el modelo de asignación de tráfico con modos combinados (TAP-M) con costes simétricos desarrollado en el capítulo 1.

Palabras clave: Descomposición simplicial, generación de columnas no lineales, cara óptima, aproximación interior, programación convexa diferenciable, problema de flujos en redes.

3.1 Introducción

La forma clásica de la descomposición simplicial fue primeramente descrita por Holloway [128] y Von Hohenbalken [127] para problemas no lineales con restricciones lineales. Este algoritmo usa una aproximación lineal en el CGP, y la aproximación interior en el RMP se obtiene mediante la envoltura convexa de un subconjunto de puntos extremos generados anteriormente. Hearn y otros [123, 125] introducen una mejora en este esquema basada en mantener el número de puntos extremos retenidos en el RMP inferior a una cantidad positiva r; cuando esta cantidad es alcanzada, el nuevo punto extremo reemplaza a la columna con menor peso en la combinación convexa de la última solución del RMP y se almacena la solución del último RMP como una columna individual. Este método se denomina *descomposición simplicial restringida* (RSD). Una extensión del RSD fue realizada por Larsson y otros [154, 141], la llamada *descomposición simplical no lineal* (NSD). El NSD se obtiene reemplazando el subproblema lineal de generación de columnas en el CGP por un subproblema convexo (no lineal). Este subproblema constituye una mejor aproximación al problema original que la obtenida en el SD.

El SD también ha sido extendido a problemas con restricciones no lineales. Ventura y Hearn [232] abordan la adaptación del RSD a problemas con restricciones convexas (RSDCC). El CGP es convertido a un problema lineal mediante la aproximación lineal a trozos de las restricciones no lineales, tal como en el algoritmo de Topkis-Veinott [228]. El NSD de Larsson y otros [154, 141] se aplica directamente sobre una región factible convexa, sin recurrir a una linealización de esta región.

Una combinación de la programación cuadrática secuencial (SQP) y del NSD se desarrolla en Patriksson [197]. En este método, los problemas CGP reemplazan las restricciones no lineales con aproximaciones lineales y su curvatura se tiene en cuenta, en la función objetivo, a través de la información derivada de las variables duales.

La clase de métodos de descomposición simplical también ha sido extendida al problema más general de las *desigualdades variacionales* (VIP) (ver por ejemplo la excelente monografía sobre el tema de Patriksson [197]) y a la programación matemática convexa *no diferenciable*. Larsson y otros [139, 142].

En este capítulo extenderemos los métodos CG/SD al problema general de optimización

$$\underset{x\in X}{\text{minimize}}\ f(\mathbf{x}), \qquad [\mathrm{P}(f,X)]$$

donde el conjunto $X \subset \Re^n$ es no vacío y compacto, y la función $f : X \mapsto \Re$ es continua en X.

La diferencia sustancial con los anteriores métodos de descomposición simplicial radica en la definición del CGP. En los esquemas anteriores el CGP se interpreta como una aproximación al problema original. El nuevo esquema CG/SD construye las columnas resolviendo aproximadamente el problema original a través de la realización de un número de iteraciones de algún algoritmo eficiente. En este contexto, el énfasis se sitúa en la elección de los algoritmos empleados en el CGP y no en la forma de aproximar el problema original.

Los algoritmos CG/SD pueden ser vistos como algoritmos modulares de programación matemática no lineal, donde se dispone de un algoritmo para resolver eficientemente el RMP y de otro para resolver el problema original. Este último algoritmo es el procedimiento para obtener las columnas en el CGP. Esta clase puede ser interpretada como un principio para acelerar la convergencia de un algoritmo convergente de puntos factible (algoritmo empleado en el CGP) mediante un esquema de descomposición simplicial.

En este capítulo hemos elegido una formulación más general que la usada en el capítulo 2, focalizada más en las propiedades de las sucesiones obtenidas en el CGP y en el RMP que en la de los algoritmos con las que son obtenidas. Hemos adoptado esta formulación, no por su mayor generalidad, sino porque enfatiza la interpretación del CG/SD como un procedimiento de aceleración de algoritmos.

Más concretamente, consideramos algoritmos del siguinete tipo

Definición 3.1.1 (Algortimos factibles convergentes). Sea $Y \subset X$ un conjunto no vacío y sea $\mathcal{A} : \widehat{Y} \mapsto 2^{\widehat{Y}}$ una aplicación multievaluda sobre Y. Diremos que la aplicación A es un algoritmo

factible convergente para resolver $\mathrm{P}(f, Y)$ sii cumple

(a) Es de puntos factibles, esto es , si $\mathbf{y} \in Y$ entonces $\mathcal{A}(\mathbf{y}) \subset Y$.

(b) Es convergente para cualquier punto inicial $\mathbf{y}^0 \in Y$. Es decir, si consideramos la sucesión generada por $\mathbf{y}^{t+1} \in \mathcal{A}(\mathbf{y}^t)$, entonces cualquier punto de acumulación es solución de $\mathrm{P}(f, Y)$. □

HIPÓTESIS 3.1.2 (Algoritmos factibles convergentes en el CG/SD). *Asumiremos que los algoritmos $\mathcal{A}_c$ y $\mathcal{A}_r$ satisfacen las siguientes propiedades*

(a) El algoritmo $\mathcal{A}_c$ es factible convergente para el problema $P(f, X)$.

(b) El algoritmo $\mathcal{A}_r$ es factible convergente para todos los problemas maestros restringidos de la forma

$$\underset{x \in \hat{X}}{\text{minimize}}\ f(\mathbf{x}), \qquad [\mathrm{RMP}(f, \hat{X})]$$

donde $\hat{X}$ es un subconjunto compacto de X tal que $\{\mathbf{x}, \hat{\mathbf{y}}\} \subset \hat{X}$, siendo $\hat{\mathbf{y}}$ la columna generada anteriormente.

En la tabla 3.1 resumimos los diferentes pasos de los métodos CG/SD. En este capítulo asumimos las reglas dadas en la tabla 2.2 para definir el conjunto X^t. Las hipótesis 3.1.2 garantizan que en un número finito de iteraciones se produce un descenso de la función objetivo, tanto en el CGP como en el RMP.

Tabla 3.1: El algoritmo CG/SD generalizado

0. (*Inicialización*): Elegir un punto inicial $\mathbf{x}^0 \in X$, y tomar $t := 0$.
1. (*Fase de generación de columnas*): Aplicar varias iteraciones del algoritmo $\mathcal{A}_\mathrm{c}$, comenzando desde $\mathbf{x}^t$, de modo que se obtenga un descenso de la función objetivo. El número de iteraciones es denotado por n_c^t. Denotemos el punto resultante por $\hat{\mathbf{y}}^t$.
2. (*Criterio de terminación*): Si $\mathbf{x}^t$ resuelve el $\mathrm{P}(f, X)$ entonces parar. En caso contrario continuar.
3. (*Conjunto de aumento*): Sea $X^{t+1} \subset X$ un conjunto compacto de modo que los puntos $\mathbf{x}^t, \hat{\mathbf{y}}^t$ pertenecen al conjunto X^{t+1}, esto es $\{\mathbf{x}^t, \hat{\mathbf{y}}^t\} \subseteq X^{t+1}$.
4. (*Problema maestro restringido*): Realizar varias iteraciones del algoritmo $\mathcal{A}_\mathrm{r}$ para resolver $\mathrm{CDP}(f, X^{t+1})$, comenzando desde $\hat{\mathbf{y}}^t$, de modo que se produzca un descenso en el valor de la función objetivo. Sea el punto resultante $\mathbf{x}^{t+1}$.
5. (*Actualizar*): Sea $t := t + 1$. Ir al paso 1.

El algoritmo descrito en el capítulo 2 se sitúa dentro de este marco, debido a que los algoritmos cerrados de descenso convergen a una solución (global) de un problema convexo. Por hipótesis, el $\mathrm{CDP}(f, X)$ y, por construcción, el $\mathrm{RMP}(f, X)$ son problemas convexos. Las hipótesis 2.2.1 garantizan, junto a la convexidad de los problemas $\mathrm{CDP}(f, X)$ y de los $\mathrm{RMP}(f, \hat{X})$ que los algoritmos que satisfacen las hipótesis 2.2.1 también cumplen las hipótesis 3.1.2. En este trabajo hemos omitido el análisis de la convergencia siendo ésta una línea de investigación futura.

La *descomposición simplicial* clásica puede rescatarse en este esquema considerando problemas de optimización donde X es un conjunto poliedral, $\mathcal{A}_\mathrm{c}$ es el algoritmo de Frank-Wolfe y solamente se realiza una iteración. Los métodos NSD desarrollados en Larsson y otros [154, 141] pueden ser descritos como métodos CG/SD donde $\mathcal{A}_\mathrm{c}$ realiza una única iteración de un método truncado de Newton. El RSDCC puede ser obtenido considerando X un conjunto convexo general y $\mathcal{A}_\mathrm{c}$ el algoritmo de Topkis-Veinott para el que se realiza solamente una iteración.

3.1.1 Motivaciones

El estudio teórico de la convergencia asintótica y finita de los algoritmos, realizado en el capítulo anterior, no contesta a todas las cuestiones que se plantean en las aplicaciones. Este capítulo contiene

un estudio computacional complementario de la eficiencia de los métodos CG/SD, en función de los elementos empleados en su definición. Más concretamente, un algoritmo de la clase CG/SD está caracterizado por las siguientes elecciones.

- En la definición de la fase de generación de columnas (CGP):
  - ◇ El algoritmo $\mathcal{A}_{\rm c}$ empleado en el CGP.
  - ◇ El número de iteraciones que $\mathcal{A}_{\rm c}$ efectúa en la iteración t. Este número es denotado por n_c^t. Si el parámetro es constante a través de todas las iteraciones, será denotado por n_c.
- En el RMP éstas son:
  - ◇ El algoritmo $\mathcal{A}_{\rm r}$ empleado en el RMP.
  - ◇ El número de iteraciones del algoritmo $\mathcal{A}_{\rm r}$ que son efectuadas en la iteración t, denotado por n_r^t. Si el parámetro es constante en todas las iteraciones, será denotado por n_r.
  - ◇ El conjunto X^t. La definición de la región factible del RMP se basa en un conjunto de reglas. Si se emplea un esquema de descomposición simplicial restringida se debe especificar el valor del parámetro r.

Hemos introducido la siguiente notación para reflejar todas las elecciones que definen un algoritmo específico de la clase CG/SD

$$(\{\mathcal{A}_{\rm r}\}_r^{n_r^t}, {\mathcal{A}_{\rm c}}^{n_c^t})$$

En este trabajo podremos abreviar la notación porque siempre se usa el mismo algoritmo para resolver el RMP, y por tanto, no es necesario que se especifique explícitamente. Consideraremos la notación

$$\{\mathcal{A}_{\rm c}\}_r^{n_r^t, n_c^t}$$

Esta notación se puede simplificar en el caso que en todas las iteraciones se aplique el mismo número de veces los algoritmos $\mathcal{A}_{\rm c}$ y $\mathcal{A}_{\rm r}$, en dicho caso emplearemos la notación

$$\{\mathcal{A}_{\rm c}\}_r^{n_r, n_c}$$

Los siguientes cuatro aspectos de la clase CG/SD son estudiados en este capítulo.

Parámetros de la clase CG/SD

Se podrían elaborar problemas de optimización convexa diferenciable cuya solución se alcanzase en el punto $\mathbf{x}^* := (\frac{n-2}{n^2} \dots \frac{n-2}{n^2}) \in \Re^n$ y cuya región factible fuese el $X := \{\mathbf{x} \in \Re^n : \mathbf{0} \leq \mathbf{x} \leq \mathbf{1}, \mathbf{1}^T\mathbf{x} \leq 1 - \frac{1}{n}\}$ para $n \geq 3$. El mínimo número de puntos extremos para expresar $\bar{\mathbf{x}}$ como combinación convexa de puntos extremos de X es n. Esto implica que si empleásemos el RSD o el SD deberíamos realizar un mínimo de n iteraciones principales e ir aumentando en una variable el tamaño del RMP en cada iteración. Por otro lado si utilizásemos un algoritmo CG/SD podríamos elegir cualquier punto de la frontera, no necesariamente un punto extremo, y el tamaño del RMP podría ser de solamente dos columnas. Por ejemplo, podemos considerar $\mathbf{x}^* := \frac{n-2}{n-1}(\frac{n-1}{n^2}, \cdots, \frac{n-1}{n^2}) + \frac{1}{n-1}(0, \cdots, 0)$. Esto es un indicio de un mecanismo que posibilita a los métodos CG/SD no lineales ser menos sensibles a la dimensión de F^* que la descomposición simplicial lineal y, por tanto, más eficientes para este tipo de problemas. Larsson y otros [154, 141] observaron este hecho en el método NSD al aplicarlo en los experimentos numéricos realizados para problemas de flujos en redes.

Estas consideraciones sobre el modo en que los métodos de descomposición no lineales pueden mejorar la eficiencia de los métodos simpliciales lineales, nos conducen a estudiar el papel de los parámetros n_r^t, r y n_c^t; así como sus interacciones, en la eficiencia de los métodos.

Hemos desarrollado una herramienta adaptativa para decidir en cada iteración t el valor de n_c^t. Esta regla establece un compromiso entre el coste computacional en la generación de las columnas y la eficiencia computacional del método CG/SD.

Metodología para la elaboración de algoritmos de tasa de convergencia superlineal para problemas de grandes dimensiones. El papel de la prolongación de la columna a la frontera relativa

La experiencia con el método RSD ha mostrado que inicialmente efectúa grandes progresos hasta alcanzar un entorno de la solución óptima, especialmente cuando se emplean grandes valores del parámetro r y cuando el algoritmo de resolución del RMP tiene convergencia superlineal (por ejemplo, cuadrática). Por contra, la convergencia se vuelve lenta cerca de la solución óptima. Este comportamiento se explica por el papel que juega el parámetro r, (máximo número de columnas retenidas en el RMP), en la eficiencia del algoritmo. Si $r \geq \mathtt{dim}\, F^* + 1$, donde F^* es la cara óptima, entonces la tasa de convergencia local del algoritmo esta monitorizada por la tasa de convergencia del método elegido para la resolución del RMP. Por tanto, se puede alcanzar una tasa de convergencia superlineal o cuadrática si se emplea un método (proyectado) de Newton. En caso contrario, $r < \mathtt{dim}\, F^* + 1$, el algoritmo sólo posee convergencia asintótica y la tasa de convergencia es la misma que el algoritmo de *Frank–Wolfe*, que es una convergencia sublineal. Este resultado también justifica la menor eficiencia en problemas con valores grandes de $\mathtt{dim}\, F^*$, que requieren de grandes valores de r y, por tanto, el número y el tamaño de los problemas RMP llegan a ser excesivamente grandes para poder ser resueltos con un coste computacional moderado.

Larsson y otros [154, 141] investigaron el problema de transporte estocástico (STP) y el problema de asignación de tráfico (TAP). Ambos problemas se diferencian sustancialmente en algunas propiedades numéricas: los STP empleados son altamente no lineales y la aproximación cuadrática al problema original empleada en el CGP puede ser resuelta analíticamente; los problemas TAP empleados son medianamente no lineales y los CGP son imposibles de resolver analíticamente, sólo pueden ser resueltos aproximadamente.

La mejor implementación de los algoritmos NSD reduce el tiempo de CPU de la mejor implementación del método RSD para los problemas STP entre $10 - 50$ veces, mientras que para el TAP es menor de 2. Este hecho podría ser explicado por la gran diferencia entre niveles de no linealidad en los problemas, por la $\dim F^*$, y/o por la elección del algoritmo para la resolución de los subproblemas. En este trabajo añadimos un nuevo elemento para intentar explicar este hecho: la prolongación de las columnas a la frontera relativa. Conjeturamos, basándonos en la experiencia computacional desarrollada, que:

> *si no se efectúa la prolongación de las columnas a la frontera relativa, entonces la tasa de convergencia del algoritmo CG/SD es la misma que el algoritmo $\mathcal{A}_c$ empleado en el CGP. En caso contrario, y si $r \geq \dim(F^*) + 1$, la tasa de convergencia es la misma que la del algoritmo $\mathcal{A}_r$.*

Esto explicaría la diferencia de eficiencia de los algoritmos NSD aplicados al TAP y STP, debido a que la prolongación no es calculada en el TAP, pero si para el STP.

Por tanto, para elaborar algoritmos eficientes para problemas de grandes dimensiones con convergencia superlineal, no basta con elaborar algoritmos menos sensibles a la dimensión de la cara óptima, ($\mathtt{dim}\, F^*$), tal como lo hace el NSD, sino que es necesario calcular la prolongación de la columna a la frontera relativa. Esta es la principal motivación para elaborar fórmulas para calcular dicha prolongación. En este trabajo contestamos a dicha cuestión cuando las columnas son generadas con ciertos métodos de direcciones factibles.

El papel de X^t

La clase de algoritmos CG/SD permite regiones factibles más generales para el RMP que en el SD, RSD o NSD. Una línea de investigación podría estudiar la eficiencia computacional de problemas RMP no restringidos linealmente, por ejemplo, considerar regiones factibles de la forma $\hat{X} = \mathtt{aff}\,(\hat{\mathcal{P}}) \cap X$. En este trabajo analizamos teóricamente esta posibilidad.

Generalización de los métodos de puntos factibles

Una motivación fundamental de este trabajo es mostrar como se puede acelerar la convergencia de un método de direcciones factibles de descenso mediante un esquema CG/SD. Este algoritmo es elegido para generar las columnas en el CGP, es decir, tomándolo como $\mathcal{A}_c$. Hay dos formas de efectuar esta aceleración:

- ◇ Búsquedas multidimensionales. Este procedimiento efectúa varias iteraciones del algoritmo, prolonga la columna obtenida a la frontera relativa y la introduce en el RMP. Este nuevo RMP conduce a un nuevo punto donde volver a iniciar el algoritmo $\mathcal{A}_c$. Esta extensión generaliza los algoritmos de búsquedas unidmimensionales a búsquedas multidimensionales en los conjuntos X^t. La ventaja de las búsquedas unidimensionales es que son fácilmente realizables, no obstante la estructura de los RMP hace que se puedan resolver con un bajo coste computacional.

- ◇ Generalización del procedimiento de las tangentes paralelas (PARTAN). Esta extensión se basa en tomar $r = 1$ en RMP. Ello produce una nueva búsqueda lineal después de realizar n_c^t búsquedas lineales en la fase CGP. La nueva dirección está definida por la última solución del RMP y la columna generada en el CGP. Esta modificación tiene una fácil implementación numérica, debido a que el algoritmo empleado para efectuar las búsquedas unidimensionales en el CGP, puede ser usado para resolver el RMP y, por tanto, no requiere de un algoritmo especial para resolver el RMP.

  El nombre dado a este procedimiento tiene su origen en el hecho de que cuando se realizan solamente dos iteraciones del algoritmo $\mathcal{A}_c$, este método es el procedimiento de las tangentes paralelas PARTAN.

En este trabajo hemos considerado ambas extensiones. Los algoritmos de direcciones factibles de descenso que hemos empleado son: Frank-Wolfe y Evans. Estas mejoras se han comparado con el SD, RSD y NSD.

3.2 Aplicaciones de la clase CG/SD

En esta sección describimos los problemas de prueba y los algoritmos empleados en su resolución.

3.2.1 Problemas de prueba

Hemos aplicado los métodos CG/SD a un problema uniproducto de flujos no lineales en redes y al TAP-M. Existen varias razones para elegir estas aplicaciones.

- ◇ Las aproximaciones lineales en el CGP son fácilmente resolubles. La aproximación lineal del problema uniproducto es el denominado *problema de flujo en redes de coste mínimo*; la linealización del TAP-M se transforma en una colección de *problemas de caminos mínimos* sobre las redes A y B, y sobre estos caminos, y sus caminos combinados, se calcula el de menor coste *extendido* (ver capítulo 1).

- ◇ La principal motivación para elegir el problema uniproducto es que siempre se puede calcular la prolongación de las columnas a la frontera relativa. La posibilidad de calcular esta prolongación, en el problema multiproducto, dependerá del algoritmo elegido en el CGP.

Problema de flujo en redes uniproducto (SNFP)

Considerar un grafo dirigido $(\mathcal{N}, \mathcal{A})$ con n nodos y m arcos. Para cada nodo $i \in \mathcal{N}$ se da un escalar s_i que es la cantidad de producto que es atraído o generado en el nodo i, y para cada arco $(i, j) \in \mathcal{A}$

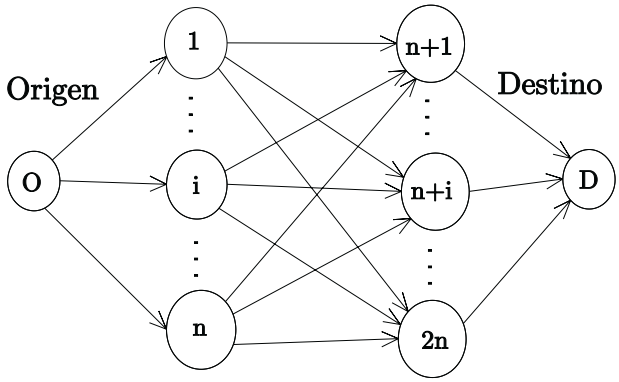


Figura 3.1: Topología de los problemas del tipo AUT

se considera una función continuamente diferenciable y convexa f_{ij}. El problema de flujo en redes no lineales uniproducto con costes separables se formula por

$$\begin{array}{ll} \text{minimizar} & Z = \sum_{(i,j)\in\mathcal{A}} f_{ij}(x_{ij}) \\ \text{sujeto a} & \sum_{\{j:(i,j)\in\mathcal{A}\}} x_{ij} - \sum_{\{j:(j,i)\in\mathcal{A}\}} x_{ji} = s_i, \quad \forall i \in \mathcal{N} \\ & 0 \le x_{ij} \le u_{ij}, \quad \forall (i,j) \in \mathcal{A} \end{array} \qquad \text{[SNFP]}$$

donde la variable x_{ij} se denomina *flujo* del arco (i,j) y el vector $\mathbf{x} = \{x_{ij} : (i,j) \in \mathcal{A}\}$ es el *vector de flujo*. Los números reales u_{ij} son las capacidades de los arcos (i,j).

Hemos creado tres tipos de problemas de prueba. Los problemas del primer y del segundo tipo fueron creados empleando los generadores de redes de dominio público NETGEN, escrito por Klingman y otros [136], que genera problemas lineales de transporte, asignación y transbordo; y GRIDGEN, escrito por Bertsekas [19], que construye problemas aleatorios con una estructura de rejilla bidimensional. A esta red se le añaden aleatoriamente arcos. El tercer tipo de problemas ha sido creado por nosotros. La topología de la red consta de un grafo bipartido completo $n \times n$, más dos conjuntos de arcos. El primer conjunto conecta un nodo origen con el primer conjunto de nodos y el otro conjunto de arcos conecta los nodos de la segunda componente del grafo bipartido con el nodo destino. Un ejemplo de estos grafos está recogido en la figura 3.1. Los problemas de prueba se nombrarán como: NET, GRI y AUT.

Además de la topología de la red hemos generado las funciones que producen los costes en los arcos. Las expresiones funcionales de estos costes son:

(1) $f_{ij}(x_{ij}) = a_{ij}(x_{ij} + 0.2b_{ij}(x_{ij}^5/c_{ij}^4))$

(2) $f_{ij}(x_{ij}) = a_{ij}x_{ij}^3 + b_{ij}x_{ij}^2 + c_{ij}x_{ij}$

(3) $f_{ij}(x_{ij}) = a_{ij}x_{ij}^2 + b_{ij}x_{ij}$

(4) $f_{ij}(x_{ij}) = a_{ij}x_{ij}\,(\log(x_{ij}/b_{ij}) - c_{ij})$

donde a_{ij}, b_{ij}, c_{ij} son parámetros.

La tabla 3.2 muestra la descripción de los problemas de prueba. Los problemas NET3a y NET4a son estrictamente convexos mal condicionados, tal y como son considerados en Bertsekas y otros [22]. Estos problemas fueron creados asignando a algunos arcos un valor positivo muy pequeño, en comparación a los otros arcos, al coeficiente cuadrático. El 50% de los arcos tienen un coeficiente 1 y para el otro 50% este coeficiente ha sido generado mediante una distribución uniforme en $[5, 10]$. Cuando los arcos tienen esta estructura suele ocurrir el problema de mal acondicionamiento, en el sentido clásico de la programación matemática sin restricciones.

Los problemas NET3b y NET4b tienen costes lineales y cuadráticos. Los costes lineales se han obtenido reemplazando los coeficientes cuadráticos pequeños de las redes NET3a y NET4a por cero.

Los problemas generados con GRIDGEN tienen dos clases de arcos: el conjunto de arcos de la rejilla (que definen un grafo tipo *Manhattan*) y el conjunto de arcos aleatorios. El coste de los arcos de la rejilla es *MAXCOST* y los coeficientes para el resto de las funciones de coste han sido generados mediante una distribución uniforme en el intervalo definido por *MINCOST* y *MAXCOST*. Debido al alto coste de los arcos de la rejilla hay un incentivo en abandonarlos. La capacidad de cada arco de la rejilla es la demanda total y esta elección garantiza la factibilidad del problema. La capacidad de los arcos aleatorios se genera mediante una variable aleatoria uniforme. Hemos generado dos versiones de la misma red, la primera tiene muchos arcos aleatorios, por lo que fácilmente se puede evitar la rejilla y la otra contiene mucho menor número de arcos aleatorios. El nombre de los problemas de la primera versión finaliza en *a* y los otros en *b*.

El tercer tipo de problemas (AUT) ha sido diseñado para conseguir problemas donde la solución del problema tenga la mayor parte de sus arcos a su capacidad máxima. Hemos distinguido entre los arcos que unen el nodo origen y destino del resto de arcos. La capacidad en cada uno de estos arcos es igual a la demanda total de flujo entre el nodo origen y el nodo destino. La suma de la capacidad de todos los arcos interiores es denominada MAXFLOW y ésta es distribuida entre ellos del siguiente modo. Se han generado las variables aleatorias X_i, e Y_{ij} para todo $i,\ j\ = 1, \ldots, n$ uniformemente distribuidas en $(0, 1)$, entonces la capacidad de un arco interior (i, j) se calcula por

$$u_{ij} = \frac{X_i}{\sum_{j=1}^{n} X_j} \frac{Y_{ij}}{\sum_{k=1}^{n} Y_{ik}} \text{ MAXFLOW.}$$

El grado de saturación de la red está monitorizada por la demanda total en relación al parámetro MAXFLOW, que constituye *una cortadura de flujo* en la red.

Tabla 3.2: Descripción de los problemas de prueba

Problema	Nodos	Arcos	Func.	a_{ij}	b_{ij}	c_{ij}	u_{ij}	Demanda
NET1a	500	2500	(1)	$[1-10]$	$[1-10]$	50	35	$(250^a, 250^b, 5000^c)$
NET1b	500	2500	(2)	$[1-50]$	$[1-10]$	25	35	$(250, 250, 5000)$
NET1c	500	2500	(3)	$[1-10]$	$[1-10]$	50	35	$(250, 250, 5000)$
NET2a	1000	5000	(1)	$[1-10]$	$[1000-10000]$	$[5-25]$	15	$(500, 500, 5000)$
NET2b	1000	5000	(2)	$[1-50]$	$[1-10]$	25	15	$(500, 500, 5000)$
NET3a	200	1300	(3)	$[5-10]$ or 1	$[1-100]$	0	$[100-300]$	$(1, 1, 10000)$
NET3b	200	1300	(3)	$[5-10]$ o 0	$[1-100]$	0	$[100-300]$	$(1, 1, 10000)$
NET4a	400	4500	(3)	$[5-10]$ o 1	$[1-100]$	0	$[100-300]$	$(1, 1, 10000)$
NET4b	400	4500	(3)	$[5-10]$ o 0	$[1-100]$	0	$[100-300]$	$(1, 1, 10000)$
GRI1a	100	3000	(1)	$[1-50]$	20	1	$[3-5]$	$(1, 1, 100)$
GRI1b	100	1000	(1)	$[1-10]$	20	1	$[1-3]$	$(1, 1, 100)$
GRI2a	100	3000	(2)	1	0.05	0	1	$(1, 1, 6000)$
GRI2b	100	1000	(2)	1	0.05	0	1	$(1, 1, 2000)$
GRI3a	100	3000	(4)	1	1	1	1	$(1, 1, 6000)$
GRI3b	100	1000	(4)	1	1	1	1	$(1, 1, 2000)$
AUT1	42	440	(4)	1	2	1	MAXFLOW =425	$(1, 1, 300)$
AUT2	102	2600	(4)	1	2	1	MAXFLOW =2514	$(1, 1, 100)$
AUT3	102	2600	(4)	1	0.5	1	MAXFLOW =2514	$(1, 1, 2400)$

[a] Número de sumideros (nodos con $s_i < 0$).
[b] Número de fuentes (nodos con $s_i > 0$).
[c] Oferta total.

La tabla 3.3 muestra la precisión requerida en la resolución de los problemas y el porcentaje de arcos que toman el valor de sus cotas en la mejor solución obtenida.

Tabla 3.3: Precisión requerida, y porcentaje de arcos en sus cotas en la solución (casi) óptima

Red	Precisión	% de arcos en su cota inferior	% de arcos en su cota superior	Red	Precisión	% de arcos en su cota inferior	% de arcos en su cota superior
NET1a	10^{-5}	79.48	0.40	NET1b	10^{-3}	43.04	0.04
NET1c	10^{-4}	55.40	0.00	NET2a	10^{-4}	40.24	0.04
NET2b	10^{-3}	43.48	0.06	NET3a	10^{-4}	71.38	0.54
NET3b	10^{-4}	85.84	0.77	NET4a	10^{-4}	83.69	0.09
NET4b	10^{-4}	93.26	0.13	GRI1a	10^{-3}	80.36	0.90
GRI1b	10^{-3}	12.40	1.10	GRI2a	10^{-2}	23.36	14.06
GRI2b	10^{-2}	23.22	13.30	GRI3a	10^{-2}	15.30	39.53
GRI3b	10^{-2}	9.42	23.35	AUT1	10^{-4}	0.00	15.26
AUT2	10^{-3}	0.00	0.00	AUT3	10^{-5}	0.00	73.88

Redes en equilibrio con modos combinados (TAP-M)

El segundo tipo de problema es el modelo de equilibrio con modos combinados con costes simétricos, desarrollado en el capítulo 1. Las redes de prueba se describieron en el capítulo 1 (ver tabla 1.4). Los parámetros empleados para el modelo logit se muestra en la tabla 3.4

Tabla 3.4: Parámetros logit para las redes de prueba para el TAP-M

Problema	β_1	β_2	θ_a	θ_b	τ_ω	α_t^c	α^k
NgD2	2.00	4.00	1.0	1.0	1.0	1.0	1.0
SiF2	1.00	1.20	1.0	1.0	1.0	1.0	1.0
Hul2	1.00	1.50	1.0	1.0	1.0	1.0	1.0

3.2.2 Algoritmos CG/SD empleados en la experiencia computacional

Algoritmos para el CGP

Hemos considerado los siguientes algoritmos para $\mathcal{A}_c$

(a) Descomposición simplicial restringida (Hearn y otros [125]), que es denotada por $\mathcal{A}_c$ =RSD($\tilde{r}$) donde $\tilde{r}$ es el parámetro empleado en el problema maestro restringido. Este algoritmo se transforma en el de Frank-Wolfe [88] cuando $\tilde{r} = 1$. Este caso especial es denotado por $\mathcal{A}_c$ =FW. Cuando $\tilde{r} = \infty$ se obtiene la descomposición simplicial original (Holloway [128], Von Hohenbalken [127]), denotada por $\mathcal{A}_c$ =SD.

(b) Algoritmo de Evans [72]. Este algoritmo ha sido desarrollado para modelos combinados de asignación en equilibrio donde la función objetivo es separable, esto es $f(\mathbf{x}_1, \mathbf{x}_2) = f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2)$, donde la función $f_1(\mathbf{x}_1)$ está asociada a los costes de transporte mientras que $f_2(\mathbf{x}_2)$ lo está con el modelo de demanda.

El algoritmo es un método de direcciones factibles de descenso. La dirección de búsqueda en $(\mathbf{x}_1, \mathbf{x}_2)$ viene definida por la solución del problema de optimización

$$\min_{(\mathbf{y}_1, \mathbf{y}_2) \in X} f_1(\mathbf{x}_1) + \nabla^T f_1(\mathbf{x}_1)(\mathbf{y}_1 - \mathbf{x}_1) + f_2(\mathbf{y}_2).$$

El algoritmo, tras resolver el problema anterior, realiza una búsqueda unidimensional en la dirección $\mathbf{d} = \mathbf{y}^* - \mathbf{x}$, donde $\mathbf{y}^*$ es la solución obtenida. Hemos denotado este algoritmo por $\mathcal{A}_c$ =E.

Los algoritmos de Evans y de Frank-Wolfe tienen, esencialmente, el mismo coste computacional para obtener las direcciones de descenso, sin embargo, el algoritmo de Evans posee una mejor eficiencia computacional, además de proveer mejores cotas inferiores a la solución del problema. En el trabajo de García y otros [103] se muestra como obtener cotas inferiores del problema de optimización con la clase de algoritmos de linealización parcial, en particular, con los algoritmos de Frank-Wolfe y de Evans.

La fase CGP, para la clase NSD, está definida mediante la realización de n iteraciones de un algoritmo cerrado de descenso A sobre el problema auxiliar

$$\min_{\mathbf{y}\in X} \Pi(\mathbf{y}, \mathbf{x}), \qquad [\mathrm{CGP}(\Pi, X, \mathbf{x})]$$

donde $\Pi(\cdot, \mathbf{x}) : X \mapsto \Re$ es una aproximación de f en el punto $\mathbf{x}$. La combinación de ambas elecciones, por un lado el algoritmo de resolución y por otro, la aproximación del problema, definen el método de generación de columnas $\mathcal{A}_c = (\Pi, A)$. En la clase NSD solamente se efectúa una iteración con el algoritmos $\mathcal{A}_c$. A continuación listamos los diferentes métodos NSD que se han empleado en este capítulo.

(c) Método truncado de Newton, donde la aproximación está definida por

$$\Pi_N(\mathbf{y}, \mathbf{x}) = f(\mathbf{x})^T\mathbf{y} + (1/2)\mathbf{y}^T\nabla^2 f(\mathbf{x})\mathbf{y} \tag{3.1}$$

y se realizan n iteraciones del algoritmo RSD($\tilde{r}$). Este algoritmo lo hemos denotado por N($\tilde{r}, n$)

(d) Método de Goldstein-Levitin-Polyak's (Goldstein [112], Levitin y Polyak [150]), que está definido por el subproblema

$$\Pi_{GLP}(\mathbf{y}, \mathbf{x}) = \nabla f(\mathbf{x})^T\mathbf{y} + (\gamma/2)\mathbf{y}^T\mathbf{y} \tag{3.2}$$

donde $\gamma > 0$ y se realizan n iteraciones del algoritmo RSD($\tilde{r}$). Este algoritmo se denota por GLP($\tilde{r}, n$).

(e) Método de Newton-Evans. Si consideramos una función objetivo separable de la forma $f(\mathbf{x}_1, \mathbf{x}_2) = f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2)$, podemos aproximar la función por

$$\Pi_{NE}(\mathbf{y}_1, \mathbf{y}_2, \mathbf{x}_1, \mathbf{x}_2) = \nabla f_1(\mathbf{x}_1)^T\mathbf{y}_1 + (1/2)\mathbf{y}_1^T\nabla^2 f_1(\mathbf{x}_1)\mathbf{y}_1 + f_2(\mathbf{y}_2) \tag{3.3}$$

y resolver truncadamente este problema realizando n iteraciones del algoritmo de FW. Este algoritmo lo denotamos por NE(n) o simplemente NE.

(f) Método GLP-Evans. Si la función objetivo es de la forma $f(\mathbf{x}_1, \mathbf{x}_2) = f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2)$, definimos la aproximación del problema original por

$$\Pi_{GLPE}(\mathbf{y}_1, \mathbf{y}_2, \mathbf{x}_1, \mathbf{x}_2) = \nabla f_1(\mathbf{x}_1)^T\mathbf{y}_1 + (\gamma/2)\mathbf{y}_1^T\mathbf{y}_1 + f_2(\mathbf{y}_2) \tag{3.4}$$

donde $\gamma > 0$, y es resuelta truncadamente realizando n iteraciones del algoritmo FW. Este algoritmo es denotado por GLPE(n) o simplemente GLPE.

(g) Una modificación de los métodos NSD aplicados a problemas de flujos en redes multiproducto

Ahora analizaremos una modificación de los subproblemas cuadráticos de los métodos NSD, cuando éstos son aplicados a ciertos problemas de flujos en redes. Consideraremos que el NSD está siendo aplicado a problemas que conducen a la resolución de problemas de caminos mínimos. Ejemplos de estos problemas son el TAP o el TAP-M.

Tabla 3.5: Algoritmos empleados en el CGP

Problema	$\Pi(\mathbf{y},\mathbf{x})$	A	$\mathcal{A}_c$
SNFP	N, GLP	RSD	N$(\tilde{r},n)$, GLP$(\tilde{r},n)$ y RSD$(\tilde{r})$.
TAP-M	$\widehat{\text{NE}}$, $\widehat{\text{GLPE}}$	FW	$\widehat{\text{NE}}$, $\widehat{\text{GLPE}}$, FW y E.

La eficiencia de los métodos NSD radica en la aplicabilidad de los algoritmos RSD o Evans para resolver truncadamente la aproximación cuadrática. Por otro lado, estos algoritmos se pueden aplicar directamente a la resolución del problema original. Ambos problemas son de flujos en redes, separables, convexos y diferenciables; pero existe una diferencia sustancial entre la aplicación a ambos subproblemas. En el problema original, los subproblemas lineales son un problema de caminos mínimos con costes no negativos. Esto es debido a que el coste en cada arco es calculado como la derivada de la función de coste en el arco en el actual flujo, y como los costes son funciones crecientes tienen su primera derivada positiva. Este subproblema puede ser resuelto eficientemente con el algoritmo de Dijkstra [67]. Por contra, si los algoritmos son aplicados a los subproblemas cuadráticos del CGP, esta propiedad se pierde, como ilustra la figura 3.2. La aproximación cuadrática es convexa pero no necesariamente monótona creciente. Esta situación conduciría a emplear algoritmos de caminos mínimos para costes generales (que son menos eficientes).

Proponemos una modificación del problema cuadrático que evita esta desventaja. Supongamos que la función objetivo (la parte asociada con los costes) es separable, esto es, $f(\mathbf{x}) = \sum_{i\in I} f_i(\mathbf{x}_i)$ donde $f_i(x_i)$ es una función de una variable dependiente de x_i. La aproximación cuadrática de f en el punto $\mathbf{x}$ es $\Pi(\mathbf{y},\mathbf{x}) = f(\mathbf{x}) + \nabla f(\mathbf{x})(\mathbf{y}-\mathbf{x}) + 1/2(\mathbf{y}-\mathbf{x})H(\mathbf{x})(\mathbf{y}-\mathbf{x})$. Esta función es separable y puede ser expresada por $\Pi(\mathbf{y},\mathbf{x}) = \sum_{i\in I}\Pi_i(y_i,x_i)$ donde

$$\Pi_i(y_i,x_i) = f_i(x_i) + f_i'(x_i)(y_i-x_i) + 1/2H_i(x_i)(y_i-x_i)^2$$

donde $H_i(x_i) = f_i''(x_i)$ para la aproximación tipo Newton y $H_i(x_i) = \gamma$ para la aproximación de GLP.

Denotamos con $T_i(y_i,x_i)$ la ecuación de la recta tangente a la curva $\Pi(y_i,x_i)$ en el punto ω cumpliendo que $f_i(0) = T_i(0,x_i)$. Hemos reemplazado la aproximación $\Pi_i(y_i,x_i)$ por

$$\hat{\Pi}_i(y_i,x_i) = \begin{cases} \Pi_i(y_i,x_i), & y_i \geq w \\ T_i(y_i,x_i), & y_i < w \end{cases}$$

donde w es la abscisa del punto de tangencia. Este punto se calcula por

$$w(x_i) = \sqrt{\frac{2\left(f_i(x_i) - f_i'(x_i)x_i - f_i(0)\right)}{H_i(x_i)} + x_i^2}$$

Esta aproximación es convexa, diferenciable y creciente en $\Re$, tal como se ilustra en la figura 3.2.

Hemos añadido el símbolo $\hat{}$ encima de los nombres de los métodos empleados en el CGP cuando se usa esta modificación. En este trabajo hemos considerado los métodos $\widehat{\text{NE}}$, $\widehat{\text{GLPE}}$.

La tabla 3.5 resume los algoritmos empleados en la fase CGP en función del tipo de problema de prueba.

Para estar completamente definida la fase CGP, debemos especificar el número de iteraciones que realizamos con el algoritmo $\mathcal{A}_c$ en cada iteración, es decir, el valor de n_c^t. Para los algoritmos (c)-(g) hemos tomado $n_c^t = 1$ para todo t. Estos algoritmos son versiones del método truncado de Newton o

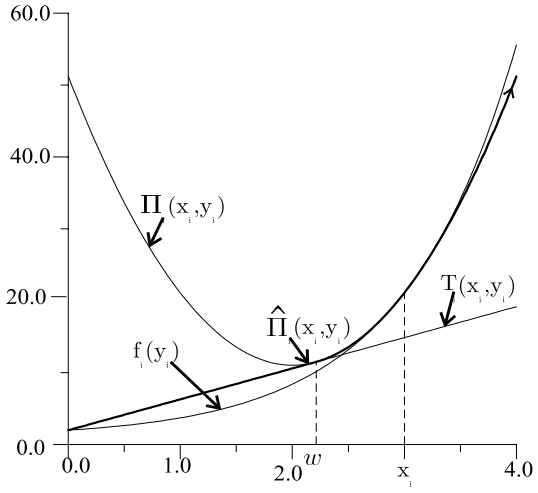


Figura 3.2: Aproximación monótona y diferenciable de la función de mérito

del algoritmo proyectado de Goldstein-Levitin-Polyak. Esta elección conduce a métodos CG/SD que también pertenecen a la clase NSD. Para los algoritmos (a) y (b) hemos empleado dos forma de definir la sucesión $\{n_c^t\}$. La primera emplea en todas las iteraciones el mismo valor del parámetro, esto es, $n_c^t = n_c \geq 1$ y la segunda se obtiene mediante la herramienta adaptativa que hemos desarrollado para tal fin. En cada iteración, en función de ciertos ratios obtenidos en la iteración anterior, se elige el valor para n_c^{t+1}.

Descripción del RMP

La región factible del RMP ha sido definida mediante las mismas reglas que la descomposición simplicial restringida (ver tabla 2.2).

Hemos empleado solamente el algoritmo proyectado de Newton (ver Bertsekas [17]) para resolver los RMP. Este algoritmo tiene convergencia superlineal y es ventajoso para la estructura del RMP. No hemos considerado otros algoritmos debido a que la clave de esta fase es la velocidad de convergencia del algoritmo $\mathcal{A}_r$. Si eligiésemos un algoritmo con convergencia sublineal, haría ineficiente el algoritmo CG/SD. Por contra, si empleásemos un algoritmo diferente con convergencia superlineal, el comportamiento del CG/SD sería similar, exceptuando (quizás) los tiempos empleados en la resolución del RMP.

El RMP está resuelto truncadamente y el nivel de exactitud está implícitamente definido a través del número de iteraciones (proyecciones) que aplicamos del método de Newton. Este número lo denotamos por n_r^t y este valor será siempre el mismo para todas las iteraciones y lo llamaremos n_r.

Notación

Como ya adelantamos hemos empleado la siguiente notación para los algoritmos CG/SD.

$$\mathcal{A}_{cr}^{n_r,n_c}$$

Observar que está notación define completamente un método CG/SD. Por ejemplo, $\mathrm{E}_{\infty}^{n_r,n_c}$ significa efectuar n_c iteraciones del algoritmo de Evans en la fase CGP para generar la columna y resolver truncadamente el problema RMP mediante n_r iteraciones del algoritmo proyectado de Newton donde el parámetro de restricción vale $r = \infty$. Esta elección conduce a un esquema de descomposición simplicial.

Otra observación es que el nombre del algoritmo $\mathcal{A}_c$ define unívocamente el tipo de problema de

prueba que se está resolviendo debido a que los problemas SNFP y TAP-M se resuelven con algoritmos distintos (ver tabla 3.5).

Esta notación nos permite definir el mismo algoritmo de diferentes formas. Por ejemplo, el algoritmo truncado de Newton o el algoritmo GLP pueden ser denotados por $\mathrm{N}(\tilde{r}, n)$ y $\mathrm{GLP}(\tilde{r}, n)$, sin embargo, la notación $\mathrm{N}(\tilde{r}, n)_1^{n_r,1}$ y $\mathrm{GLP}(\tilde{r}, n)_1^{n_r,1}$ definen los mismos algoritmos. Esto se debe a que ambos definen la misma dirección de búsqueda y el parámetro $r = 1$ transforma el RMP en un problema de búsqueda lineal. La búsqueda lineal realizada en el CGP se deshace cuando la columna se prolonga a la frontera. Otro ejemplo lo constituye la descomposición simplicial que puede ser denotada por SD, $\mathrm{RSD}(\infty, n)$, $\mathrm{FW}_{\infty}^{n_r,1}$, $\mathrm{RSD}(1,,1)_{\infty}^{n_r,1}$, etc.

3.2.3 Detalles de implementación

Detalles de la codificación de los algoritmos CG/SD para el problema SNFP

La tabla 3.6 muestra los parámetros γ empleados por el algoritmo GLP en cada problema de prueba. Este parámetro ha sido calibrado realizando cinco pruebas y tomando el valor que produce los mejores resultados computacionales.

Tabla 3.6: Valores del parámetro γ del algoritmo GLP

Red	γ	Red	γ	Red	γ	Red	γ
NET1a	3	NET1b	75	NET1c	10	NET2a	2000
NET2b	50	NET3a	50	NET3b	15	NET4a	50
NET4b	25	GRI1a	100	GRI1b	50	GRI2a	100
GRI2b	25	GRI3a	100	GRI3b	25	AUT1	75
AUT2	50	AUT3	200				

Se ha tomado la subrutina del método proyectado de Newton para resolver el RMP del código de Hearn y otros [121]. Este algoritmo incluye una búsqueda lineal del tipo Armijo y se deben especificar dos parámetros σ y β. Hemos empleado los valores de 10^{-4} y 0.5 respectivamente, los cuales son adecuados en muchos problemas (ver Bertsekas [17]).

Los algoritmos CG/SD, empleados en los SNFP, requieren del algoritmo RSD para generar las columnas o para resolver truncadamente la aproximación cuadrática del CGP. Hemos utilizado el código del RSD desarrollado para este tipo de problemas en Hearn y otros [121]. Este código emplea el algoritmo del simplex primal para resolver los problemas lineales, más concretamente usa el código NETFLOW, implementado en Kennington y Helgason [135], para resolver el problema de flujo en redes lineales de mínimo coste.

Todos los códigos se han elaborado en FORTRAN (Visual Workbench) y se han ejecutado con doble precisión sobre un ordenador PC de 64 megabytes de RAM a 200 Mhz.

Detalles de la codificación de los algoritmos CG/SD para el problema TAP-M

Los algoritmos de FW y E están definidos por dos fases: la resolución de un problema para definir la dirección de búsqueda (ver apéndices I y II del capítulo 1 para su definición y resolución) y un problema unidimensional en la dirección de búsqueda obtenida para determinar el avance en dicha dirección (tamaño del paso). La resolución de los subproblemas que definen la dirección de búsqueda requieren de la resolución de varios problemas de caminos mínimos. Para este fin, hemos empleado el algoritmo L2QUE de Gallo y Pallotino [92]. El algoritmo empleado en la búsqueda unidimensional es el de la *sección áurea*.

Los códigos desarrollados emplean la misma subrutina del algoritmo de FW, tanto para generar la columna, como para resolver aproximadamente el subproblema cuadrático para los métodos NSD.

En esta aplicación también empleamos el método de Newton proyectado de Bertsekas [17] para resolve el RMP. La subrutina es la misma que para el problema SNFP.

Los programas fuentes se han escrito en FORTRAN (Visual Workbench) y se ha empleado precisión simple en las operaciones de representación y precisión doble en aquellas operaciones que pudieran originar errores de redondeo. Por ejemplo, la precisión simple se emplea para resolver el problema de caminos mínimo y precisión doble para resolver el problema RMP, porque las proyecciones son muy sensibles a los errores de redondeo. Los códigos han sido ejecutados sobre un PC de 384 megabytes de RAM y 400 MHz.

3.3 Experimentos numéricos

Hemos diseñado cinco bloques de experimentos para investigar el comportamiento computacional de los métodos CG/SD. El primer bloque estudia el papel de los parámetros r, n_c y n_r sobre la eficiencia de la clase CG/SD. Como es demasiado costoso computacionalmente cubrir todas las posibles combinaciones de los parámetros, el procedimiento que hemos seguido fija dos parámetros y varía el otro para estudiar la eficiencia del algoritmo en función de él. El segundo bloque es una incursión en el campo del diseño de criterios dinámicos para determinar el número de iteraciones a realizar con $\mathcal{A}_c$ en cada iteración, es decir, obtener reglas para generar $\{n_c^t\}_{t\geq 1}$. Se describe el procedimiento y se evalúa numéricamente. El tercer bloque de experimentos estudia la prolongación de las columnas en la eficiencia del algoritmo CG/SD. El cuarto bloque es una nota teórica sobre la posibilidad de elaborar métodos con problemas RMP no restringidos linealmente. El último bloque de experimentos es una comparación entre los nuevos algoritmos que aparecen en la clase CG/SD con los algoritmos clásicos.

Bloque 1: estudio de los parámetros de la clase CG/SD

EXPERIMENTO 1.0: estrategias de truncamiento en los algoritmos NSD.

El primer experimento calcula el tiempo de CPU empleado por los algoritmos NSD: $\mathrm{N}(\tilde{r}, n)_\infty^{n_r,1}$ y $\mathrm{GLP}(\tilde{r}, n)_\infty^{n_r,1}$ en función de ciertas combinaciones de los parámetros $\tilde{r}$ y n de los algoritmos $\mathrm{N}(\tilde{r}, n)$ y $\mathrm{GLP}(\tilde{r}, n)$.

El parámetro n, número de iteraciones que se aplica al RSD para resolver los subproblemas, define la estrategia de truncamiento para los problemas cuadráticos empleados en el algoritmo NSD. El parámetro $\tilde{r}$ define diferentes algoritmos para resolver los subproblemas. Cuando $\tilde{r} = 1$ se obtiene el algoritmo de Frank-Wolfe, cuando $\tilde{r} = \infty$ aparece la descomposición simplicial y cuando $1 < \tilde{r} < \infty$ aparece propiamente el RSD.

Este experimento puede ser visto como un pre-experimento, con el objeto de obtener una estrategia óptima de truncamiento y un algoritmo especifico para resolver los subproblemas cuadráticos en el CGP.

Las redes que hemos considerado han sido: NET1a, NET2b, GRI3a y AUT2. Los valores elegidos para el parámetro n_r han sido $8, 8, 15$ y 15 para cada red, respectivamente.

La primera cuestión que se puede plantear, es la inicialización de los subproblemas cuadráticos. Hemos considerado dos opciones: la primera inicializa cada subproblema con los puntos extremos retenidos en el último RMP para resolver el problema cuadrático, la segunda inicialización comienza de nuevo en cada iteración. Computacionalmente hemos observado que la segunda inicialización es mejor que la primera y es la empleada en todas las pruebas.

La figura 3.3 muestra los resultados obtenidos por el algoritmo $\mathrm{N}(\tilde{r}, n)_\infty^{n_r,1}$ en dos redes de prueba y se ve que el mejor algoritmo es el FW (exceptuando valores pequeños de n en la red NET1a).

En la figura 3.4 aparecen los resultado obtenidos con el algoritmo $\mathrm{GLP}(\tilde{r}, n)_\infty^{n_r,1}$. El mejor algoritmo para resolver truncadamente los subproblemas cuadráticos es el FW. Además el RSD es muy

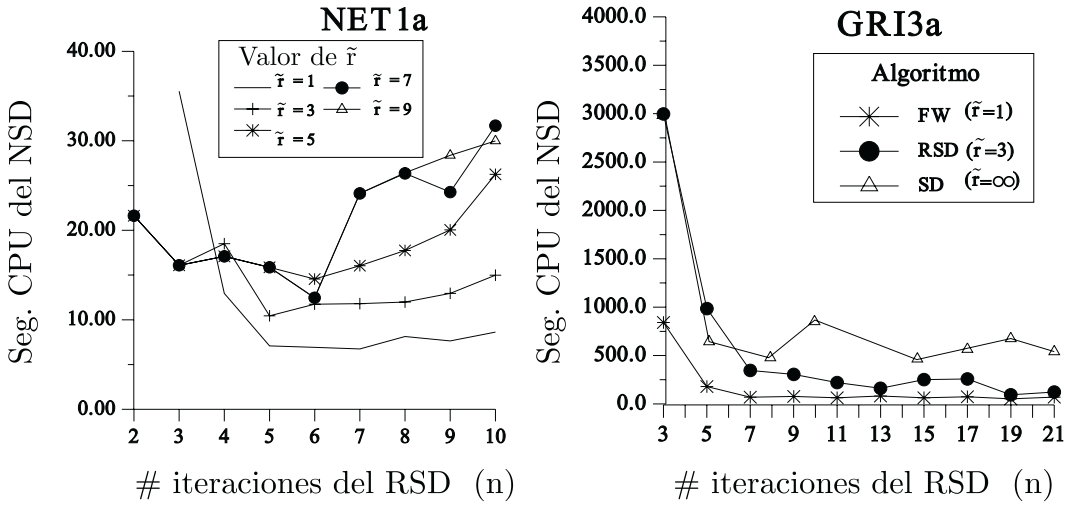


Figura 3.3: Eficiencia del uso del algoritmo RSD en la resolución de los problemas cuadráticos del método NSD (tipo Newton): estudio de los parámetros r̃ y n

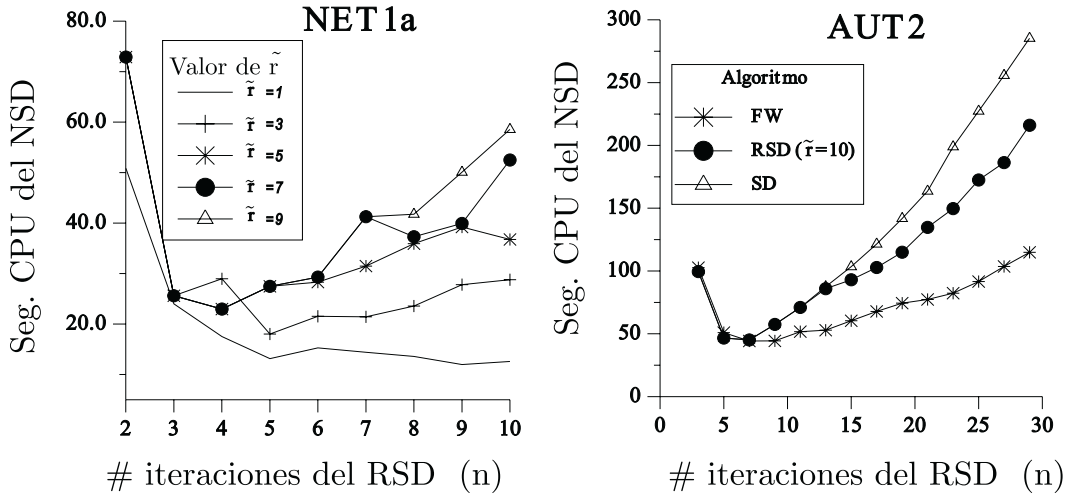


Figura 3.4: Eficiencia del uso del algoritmo RSD en la resolución de los problemas cuadráticos del método NSD (tipo GLP): estudio de los parámetros r̃ y n

sensible a elegir valores grandes del parámetro n.

La figura 3.5 muestra el tiempo de CPU empleado en la resolución de la red NET2b mediante el algoritmo $\mathrm{GLP}(1,n)^{8,1}_{\infty}$ y $\mathrm{N}(1,n)^{8,1}_{\infty}$. En estos algoritmos se ha aplicado el algoritmo de FW para aproximar los problemas cuadráticos. En la figura se observa que la eficiencia de los métodos depende significativamente del parámetro n (número de iteraciones realizadas con el FW para aproximar el subproblema cuadrático del CGP).

Las conclusiones derivadas de este experimento concuerdan con las obtenidas en Larsson y otros [154, 141]. El óptimo número de iteraciones depende de cada algoritmo y de cada problema. Un nuevo aspecto se plantea en estos experimentos, como es, la introducción del algoritmo RSD en lugar del FW para aproximar los subproblemas cuadráticos en el CGP. A priori, el RSD posee mejores propiedades de convergencia que el FW, pero estas ventajas no conducen a una mejora del NSD, como muestra este experimento.

EXPERIMENTO 1.1: estudio del parámetro n_c.

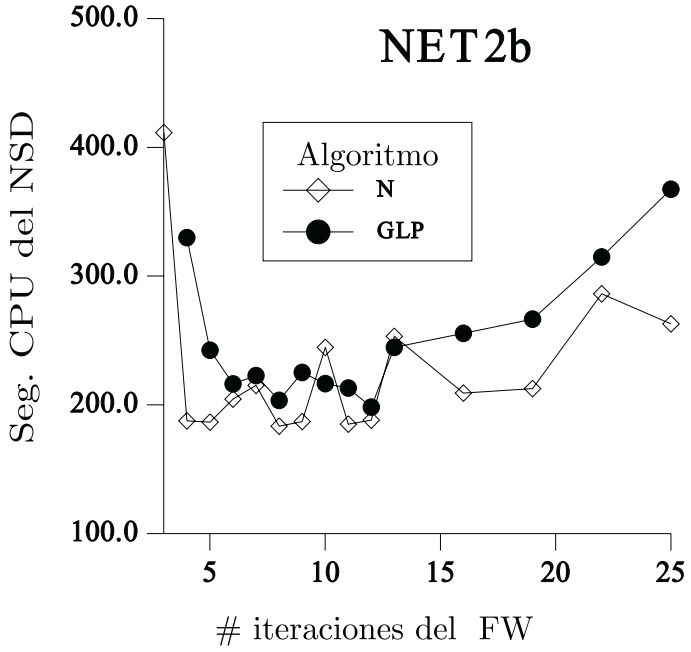


Figura 3.5: Tiempo de CPU empleado por los algoritmos $\mathrm{GLP}(1,n)_{\infty}^{8,1}$ y $\mathrm{N}(1,n)_{\infty}^{8,1}$ en función del parámetro n sobre la red NET2b

Este parámetro controla la calidad de las columnas generadas y es un mecanismo para obtener columnas de gran calidad, diferente de los empleados por la clase NSD para este fin, que son: mejorar en las aproximaciones del problema original y/o aumentar la precisión en su resolución. El CG/SD podría mejorar la calidad de las columnas incrementando el número de veces que el algoritmo $\mathcal{A}_{\mathrm{c}}$ se aplica en el CGP.

El parámetro n_c de la clase CG/SD permite generalizar la clase NSD, en la que este valor siempre es $n_c = 1$.

EXPERIMENTO 1.1.a: estudio del parámetro n_c en el coste computacional de los algoritmos CG/SD.

El experimento 1.1.a tiene el objetivo de evaluar el efecto del número de iteraciones realizadas con el algoritmo empleado en el CGP, n_c, sobre la eficiencia computacional del algoritmo CG/SD. El experimento resuelve las redes NET1a y AUT1 con el algoritmo $\mathrm{RSD}(\tilde{r})_{\infty}^{n_r,n_c}$, para varios valores de los parámetros n_c y $\tilde{r}$, y calcula el tiempo total empleado por el algoritmo CG/SD. Los valores para el parámetro n_r han sido 8 y 15 para las redes NET1a y AUT1 respectivamente.

La figura 3.6 muestra los resultados obtenidos. El comportamiento del algoritmo CG/SD frente al parámetro n_c es similar al comportamiento que el NSD presenta frente al número de iteraciones del algoritmo FW empleadas para resolver truncadamente el problema cuadrático, esto es n . Se observa que pequeños valores del parámetro n_c reducen el coste computacional del algoritmo, pero éste empieza a crecer a partir de ciertos valores del parámetro. Esto podría indicar que debe existir un equilibrio entre la calidad de las columnas y el coste computacional necesario para obtenerlas.

La principal conclusión es que la mejor elección del parámetro n_c tine que ser un valor mayor que uno. El rango de valores óptimos está entre 5 y 10. Esto ilustra la mejora de la clase CG/SD respecto a la descomposición simplicial original, obtenida para $n_c = 1$.

En la red NET1a (dibujo de la izquierda) se puede observar que el mejor algoritmo empleado en el CGP está definido por la elección de $\tilde{r} = 1$, es decir, un FW. En la red AUT1 (dibujo de la derecha) el mejor algoritmo para emplear en el CGP se obtiene para $\tilde{r} = \infty$, es decir, la descomposición simplicial.

La figura 3.7 repite el experimento anterior para la red NET2b. La diferencia está en la inicialización de los RMP del algoritmo RSD aplicado en el CGP. En este experimento, se inicializa el RMP

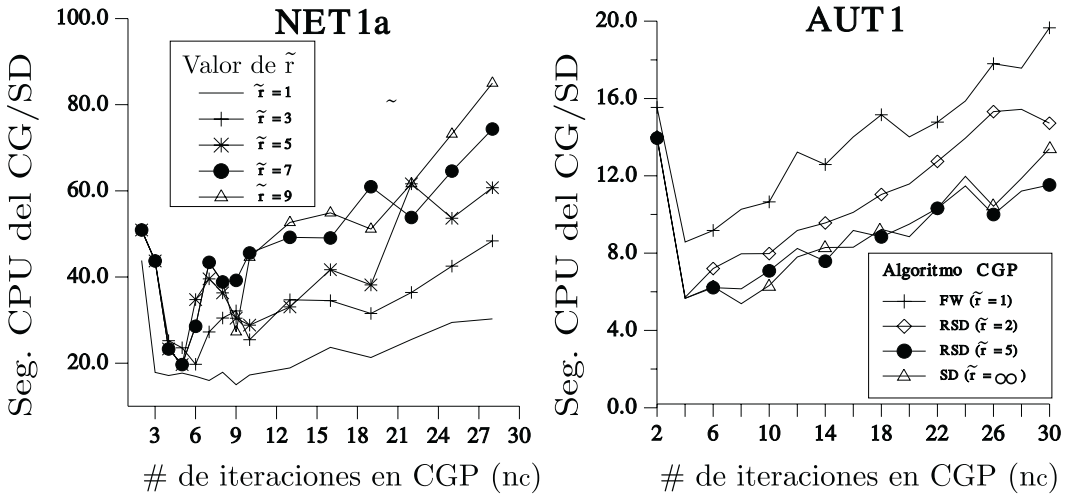


Figura 3.6: Tiempo de CPU empleado por $\text{RSD}(\tilde{r})_{\infty}^{n_r,n_c}$ para resolver NET1a y AUT1 en función de n_c

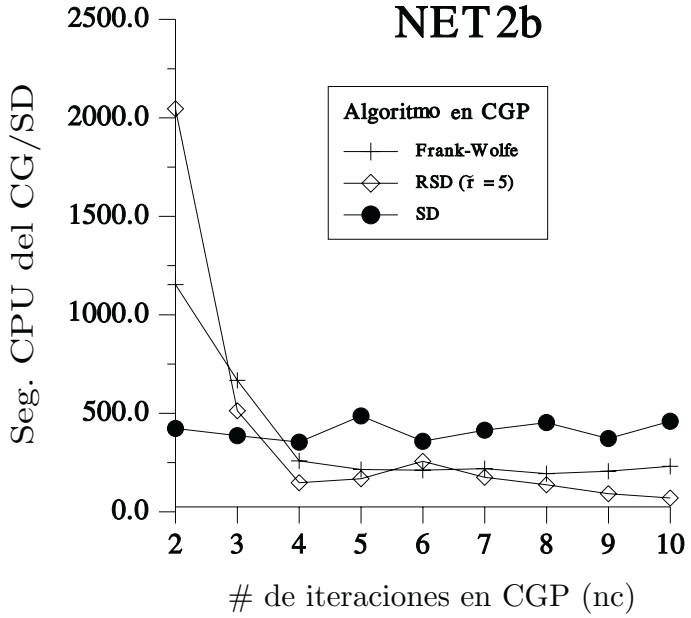


Figura 3.7: Tiempo de CPU empleado por el algoritmo $\text{RSD}(\tilde{r})_{\infty}^{n_r,n_c}$ aplicado a la red NET2b en función de n_c. La inicialización de los RMP en la fase CGP son los puntos extremos retenidos en el último RMP del anterior CGP.

(del CGP) con los puntos extremos retenidos en el último RMP del anterior CGP. En el ejemplo, el RSD(5) es más ventajoso que el algoritmo de FW.

Para la red AUT1, el valor óptimo de n_c es 5, pero para la red NET2b, el valor óptimo de n_c es cercano a 15. En la red NET1a es recomendable emplear el algoritmo de FW mientras que en las AUT1 y NET2b es más ventajoso emplear el RSD. Estos hechos introducen dos preguntas a contestar: ¿Cuáles son los factores que determinan la elección del valor óptimo de n_c? ¿Por qué en un ejemplo es recomendable usar el RSD con $\tilde{r} > 1$ en el CGP y en otro ejemplo es preferible el algoritmo de FW?

EXPERIMENTO 1.1.b: estudio de la influencia del parámetro n_c sobre el número de iteraciones principales, y el número de puntos extremos generados por el algoritmo CG/SD.

El objetivo del experimento 1.1.b es entender los mecanismos mediante los cuales el uso del

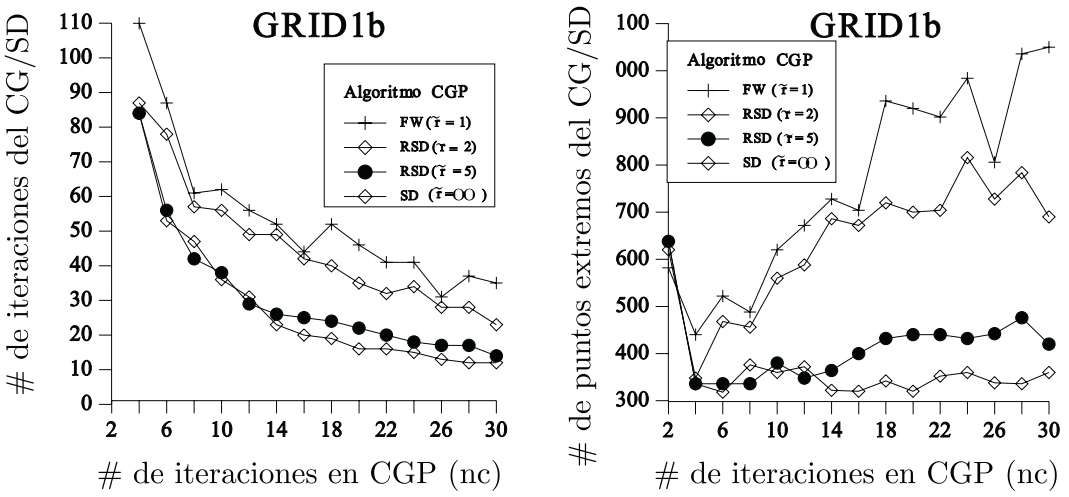


Figura 3.8: Número de iteraciones principales y número de puntos extremos de los algoritmos CG/SD frente al parámetro n_c

parámetro n_c mejora la descomposición simplicial original.

El experimento resuelve los problemas NET1 y GRI1b con el algoritmo RSD$(\tilde{r})_{\infty}^{n_r,n_c}$ para varias combinaciones de los parámetros n_c y $\tilde{r}$. El valor de n_r ha sido considerado fijo para cada red y con un valor de 10.

La figura 3.8 muestra el número de iteraciones principales y el número total de puntos extremos generados por el algoritmo CG/SD, en función del número de iteraciones realizadas en el CGP para la red GRI1b. Notar que el número de puntos extremos generados por el algoritmo CG/SD se calcula como el número de iteraciones principales realizadas con el algoritmo CG/SD multiplicado por el número de n_c puntos extremos generados en cada iteración principal.

En la figura 3.8 (en la izquierda) se puede observar que si el parámetro n_c es incrementado, entonces el número de iteraciones principales del método CG/SD, para alcanzar una determinada precisión, se reduce. Esto implica que un incremento del parámetro n_c produce un número menor de RMP y además, estos problemas tienen un número menor de variables. Por tanto, un incremento del valor del parámetro n_c siempre reduce el tiempo de CPU empleado en la resolución de los problemas RMP.

En la parte derecha de la figura 3.8, se ve que el número total de puntos extremos generados (que luego definirán las columnas) depende de n_c. Esta cantidad disminuye para pequeños valores de n_c y cuando cierto límite es sobrepasado comienza a aumentar. También se muestra que, para ciertos valores de n_c, se reduce el coste computacional de los CGP respecto a la descomposición simplicial original ($n_c = 1$).

El coste de computación de un algoritmo CG/SD es la suma del coste de la resolución de los CGP más el coste de resolución de los RMP. El método CG/SD resuelve un CGP y un RMP en cada iteración principal, por lo que el número total de cada uno de ellos coincide con el número de iteraciones principales. El coste de cada CGP se mantiene constante a lo largo de todas las iteraciones, mientras que la complejidad de los RMP va aumentando conforme el algoritmo progresa, debido a que en cada iteración se suele incluir una nueva variable. El hecho de que el parámetro n_c reduzca el número de iteraciones principales hace que el tiempo empleado en la resolución de los RMP se vea reducido por dos motivos: por un número menor de problemas y por un tamaño menor de éstos.

En la misma figura se observa que aumentando el parámetro n_c se mejora el tiempo empleado en la fase CGP y RMP, pero cuando se sobrepasa cierta cantidad, esta mejoría sólo se produce en la fase RMP.

La conclusión obtenida sobre el parámetro n_c, en este experimento, puede ser formulada en términos del concepto más general de *calidad* de las columnas. Este término aúna el papel del algoritmo

$\mathcal{A}_c$ y del parámetro n_c como mecanismos de generación de columnas de alta calidad. Es decir, podemos incrementar la calidad de una columna eligiendo cuidadosamente el algoritmo $\mathcal{A}_c$ o, por contra, aumentando el número de iteraciones que realizamos con él. En ambos casos este incremento de la calidad de las columnas reduce el número de RMP y de CGP, pero por otro lado, el esfuerzo computacional empleado en los CGP aumenta, esto conduce a un compromiso entre la reducción del número de CGP y su incremento en el coste computacional.

Esta evidencia justifica los resultados del experimento 1.1.a (NET1a) donde el algoritmo con peor tasa de convergencia, el algoritmo de FW, es ventajoso frente al algoritmo RSD (con mejores propiedades de convergencia). Este comportamiento computacional se explica porque, aunque las columnas generadas por el RSD($\tilde{r}$) son de mayor calidad, no lo son lo suficiente para reducir significativamente el número de iteraciones principales, de modo que compense el mayor esfuerzo computacional en la generación de estas columnas. Sin embargo el FW produce unas columnas de menor calidad pero a un coste computacional menor, lo cual es globalmente ventajoso para el algoritmo CG/SD. Para la red AUT1 este compromiso entre calidad y esfuerzo computacional es en el sentido contrario, esto es, las columnas de mayor calidad son recomendables. La conclusión general del experimento 1.a indica que el nivel de calidad de las columnas dependerá del tipo de problema. El experimento siguiente introduce un elemento clave en este análisis: el nivel de precisión exigido en la resolución del problema CDP(f, X).

EXPERIMENTO 1.1.c: el parámetro n_c frente a la precisión exigida en la resolución del CDP(f, X).

Este experimento se ha diseñado para evaluar la influencia del parámetro n_c en la eficiencia del método CG/SD, en función de la precisión demandada en la solución del CDP.

En este experimento se resuelve la red SiF2 con el algoritmo $\mathrm{E}_r^{n_r,n_c}$ para las precisiones relativas de $1.D+00, 1.D-01, 1.D-02, 1.D-03, 1.D-04$[1], empleando varias combinaciones de los parámetros n_c y n_r. Existen interacciones entre los parámetros n_c y las combinaciones de los parámetros n_r y r empleados para definir los RMP. Computacionalmente es muy costoso cubrir todas las combinaciones de estos parámetros, por lo que hemos considerado tres valores pequeños de $n_c = 2, 3, 4$ y tres valores grandes $n_c = 10, 20, 30$. Hemos elegido el valor $r = \infty$, por tratarse de la mejor elección (como se verá más adelante), y hemos considerado dos valores del parámetro n_r. El primero, $n_r = 100$, representa una solución casi exacta del RMP mientras que el otro valor elegido, $n_r = 10$, define una solución truncada del RMP.

Un algoritmo de la clase NSD se obtiene empleando una iteración del algoritmo de Evans, que con la notación introducida para la clase CG/SD, se denota por $\mathrm{E}_r^{n_r,1}$. Hemos comparado el algoritmo $\mathrm{E}_r^{n_r,1}$ con otros algoritmos para los cuales $n_c > 1$. El ratio empleado para establecer esta comparación lo definimos por

$$\text{Ratio de tiempos de CPU } = \frac{\text{Tiempo de CPU empleado por } \mathrm{E}_r^{n_r,1}}{\text{Tiempo de CPU empleado por } \mathrm{E}_r^{n_r,n_c}},$$

en función de n_c y del nivel de precisión de la solución de CDP(f, X). Esta tasa mide el nivel de mejora del CG/SD respecto a un esquema de tipo NSD.

La figura 3.9 muestra los resultados obtenidos en el experimento. Se observa, en primer lugar, que el valor óptimo para n_c depende del nivel de precisión demandado. Para valores pequeños de precisión, también se requieren valores pequeños de n_c. Por otro lado, para niveles altos de precisión se requiere de valores grandes de n_c. La segunda observación es que el esquema original $E_\infty^{n_r,1}$ es mejorado cuando los niveles de precisión demandada aumentan, siendo significativos para grandes precisiones. El rango de esta mejora, para todas las precisiones, varía entre 4.0 y 47.1 para el parámetro $n_r = 10$ y para la mejor elección de n_c. Esta mejora está comprendida entre 10.2 y 44.5 para $n_r = 100$ y la mejor elección de n_c.

Para la mejor elección de n_r, que es $n_r = 10$, y para una precisión usual en las aplicaciones reales, que está entre $1.D-02$ y $1.D-03$, obtendríamos que el factor de mejora estaría comprendido entre

[1] La notación xDy representa $x10^y$.

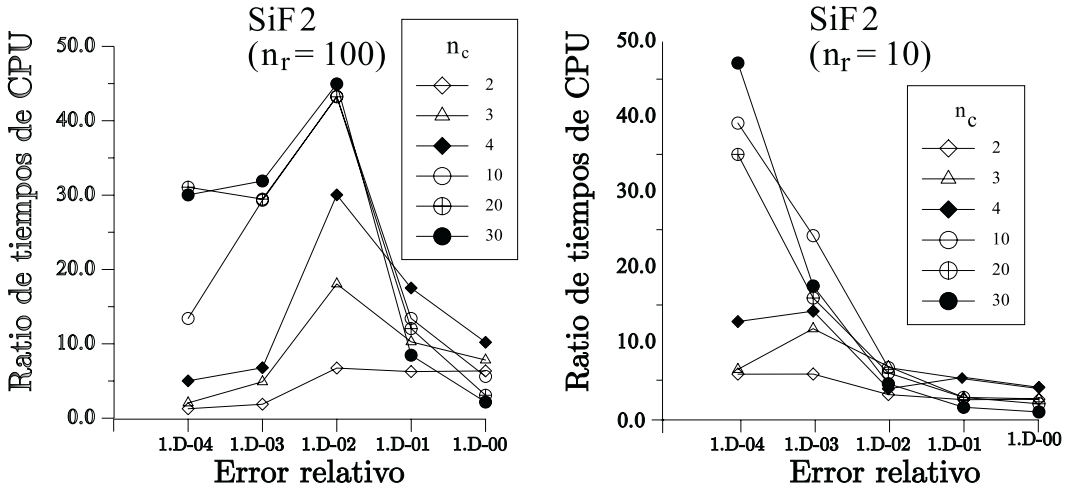


Figura 3.9: Ratio de tiempos de CPU empleados por los algoritmos $\mathrm{E}_{\infty}^{n_r,n_c}$ y $\mathrm{E}_{\infty}^{n_r,1}$ frente a la precisión relativa

6.8 y 24.2.

Existen solamente dos casos donde la elección de $n_r > 1$ no es recomendable, que son cuando $n_c = 30$, para una precisión $1.D+00$ y para los dos valores de n_r.

EXPERIMENTO 1.1.d: influencia del parámetro n_c y el nivel de precisión en el número de iteraciones principales y en el número total de iteraciones interiores realizadas en el CGP

El experimento 1.1.d ha sido diseñado para entender la mejora del algoritmo $\mathrm{E}_{\infty}^{n_r,n_c}$ con respecto al algoritmo $\mathrm{E}_{\infty}^{n_r,1}$ observada en el experimento 1.1.c .

En el experimento se resuelve la red SiF2 con $\mathrm{E}_{\infty}^{100,n_c}$ para los errores relativos de $1.D-02$ y $1.D-04$. Hemos monitorizado los parámetros: número de iteraciones principales, tamaño del último RMP y número de iteraciones totales del algoritmo de Evans. Estos parámetros definen la complejidad de la fase CGP y de los RMP. Estas medidas son independientes del ordenador y del tamaño del problema.

En el experimento 1.1.b se mostró que el algoritmo CG/SD siempre mejora el tiempo empleado en el problema RMP, pero la reducción del coste computacional en el CGP depende del balance entre la reducción del número de CGP y el incremento de su complejidad computacional. Los resultados obtenidos para el experimento 1.1.d son mostrados en la figura 3.10. La principal conclusión derivada, es que este balance depende de la precisión demandada. En este ejemplo, para un error relativo de $1.D-02$, el número de iteraciones del algoritmo de Evans para $\mathrm{E}_{\infty}^{100,n_c}$ es menor que $\mathrm{E}_{\infty}^{100,1}$ cuando $1 < n_c \leq 6$, y viceversa, en el otro caso. $\mathrm{E}_{\infty}^{100,n_c}$ con $n_c > 1$ realiza mayor número de iteraciones del algoritmo de Evans que $\mathrm{E}_{\infty}^{100,1}$ para la precisión relativa de $1.D-04$, esto significa que el coste de computación de la fase CGP crece con n_c para esta precisión, pero el coste de computación de los RMP decrece con n_c.

Cuando la precisión demandada es excesivamente grande, los tamaños de los problemas de RMP hacen impracticable su resolución. La clase CG/SD nos permite introducir una estrategia para mantener pequeños los problemas RMP. Para estos niveles de precisión, esta es la principal fuente de mejora frente al SD clásico.

EXPERIMENTO 1.2: el papel de los parámetros n_r y r

Los parámetros n_r y r definen la complejidad de los RMP. El valor de n_r monitoriza la precisión de la resolución de los RMP y r sus tamaños. Si empleamos valores pequeños de r, estos problemas serán fácilmente resolubles, pero la cantidad que debamos resolver crecerá. Si eligiésemos grandes valores del parámetro r, resolveríamos una menor cantidad de problemas, pero el coste computacional de cada uno de estos problemas sería mayor. Esto conduce a plantearnos cuál es la mejor elección de

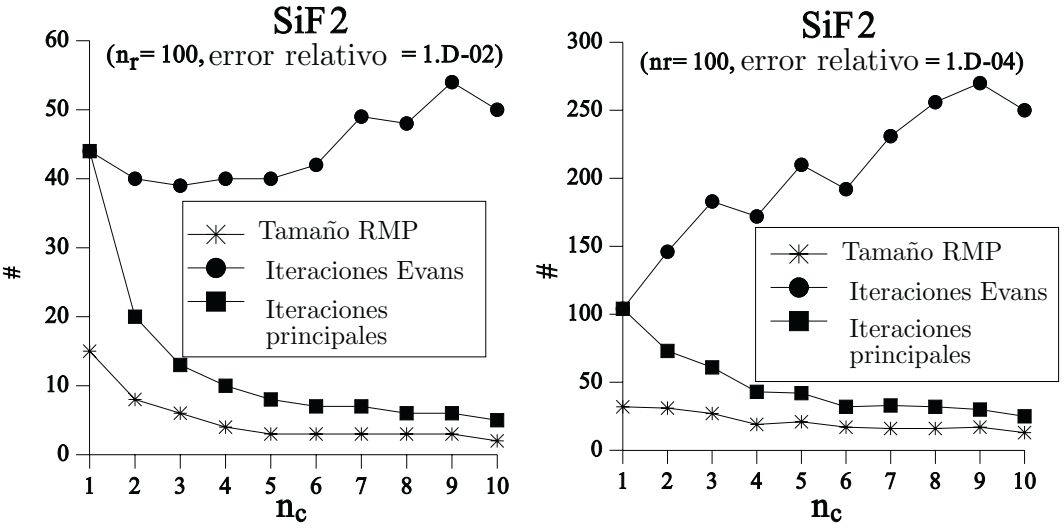


Figura 3.10: Número de iteraciones principales, número de columnas retenidas en el último RMP y número de subproblemas de Evans frente a n_c

los parámetros n_r y r.

EXPERIMENTO 1.2.a: el papel del parámetro r

El experimento 1.1 muestra que los métodos CG/SD con $n_c > 1$ resuelven un menor número de problemas RMP y éstos poseen un menor número de variables, comparados con la elección $n_c = 1$. Una consecuencia de este comportamiento, es que no es necesario introducir una estrategia para mantener pequeño el tamaño del RMP. Por tanto, la mejor elección es tomar $r = \infty$, es decir, emplear un esquema de descomposición simplicial (sin restricción del problema maestro). Para evidenciar esta afirmación, hemos considerado la evolución del tiempo de CPU empleado por los algoritmos $\mathrm{FW}_r^{n_r,1}$ (que es el RSD) y FW_r^{10,n_c} en función del parámetro r. En el experimento hemos resuelto la red Hul2, para una precisión relativa de $1.D-03$, mediante estos algoritmos y para varios valores de los parámetros n_c, n_r y r. Hemos empleado tres valores de n_r por ocho valores del parámetro r en $\mathrm{FW}_r^{n_r,1}$ y tres valores de n_c por ocho valores del parámetro r en el algoritmo FW_r^{10,n_c}.

Los resultados son mostrados en la figura 3.11. La conclusión principal es que la mejor elección se obtiene para el valor del parámetro $r = \infty$ (un esquema de descomposición simplicial) y para valores de $n_c > 1$ (figura del lado derecho). Por otro lado, $\mathrm{FW}_r^{n_r,1}$ es más sensible a los parámetros r y n_r, en este caso, el mejor valor es alcanzado en un valor finito de r (un esquema de descomposición simplicial restringida). Además, la eficiencia del algoritmo $\mathrm{FW}_r^{n_r,1}$ depende significativamente del valor n_r.

EXPERIMENTO 1.2.b: el parámetro n_r y r frente a la calidad de las columnas

Una cuestión clave es determinar la interacción entre el parámetro r y el número de proyecciones n_r (número de iteraciones del método de Newton) en el RMP con el fin de elegir la mejor combinación de ambos parámetros.

Hemos empleado como referencia el algoritmo $\mathrm{RSD}_r^{n_r,n_c}$ y la red AUT1. Este problema ha sido resuelto empleando varios valores de r y considerando dos niveles de truncamiento en el RMP. Hemos definido una precisión alta tomando el parámetro $n_r = 20$ y una precisión baja con el valor $n_r = 5$. Hemos trabajado con dos niveles en la calidad de las columnas generadas: baja y alta. Las columnas del primer tipo se obtienen con $n_c = 3$ iteraciones del algoritmo de Frank-Wolfe. Las del segundo tipo se obtienen mediante dos procedimientos, en el primero efectuamos $n_c = 3$ iteraciones del SD y en el segundo se realizan $n_c = 6$ iteraciones del algoritmo de Frank-Wolfe. La figura 3.12 muestra los resultados obtenidos.

La primera observación es que el comportamiento de este ejemplo, con valores de $n_c > 1$, es

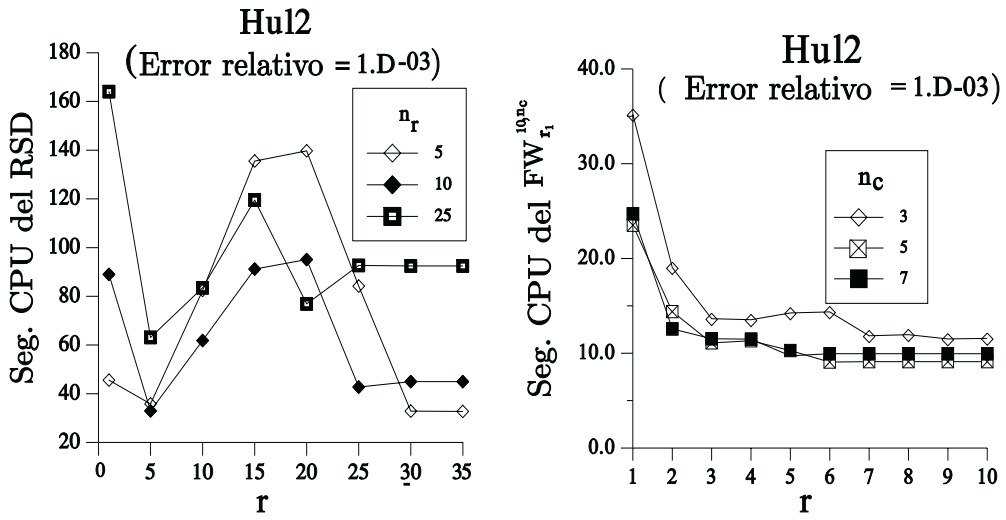


Figura 3.11: Tiempo de CPU empleado por los algoritmos RSD y FW_r^{10,n_c} frente al parámetro r

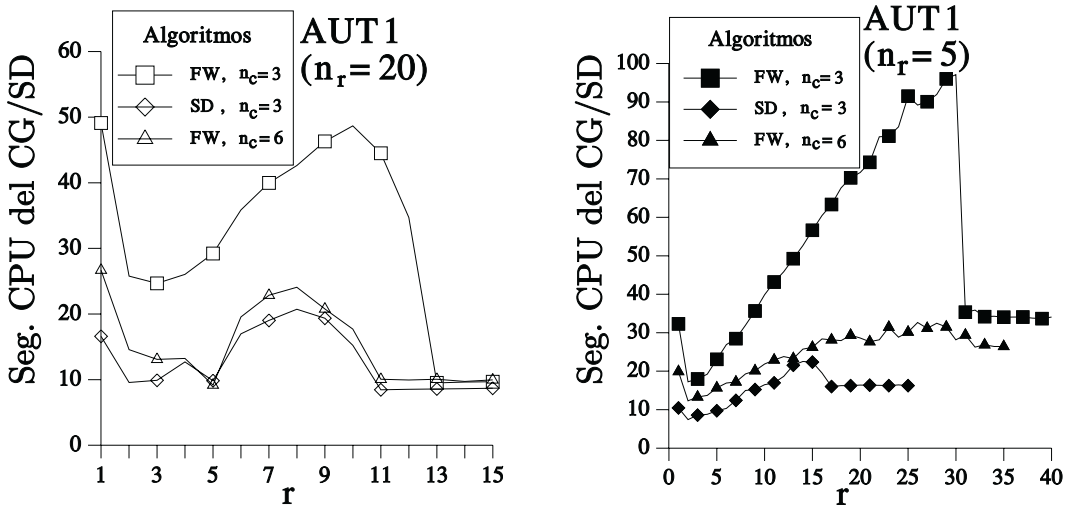


Figura 3.12: Interacciones entre el parámetro r y n_r para el algoritmo $\mathrm{RSD}_r^{n_r,n_c}$ sobre la red AUT1

similar al encontrado en el experimento 1.2.a para el algoritmo RSD (gráfica del lado izquierdo de la figura 3.11). Para la red AUT1 y para $n_r = 5$ (ver figura 3.12 en el lado derecho) la mejor elección del parámetro r es 4. Este resultado cuestionaría las conclusiones obtenidas en el experimento 1.2.a donde la mejor elección era $r = \infty$ para los métodos CG/SD en los que $n_c > 1$.

Lo que ocurre es que si la calidad de las columnas es baja respecto de la precisión demandada para resolver CDP(f, X), entonces el número de problemas RMP será grande, y los parámetros n_r y r tendrán el mismo papel que en el algoritmo RSD. Este hecho añade un nuevo elemento para analizar los parámetros n_r y r: *la interacción entre la calidad de las columnas generadas en relación a la precisión demandada*. La conclusión obtenida en el experimento 1.2.a se circunscribe a columnas de alta calidad.

La primera observación, derivada de la figura 3.12, es que el rango donde varía el tiempo de CPU en las columnas de baja calidad es mayor que el rango para las columnas de calidad alta. Esto muestra mayor sensibilidad de los algoritmos CG/SD a los parámetros n_r y r para columnas de baja calidad. Esto es debido a que las columnas de baja calidad generan mayor número de problemas RMP y por tanto se vuelve clave la estrategia empleada en su resolución.

Los parámetros n_r y r también condicionan el número de iteraciones principales del algoritmo CG/SD. Por ejemplo, cuando se elige $r = 1$ se obtiene un algoritmo de direcciones de búsqueda y el número de RMP es en general grande, pero estos problemas son fácilmente resolubles y en ellos la importancia del parámetro n_r es pequeña. Por otro lado, cuando $r = \infty$ el número de iteraciones del algoritmo CG/SD es mucho menor, pero el coste computacional de estos problemas es grande. Una situación intermedia, tomando valores moderados de r, podría producir un importante número de problemas RMP con un coste computacional no necesariamente moderado. Esto justifica que la mejor elección del parámetro r es tomar valores pequeños (alrededor de 5) o grandes ($r = \infty$). En el primer caso, es recomendable emplear una precisión pequeña en su resolución, mientras que en el otro caso, es recomendable una precisión grande.

Una especial observación es que un algoritmo CG/SD con columnas de baja precisión puede ser mejorado sustancialmente resolviendo los RMP con gran exactitud.

En realidad, la conclusión que se obtuvo en el experimento 1.2.a concuerda con este experimento. Cuando la calidad de las columnas es alta, como en el experimento 1.2.a, las dos recomendaciones para la elección del parámetro (valores grandes o pequeños) es la misma. Como el número de iteraciones principales es pequeño, tomar un valor pequeño del parámetro r es equivalente a emplear un esquema de descomposición simplicial, debido a que el número de columnas almacenadas (como máximo una por iteración) nunca llega a alcanzar el valor del parámetros r y por tanto equivale a tomar $r = \infty$.

Como conclusión, la elección $r = \infty$ es satisfactoria en general y entonces el parámetro n_r debe ser ajustado en función de las calidades de las columnas. Para calidades altas, este ajuste es robusto (es amplio el rango de valores satisfactorios), siendo recomendable valores grandes del parámetro n_r. Para columnas de baja calidad, esta estimación debe realizarse cuidadosamente debido a que la eficiencia del algoritmo CG/SD depende significativamente de este parámetro.

Bloque 2: actualización dinámica de n_c^t

En el experimento 1.1 se vio que el valor óptimo del parámetro n_c para un algoritmo CG/SD depende de cada problema y de la precisión pedida en la solución del problema. En esta sección, planteamos un procedimiento para ir adaptando dinámicamente (en cada iteración t) el valor del parámetro n_c (que representa la calidad de las columnas), en función de los progresos que hace el algoritmo. Suponiendo varias simplificaciones, el método estima el tiempo total de CPU para alcanzar la exactitud requerida en función del parámetro n_c. Entonces, el método elige como n_c^t (valor de n_c en la iteración t) el valor que minimiza el tiempo esperado de CPU.

Asumimos que se emplea el algoritmo iterativo $\mathcal{A}_{\rm c}$ para generar las columnas en CGP. Para describir el método es necesario definir las siguientes variables:

- $\Delta Z_{\rm RMP}(t, n_c) = f(\mathbf{x}^t) - f(\mathbf{x}^{t-1})$, donde la columna introducida en el RMP en la iteración t ha

sido generada mediante la realización de n_c iteraciones del algoritmo $\mathcal{A}_c$. Esta cantidad es el descenso de la función objetivo en el RMP en la iteración t en función del parámetro n_c.

- $\diamond$ $\Delta Z_{\text{CGP}}(t, n_c) = f(\hat{\mathbf{y}}^t, \mathbf{x}^{t-1}) - f(\mathbf{x}^{t-1}, \mathbf{x}^{t-1})$, donde la columna $\hat{\mathbf{y}}^t$ ha sido generada realizando n_c iteraciones del algoritmo $\mathcal{A}_c$. Esta cantidad es la reducción de la función objetivo en el CGP en la iteración t en función del parámetro n_c.
- $\diamond$ $T_{\text{RMP}}(t)$ es el tiempo de CPU empleado en la resolución del RMP en la iteración t.
- $\diamond$ $T_{\text{CGP}}(t)$ es el tiempo de CPU empleado en la fase de CGP en la iteración t.

La exposición del método se realiza en tres etapas. En la primera se estima el tiempo de CPU necesario para que termine el algoritmo CG/SD, asumiendo que el número de iteraciones necesarias es conocido. Este número es denominado n_1. En la segunda etapa, se estima n_1 en función del número de iteraciones realizadas con el algoritmo $\mathcal{A}_c$ en CGP. En la tercera etapa, se describe como se actualizan todos los parámetros involucrados en este método.

Etapa I: estimación del tiempo de CPU

Asumimos que el tamaño del RMP crece en una variable (la columna introducida) en cada iteración. Este fenómeno es frecuente en los ejemplos descritos en este capítulo (sobre un 87.7% en una muestra aleatoria de los problemas de prueba tipo NET). Para simplificar el procedimiento hemos supuesto que el valor del parámetro es $r = \infty$ para el RMP.

También asumimos que toda columna que es introducida en el RMP no es eliminada y el tiempo de CPU empleado para resolver el RMP depende linealmente del número de variables de este problema, esto es, del número de columnas retenidas. El tiempo estimado de CPU para terminar el algoritmo CG/SD tendrá la expresión

$$\begin{aligned} CPU_{\text{RMP}} &= \sum_{i=1}^{n_1} T_{\text{RMP}}(t+i) = \sum_{i=1}^{n_1} \alpha_{\text{RMP}}.(m^t + i) \\ &= m^t \alpha_{\text{RMP}} n_1 + \sum_{i=1}^{n_1} \alpha_{\text{RMP}} i = m^t \alpha_{\text{RMP}} n_1 + \alpha_{\text{RMP}} \frac{n_1(n_1+1)}{2} \end{aligned}$$

donde α_{RMP} es el tiempo de CPU empleado en la resolución del RMP por cada punto retenido, n_1 es el número de iteraciones que deben ser realizadas para resolver CDP(f, X) después de la iteración t, y m^t es el número de puntos retenidos en el RMP en la iteración t-ésima.

Supondremos que $\mathcal{A}_c$ tiene la misma complejidad computacional en todas las iteraciones (interiores) en el CGP. Esta hipótesis implica que el tiempo $T_{\text{CGP}}(t)$ puede ser calculado en función del número de iteraciones n_c^t realizadas por el algoritmo $\mathcal{A}_c$ mediante la expresión $T_{\text{CGP}}(t) = \alpha_{\text{CGP}} n_c^t$, donde α_{CGP} es el tiempo de CPU empleado en realizar una iteración con el algoritmo $\mathcal{A}_c$.

Si empleamos en todas las restantes iteraciones principales el mismo número de veces el algoritmo $\mathcal{A}_c$, llamémosle n_c, el tiempo total de CPU empleado en la fase CGP se puede calcular por la expresión

$$CPU_{\text{CGP}} = \sum_{i=1}^{n_1} T_{\text{CGP}}(t+i) = (\alpha_{\text{CGP}} n_c)\, n_1$$

Etapa II: estimación de n_1

Ahora estimaremos el número n_1 bajo ciertas simplificaciones e hipótesis sobre el comportamiento del algoritmo CG/SD. Suponiendo que en todas las iteraciones el decremento de la función objetivo, producido en el RMP, depende linealmente del descenso de esta función producido en la fase CGP, obtendremos

$$\Delta Z_{\text{RMP}}(t, n_c) = \mu^t \Delta Z_{\text{CGP}}(t, n_c) \tag{3.5}$$

El descenso en la fase CGP depende del número de iteraciones realizadas con el algoritmo $\mathcal{A}_c$. Supondremos que este descenso viene representado por la ecuación

$$\Delta Z_{\text{CGP}}(t, n_c) = \eta^t (n_c)^{\beta^t}, \tag{3.6}$$

donde η^t y β^t son parámetros del modelo.

Este modelo tendrá un buen ajuste para valores pequeños del parámetro n_c, debido a que el modelo puede predecir descensos tan grandes como se deseen, como si el problema fuese no acotado, y esto por lo general no es cierto. Podríamos evitar este comportamiento recurriendo a otros tipos de modelos, como una curva logística, etc., pero nos llevaría a mayores dificultades en la calibración del modelo.

Se ha observado que el modelo lineal tiene un buen ajuste para valores pequeños de n_c. Esta observación empírica conduce a tomar $\beta^t = 1 \quad$ para todo t.

Ahora el problema es estimar el número de iteraciones principales para finalizar el algoritmo CG/SD. Denotamos este número por n_1^t. El criterio de parada empleado para los algoritmos CG/SD es que el error relativo sea menor que una tolerancia dada $\epsilon > 0$. El algoritmo se terminará en la iteración m si se cumple

$$\left| \frac{f(\mathbf{x}^m) - LB^m}{f(\mathbf{x}^m)} \right| \leq \epsilon, \tag{3.7}$$

donde LB^m es una cota inferior del problema en la iteración m y $\mathbf{x}^m$ la iteración. Suponiendo que $f(\mathbf{x}^*) > 0$ y despejando $f(\mathbf{x}^m)$ de (3.7), obtenemos $(1-\epsilon) f(\mathbf{x}^m) \leq LB^m$. Sustituyendo la relación $f(\mathbf{x}^m) = f(\mathbf{x}^t) + \sum_{i=t+1}^{m} \Delta Z_{\text{RMP}}(i, n_c^i)$ en la relación anterior, obtenemos que m debe satisfacer

$$(1-\epsilon) \left[f(\mathbf{x}^t) + \sum_{i=t+1}^{m} \Delta Z_{\text{RMP}}(i, n_c^i) \right] \leq LB^m$$

Una cota superior del número de iteraciones, se obtiene reemplazando la cota inferior en la iteración m, LB^m, por la actual (un valor menor), LB^t, y llegamos a que n_1^t cumple

$$n_1^t \leq p^t = \text{arg minimizar} \left\{ n \in \mathbb{N} : \sum_{i=1}^{n} \Delta Z_{\text{RMP}}(t+i, n_c^{t+i}) \leq \frac{LB^t}{1-\epsilon} - f(\mathbf{x}^t) \right\}$$

Emplearemos como estimación de n_1^t el valor de p^t. Para que este valor esté completamente determinado debemos asumir un comportamiento de $\Delta Z_{\text{RMP}}(i, n_c^i)$ para las próximas iteraciones. Supondremos que este descenso es similar a la iteración actual, es decir:

$$\Delta Z_{\text{RMP}}(t, n_c) \approx \Delta Z_{\text{RMP}}(t+1, n_c) \approx \ldots \approx \Delta Z_{\text{RMP}}(t+n_1, n_c)$$

Bajo esta hipótesis, se puede calcular el valor de p^t y obtener la estimación de n_1^t

$$n_1^t(n_c) := \left[\frac{\frac{LB^t}{1-\epsilon} - f(\mathbf{x}^t)}{\Delta Z_{\text{RMP}}(t, n_c)} \right] + 1 = \left[\frac{\frac{LB^t}{1-\epsilon} - f(\mathbf{x}^t)}{\mu^t \eta^t n_c} \right] + 1, \tag{3.8}$$

donde [.] es la parte entera de un número.

Elegiremos como n_c^{t+1}, para el próximo CGP, el número que minimice el tiempo esperado, es decir,

$$\begin{aligned} n_c^{t+1} \quad = \quad & \text{arg minimizar}_{n_c \in \mathbb{N}} CPU_{\text{CGP}}(n_c) + CPU_{\text{RMP}}(n_c) = \\ & \text{arg minimizar}_{n_c \in \mathbb{N}} \alpha_{\text{CGP}} n_1^t(n_c) n_c + m^t \alpha_{\text{RMP}} n_1^t(n_c) + \alpha_{\text{RMP}} \frac{n_1^t(n_c)\,(n_1^t(n_c)+1)}{2} \end{aligned}$$

donde $n_1^t(n_c)$ es dado en (3.8). Para resolver este problema en la práctica, hemos restringido la variable n_c al conjunto $[4, 20] \cap \mathbb{N}$ y, mediante la evaluación de la función objetivo sobre ese conjunto discreto, calculamos el valor mínimo.

Etapa III: actualización de parámetros

En cada iteración actualizamos los parámetros empleados para calcular el valor de n_c^{t+1}.

La estimación del tiempo de CPU usado en el RMP por cada variable es

$$\alpha_{\text{RMP}} = \frac{T_{\text{RMP}}(t)}{m^t}$$

donde m^t es el número de puntos retenidos en RMP en la iteración t. No empleamos la información anterior de los RMP debido a que sus tamaños pueden ser demasiado diferentes y podría producir una estimación sesgada de este valor.

El parámero α_{CGP} se calcula como la media de los tiempos empleados en las iteraciones del algoritmo $\mathcal{A}_c$

$$\alpha_{\text{CGP}} = \frac{\sum_{i=1}^{t} T_{\text{CGP}}(i)}{\sum_{i=1}^{t} n_c^i}$$

Las predicciones realizadas por los modelos (3.5) y (3.6), que denotamos por $\widehat{\Delta Z_{\text{RMP}}}(t, n_c^t)$ y $\widehat{\Delta Z_{\text{CGP}}}(t, n_c^t)$ son diferentes de lo observado posteriormente. Corregimos los parámetros η^t y μ^t para cumplir las igualdades $\Delta Z_{\text{RMP}}(t, n_c^t) = \widehat{\Delta Z_{\text{RMP}}}(t, n_c^t)$ y $\Delta Z_{\text{CGP}}(t, n_c^t) = \widehat{\Delta Z_{\text{CGP}}}(t, n_c^t)$. Obteniendo que

$$\eta^{t+1} = \frac{\Delta Z_{\text{CGP}}(t, n_c^t)}{n_c^t}$$
$$\mu^{t+1} = \frac{\Delta Z_{\text{RMP}}(t, n_c^t)}{\Delta Z_{\text{CGP}}(t, n_c^t)}$$

Una mejor estimación del parámetro μ^{t+1} se obtiene suponiendo que este parámetro depende linealmente del GAP (diferencia entre la cota inferior y cota superior) en la iteración $t+1$, esto es, $\mu^{t+1} = \mu GAP^{t+1}$, entonces

$$\frac{\mu^{t+1}}{\mu^t} = \frac{\mu GAP^{t+1}}{\mu GAP^t}, \text{ por tanto } \mu^{t+1} = \mu^t \frac{GAP^{t+1}}{GAP^t},$$

donde $GAP^t = f(\mathbf{x}^t) - LB^t$ para toda las iteraciones t y μ^t es calculado por $\frac{\Delta Z_{\text{RMP}}(t, n_c^t)}{\Delta Z_{\text{CGP}}(t, n_c^t)}$. Se puede observar que estas expresiones requieren conocer cotas inferiores al valor de $\text{CDP}(f, X)$ y eso no es posible para ciertos algoritmos $\mathcal{A}_c$.

EXPERIMENTO 2.1: actualización dinámica del parámetro n_c^t.

El experimento 1.2 nos muestra que es aconsejable tomar un gran valor de n_r y tomar $r = \infty$ cuando las columnas son de alta calidad. Pero no hay ninguna recomendación para el parámetro n_c. Esta elección depende de la precisión demandada y del tipo de problema. Este inconveniente es el que quiere solucionar el procedimiento de actualización dinámica de n_c^t. Este experimento está diseñado para validar el mismo.

Una motivación similar aparece cuando queremos determinar el número de iteraciones a realizar con un algoritmo para aproximar el problema cuadrático del CGP para un algoritmo NSD. Hemos extendido este procedimiento a esta clase, en concreto, lo hemos aplicado a los algoritmos $\text{N}_\infty^{n_r,1}$ y $\text{GLP}_\infty^{n_r,1}$. Se ha hecho considerando que $\Delta Z_{\text{CGP}}(t, n_c)$ es proporcional al decremento de la aproximación cuadrática, esto es, $\Pi(\hat{\mathbf{y}}^t, \mathbf{x}^{t-1}) - \Pi(\mathbf{x}^{t-1}, \mathbf{x}^{t-1})$, donde Π es la aproximación cuadrática en el punto $\mathbf{x}^{t-1}$.

La figura 3.13 muestra la evolución del tiempo total de CPU para los algoritmos $\text{FW}_\infty^{n_r,n_c}$, $\text{N}_\infty^{n_r,1}$ y $\text{GLP}_\infty^{n_r,1}$, en función del número de iteraciones del algoritmo de FW, empleado en el CGP para la red NET1b.

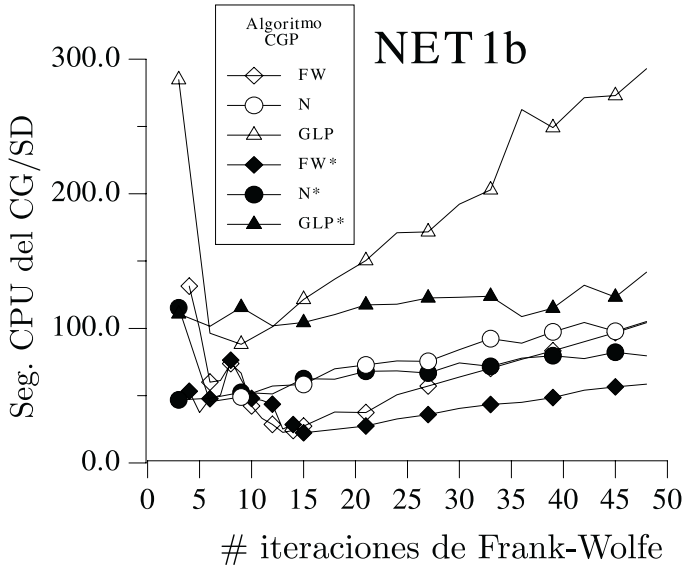


Figura 3.13: Actualización dinámica del parámetro n_c. El símbolo * denota que el algoritmo emplea la actualización dinámica

La actualización dinámica requiere un valor de n_c en la primera iteración. Las curvas en negrilla muestran la evolución de estos algoritmos en función de este valor inicial. Las otras curvas se obtienen haciendo siempre un número fijo de iteraciones del algoritmo de FW (ya sea aplicada al problema original o a la aproximación cuadrática). En resumen, en el método de actualización dinámica se elige sólo el parámetro en la primera iteración, frente al otro procedimiento que considera el mismo valor en todas las iteraciones.

En figura 3.13 se ilustra que el tiempo de CPU se reduce significativamente si el número de iteraciones n_c no es ni demasiado pequeño, ni demasiado grande, para el procedimiento estático. La actualización dinámica tiene un comportamiento similar a la elección adecuada pero para casi todos los valores iniciales de n_c. Esto prueba que este método minimiza las consecuencias de un error en la elección del parámetro del parámetro n_c.

Bloque 3: prolongación a la frontera relativa

Esta sección aborda el papel de la prolongación de la columna a la frontera, definida en (2.6), para los métodos CG/SD. Hemos realizado cuatro experimentos numéricos y una discusión teórica sobre el modo de calcular esta prolongación de la columna, en el caso de emplearse métodos de direcciones factibles para el papel de $\mathcal{A}_c$. Además, hemos analizado teóricamente el efecto de no efectuar la prolongación o de hacerlo fuera de la región factible.

Una diferencia fundamental entre el SNFP y el TAP-M es el cálculo de la prolongación a la frontera. Para el SNFP esta prolongación se puede calcular en función del flujo en los arcos (imponiendo la no negatividad de los arcos), pero esto no es posible para problemas de flujos multiproducto.

El experimento 3.1 está diseñado para mostrar el papel de la prolongación en los métodos CG/SD. Además, se obtiene una fórmula para calcular dicha prolongación, que puede ser aplicada al TAP-M. El experimento 3.2 está planteado para medir computacionalmente la velocidad de convergencia de los métodos CG/SD en función de la prolongación. El experimento 3.3 está diseñado para evaluar el efecto de prolongar la columna fuera de la región factible. El último es una aproximación a la relación existente entre la geometría de la cara óptima y prolongación de las columnas.

EXPERIMENTO 3.1: efecto de la prolongación a la frontera relativa en la eficiencia del algoritmo CG/SD.

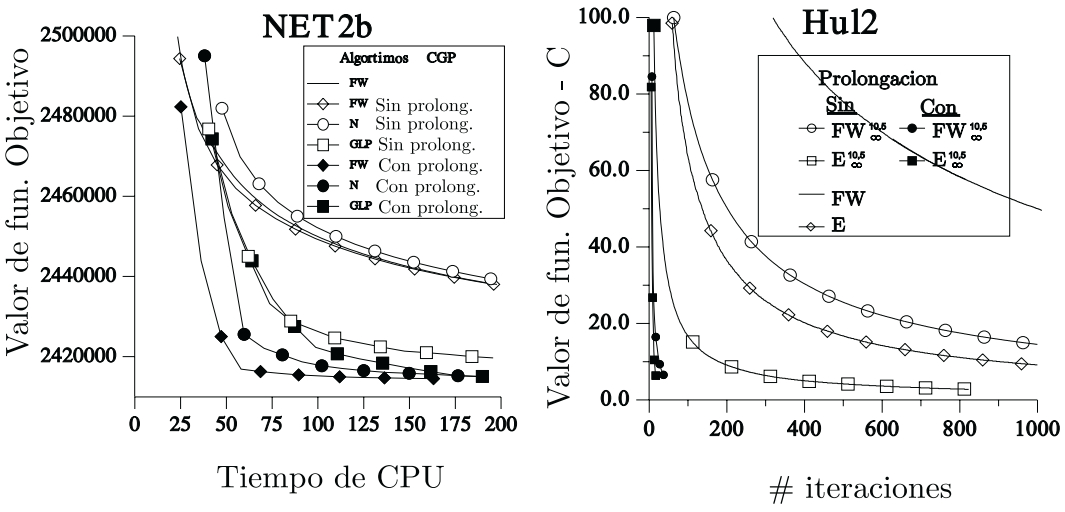


Figura 3.14: Influencia de la prolongación de las columnas a la frontera relativa

Este experimento se resuelve la red NET2b con los algoritmos $\mathrm{N}_\infty^{10,1}$, $\mathrm{GLP}_\infty^{10,1}$ y $\mathrm{FW}_\infty^{10,10}$ con y sin prolongación. También hemos empleado el algoritmo FW como referencia de convergencia sublineal. Los problemas cuadráticos de los algoritmos NSD han sido resueltos empleando 10 iteraciones del algoritmo de FW. Hemos repetido este experimento para el problema Hul2 y para los métodos $\mathrm{FW}_\infty^{10,5}$ y $\mathrm{E}_\infty^{10,5}$.

Los resultados obtenidos en este experimento se muestran en la figura 3.14. La eficiencia de los métodos $\mathrm{N}_\infty^{10,1}$ y $\mathrm{FW}_\infty^{10,10}$ es drásticamente reducida cuando no se efectúa la prolongación de las columnas. Además, la velocidad de convergencia de los algoritmos $\mathrm{FW}_\infty^{10,10}$ y $\mathrm{N}_\infty^{10,1}$ es idéntica al algoritmo de FW. El método $\mathrm{GLP}_\infty^{10,1}$ es más robusto a la supresión de la prolongación, siendo despúes de varias iteraciones cuando la convergencia se ralentiza.

Para el problema Hul2 se observa el mismo comportamiento. En este caso no está tan claro que, si la prolongación de la frontera no se efectúa, los algoritmos $\mathrm{FW}_\infty^{10,5}$ y $\mathrm{E}_\infty^{10,5}$ tengan similares comportamientos a los de FW y E respectivamente. La clave está en el número de iteraciones, en realidad, cuando se realiza una iteración del algoritmo $\mathrm{FW}_\infty^{10,5}$ y $\mathrm{E}_\infty^{10,5}$ se han realizado $n_c = 5$ iteraciones de FW y E. La comparación se debe realizar con 5 iteraciones de los algoritmos FW y E, llegando a ser evidente.

En el experimento numérico se observa que si la prolongación no se efectúa entonces existe un número entero positivo τ cumpliendo

$$\mathbf{x}^t = \hat{\mathbf{y}}^{t-1}, \quad \forall t \geq \tau.$$

Que significa, entre otras cosas, que cuando la prolongación no se efectúa la velocidad de convergencia del algoritmo CG/SD está monitorizada por el algoritmo empleado en CGP.

A continuación damos una justificación de esta afirmación para el caso de problemas convexos continuamente diferenciables, como los son el TAP-M y el SNCF. Asumiremos que la cara óptima es el conjunto $\{\mathbf{x}^*\}$, entonces $\nabla f(\mathbf{x}^*)(\mathbf{y} - \mathbf{x}^*) > 0$ para todo $\mathbf{y} \in X$.

En primer lugar probaremos que $X^t \to \{\mathbf{x}^*\}$. Sea $\tilde{\mathbf{y}}$ una columna generada en alguna iteración. Probaremos que esta columna se elimina en alguna iteración del RMP. Empleando el hecho de que la sucesión $\{\mathbf{x}^t\}$ converge a $\mathbf{x}^*$ y empleando la continuidad de $\nabla f(\mathbf{x})$ en X, obtenemos que existe un entero t cumpliendo que $\nabla f(\mathbf{x}^t)(\tilde{\mathbf{y}} - \mathbf{x}^t) > 0$. Esto implica que el coeficiente de $\tilde{\mathbf{y}}$, en la combinación convexa de puntos $\hat{y} \in \mathcal{P}^t$ para expresar $\mathbf{x}^t$, es cero y el punto es por tanto eliminado. Además, las nuevas columnas están mas cerca a $\mathbf{x}^*$ debido a que $\hat{\mathbf{y}}^t \to \mathbf{x}^*$, entonces $X^t \to \{\mathbf{x}^*\}$.

Ahora probaremos que las soluciones truncadas de los RMP se alcanzan en puntos extremos de X^t para valores suficientemente grandes de t.

El teorema de Taylor justifica la siguiente expresión

$$f(\mathbf{y}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T(\mathbf{y} - \mathbf{x}^*) + \frac{1}{2}(\mathbf{y} - \mathbf{x}^*)^T\nabla^2 f(\mathbf{c})(\mathbf{y} - \mathbf{x}^*) \tag{3.9}$$

donde $\mathbf{c} \in (\mathbf{y}, \mathbf{x}^*)$. Empleando la propiedad de que la sucesión de conjuntos $X^t \to \{\mathbf{x}^*\}$, entonces $\{\text{maximo}_{\mathbf{y}\in X^t}\|\mathbf{y} - \mathbf{x}^*\|\} \to 0$. Asumiendo que las matrices Hessianas están acotadas en X, y por cumplirse $\|\mathbf{y}-\mathbf{x}^*\| \approx 0$ para todo $\mathbf{y} \in X^t$, obtenemos que el término cuadrático en (3.9) es despreciable frente al término lineal para todo $\mathbf{y} \in X^t$. Entonces

$$\arg \text{ minimizar}_{\mathbf{x}\in X^t} f(\mathbf{x}) \approx \arg \text{ minimizar}_{\mathbf{y}\in X^t} \nabla f(\mathbf{x}^*)^T\mathbf{y},$$

y la solución del RMP se alcanza aproximadamente en un punto extremo de X^t. En la iteración siguiente, todos los puntos retenidos en el RMP son eliminados, a excepción de la solución del RMP. Además, se cumple $f(\hat{\mathbf{y}}^{t-1}) < f(\mathbf{x}^{t-1})$ y esto justifica que la columna introducida $\hat{\mathbf{y}}^{t-1}$ es un punto extremo con menor valor que el resto de puntos extremos retenidos y por tanto la solución del RMP en t es $\hat{\mathbf{y}}^{t-1}$, obteniendo $\mathbf{x}^t = \hat{\mathbf{y}}^{t-1}$.

Un resultado derivado del anterior argumento es que X^{t+1} estará definido por la iteración actual $\mathbf{x}^t$ y $\hat{\mathbf{y}}^t$. Lo que conduce a que el conjunto $\mathcal{P}^{t+1}$ tendrá dos puntos. Estos resultados concuerdan con las observaciones numéricas.

Los resultados computacionales muestran que el $\text{GLP}_\infty^{10,1}$ es menos sensible a la supresión de la prolongación. Esto es debido a que la convergencia de $X^t \to \mathbf{x}^*$ es más lenta que los otros métodos. Para justificar esta afirmación, basta considerar que cuando el parámetro γ vale cero en el $\text{GLP}_\infty^{10,1}$, los problemas se transforman al algoritmo de FW y las columnas son puntos extremos (y por tanto no se pueden prolongar más). Asumiendo ciertas propiedades de continuidad de las soluciones de los subproblemas cuadráticos del GLP frente al parámetro γ, podemos considerar que estas soluciones (por lo menos al inicio) están cerca de la frontera para valores del parámetro γ cercanos al cero. Por todo ello, el método $\text{GLP}_\infty^{n_r,n_c}$ podría ser un método eficiente para problemas en los que que fuese imposible calcular la prolongación de las columnas a la frontera relativa. La discusión anterior nos motiva a efectuar la prolongación a la frontera, pero no siempre es posible calcularla para un conjunto convexo general X.

Ahora abordaremos el cálculo para conjuntos convexos generales X y para métodos de direcciones de descenso $\mathcal{A}_c$ empleados en CGP. Supondremos que la búsqueda lineal en un punto $\mathbf{x}$ se formula

$$\min_{\lambda\in[0,1]} f(\lambda\mathbf{x} + (1 - \lambda)\mathbf{p})$$

donde el punto $\mathbf{p}$ es un punto extremo de X. Todos los algoritmos empleados en este trabajo son de este tipo. La siguiente proposición proporciona una fórmula para calcular la prolongación a la frontera relativa, en función de las búsquedas lineales obtenidas en el CGP.

PROPOSICIÓN 3.3.1 *Sea X un conjunto convexo y el algoritmo $\mathcal{A}_c$ empleado en los CGP un método de direcciones factibles. Supongamos que λ_i para $i = 1, \ldots, n_c^t$ son los n_c^t valores de las búsquedas lineales efectuadas en la iteración t en CGP, entonces la prolongación a la frontera se calcula por*

$$\ell_t = \frac{1}{1 - \prod_{i=1}^{n_c^t} \lambda_i}.$$

DEMOSTRACIÓN. Denotamos por $\hat{\mathbf{y}}_i^t$ la $i-$ésima iteración del algoritmo $\mathcal{A}_c$ realizada en GCP en la iteración principal t. La primera iteración se puede expresar por $\hat{\mathbf{y}}_1^t = \lambda_1\mathbf{x}^t + (1 - \lambda_1)\mathbf{p}_1$ donde λ_1 es el valor de la búsqueda lineal de la función f en el intervalo $[\mathbf{x}^t, \mathbf{p}_1]$. La segunda iteración se expresa por $\hat{\mathbf{y}}_2^t = \lambda_2\hat{\mathbf{y}}_1^t + (1 - \lambda_1)\mathbf{p}_2$ donde λ_2 es la solución de la segunda búsqueda lineal. Sustituyendo la anterior expresión de $\hat{\mathbf{y}}_1^t$ en la de $\hat{\mathbf{y}}_2^t$, obtenemos $\hat{\mathbf{y}}_2^t = \lambda_2\lambda_1\mathbf{x}^t + \lambda_2(1-\lambda_1)\mathbf{p}_1 + (1-\lambda_2)\mathbf{p}_2$. Si hiciésemos n_c^t iteraciones del algoritmo $\mathcal{A}_c$ en CGP obtendríamos que la columna generada en la iteración t se puede expresar como combinación convexa del punto actual $\mathbf{x}^t$ y del conjunto de puntos $\{\mathbf{p}_i \mid i \in \hat{\mathcal{P}}\}$.

$$\hat{\mathbf{y}}^t = \mu^t\mathbf{x}^t + \sum_{i\in\hat{\mathcal{P}}} \alpha_i^t\mathbf{p}_i$$

donde $\mu^t = \prod_{i=1}^{n_c^t} \lambda_i$, y

$$\mu^t + \sum_{i \in \hat{\mathcal{P}}} \alpha_i^t = 1,$$
$$\mu^t,\ \alpha_i^t \geq 0, \quad \forall i \in \mathcal{P}.$$

Calculamos la prolongación a la frontera empleando la expresión (3.10), obteniendo la relación

$$\mathbf{y}^t = \mathbf{x}^t + \ell_t(\hat{\mathbf{y}}^t - \mathbf{x}^t) = \mathbf{x}^t + (\mu^t - 1)\ell_t \mathbf{x}^t + \sum_{i \in \hat{\mathcal{P}}} \ell_t \alpha_i^t \mathbf{p}_i = \delta^t \mathbf{x}^t + \sum_{i \in \hat{\mathcal{P}}} \ell_t \alpha_i^t \mathbf{p}_i$$

donde $\delta^t = [1 + (\mu^t - 1)\ell_t]$ y $\delta^t + \sum_{i \in \hat{\mathcal{P}}} \ell_t \alpha_i^t = 1 + (\mu^t - 1)\ell_t + \ell_t(1 - \mu^t) = 1$, y todos los coeficientes son positivos. Esto muestra que la columna prolongada $\mathbf{y}^t$ es una combinación convexa del actual punto $\mathbf{x}^t$ y del conjunto de puntos $\{\mathbf{p}_i\}$. Como $\mathbf{y}^t$ está en la frontera relativa de $\texttt{aff}([\mathbf{x}^t, \hat{\mathbf{y}}^t]) \cap X$ entonces $\delta^t = 0$ y $\ell_t = 1/(1 - \mu^t)$. Sustituyendo el valor de μ^t en función de las búsquedas unidimensionales λ_i, obtenemos que el valor de ℓ_t se calcula con la expresión

$$\ell_t = \frac{1}{1 - \prod_{i=1}^{n_c^t} \lambda_i}$$

□

EXPERIMENTO 3.2: velocidad de convergencia de los métodos CG/SD

El siguiente experimento está diseñado para medir la velocidad de convergencia de los métodos CG/SD con y sin prolongación a la frontera. Primero hemos calculado la solución del problema Hul2 con un error relativo inferior a $5.8{*}10^{-4}\%$. Esta solución es denotada por $\hat{\mathbf{x}}$. Después hemos considerado la sucesión

$$\text{fAbsErr}_t = f(\mathbf{x}^t) - f(\hat{\mathbf{x}}),$$

que es una estimación del error absoluto.

Hemos resuelto el problema Hul2 con los métodos $\text{FW}_\infty^{10,5}$ y $\text{E}_\infty^{10,5}$ con y sin prolongación de las columnas a la frontera relativa. La figura 3.14 (en el lado derecho) muestra la sucesión fAbsErr_t (donde el valor de $C = f(\hat{\mathbf{x}})$) generada por los diferentes métodos frente a la iteración t. Las curvas concuerdan con la expresión funcional

$$\text{fAbsErr}_t = \frac{\alpha}{t^\beta},$$

donde $\alpha, \beta > 0$.

Hemos hecho un ajuste de los errores fAbsErr_t al modelo $\text{fAbsErr}_t = \frac{\alpha}{t^\beta}$, y los resultados se pueden ver en la tabla 3.7. Hemos estimado los parámetros α y β por el método de mínimos cuadrados. La bondad del ajuste ha sido evaluada mediante el coeficiente de determinación R^2, que representa la proporción de la varianza de la variable fAbsErr explicada por la regresión. Estos valores son muy cercanos a 1.

La principal conclusión es que, si la prolongación no es efectuada, la velocidad de convergencia es similar al algoritmo empleado en la fase CGP. Esto es, $\text{FW}_\infty^{10,5}$ es similar a hacer cinco iteraciones del algoritmo FW; $\text{E}_\infty^{10,5}$ a realizar cinco con el algoritmo de Evans. Se observa que si la prolongación se realiza, el valor de β es mayor que 1, en caso contrario β es menor que 1.

EXPERIMENTO 3.3: prolongación de las columnas fuera de la región factible

La convergencia de los algoritmos CG/SD está asegurada bajo la hipótesis de que las columnas introducidas en el RMP pertenecen a la región factible X. En este experimento contrastamos la robustez de la convergencia de los algoritmos frente a columnas no factibles. En el experimento prolongamos las columnas fuera de la región factible. La motivación de este experimento estriba en que en algunos problemas puede ser imposible calcular esta prolongación.

Tabla 3.7: Ajuste de la curva $\text{fAbsErr}_t = \frac{\alpha}{t^\beta}$ para los métodos con y sin prolongación

Algoritmo	con prolongación			sin prolongación		
	α	β	R^2	α	β	R^2
$\text{FW}_\infty^{10,5}$	2799.83	1.7468	0.9676	2013.24	0.7112	0.985
$\text{E}_\infty^{10,5}$	2726.18	2.1362	0.9978	886.305	0.8622	0.9971
FW	2271.94	0.5469	0.9914			
E	3746.75	0.8718	0.9943			

Hemos resuelto el problema NgD2 con una gran exactitud. El error relativo fue menor de $1.26\ 10^{-5}$ %. El valor de la función objetivo fue 41481.90939 y la cota inferior de 41481.90886. Denotamos esta aproximación por $\hat{\mathbf{x}}$. El problema es estrictamente convexo por lo que la solución es única y la denotamos $\mathbf{x}^*$. Esto justifica que la sucesión de iteraciones debe converger a la única solución del problema.

Hemos resuelto este problema con $\text{E}_\infty^{100,5}$ y $\text{FW}_\infty^{500,5}$. En las primeras iteraciones hemos extendido las columnas a la frontera relativa, pero después de una determinada iteración, la prolongación ha sido $10\ell_t$ o $100\ell_t$, es decir diez o cien veces mayor que la verdadera prolongación y por tanto la columna generada no pertenece a la región factible (por la convexidad de X). Hemos monitorizado la convergencia del método CG/SD empleando

$$\text{xRelErr}^t = \frac{\|\mathbf{x}^t - \hat{\mathbf{x}}\|_2}{\|\hat{\mathbf{x}}\|_2}$$

Tabla 3.8: Prolongación fuera de la región factible

Primera prolongación	$\text{E}_\infty^{100,5}$		$\text{FW}_\infty^{500,5}$	
fuera de la región	$10\ \ell_t$	$100\ell_t$	$10\ \ell_t$	$100\ell_t$
5	0.48%[a]	0.42%	error	0.86%
	41476.78 [b]	41478.76		41460.41
10	0.30 %	0.47%	0.14%	0.13%
	41479.55	41479.11	41478.46	41478.56
15	0.10 %	0.14%	0.04%	0.04 %
	41481.48	41482.37	41481.29	41479.42
20	0.18%	0.12%	0.09%	0.05%
	41482.05	41481.83	41480.82	41472.05

[a] xRelErr^t*100.
[b] Valor de la función objetivo.

La conclusión es que el método CG/SD converge a puntos cercanos de la solución óptima, pero estos puntos no son factibles debido a que el valor de la función objetivo en ellos es inferior a la cota inferior del problema. Si el número de iteraciones a partir de la cual se empieza a prolongar fuera de la región factible se incrementa, el punto obtenido en esta sucesión está más cercano a la solución óptima. Otra observación es que si la columna se extiende fuera de la región factible, entonces existen puntos donde la función no está definida (logaritmo de valores negativos). Este es el motivo que produce el error computacional para $\text{FW}_\infty^{500,5}$ y para la iteración quinta, a partir de la cual se extiende fuera de la región factible la columna.

El siguiente resultado muestra que en algunos problemas, después de un número finito de iteraciones, las columnas se pueden extender fuera de la región factible y el algoritmo sigue siendo convergente.

Proposición 3.3.2 *Bajo las hipótesis del teorema 2.4.13, después de un número finito de iteraciones se pueden prolongar las columnas generadas en el CGP fuera de la región factible y el algoritmo CG/SD sigue siendo convergente.*

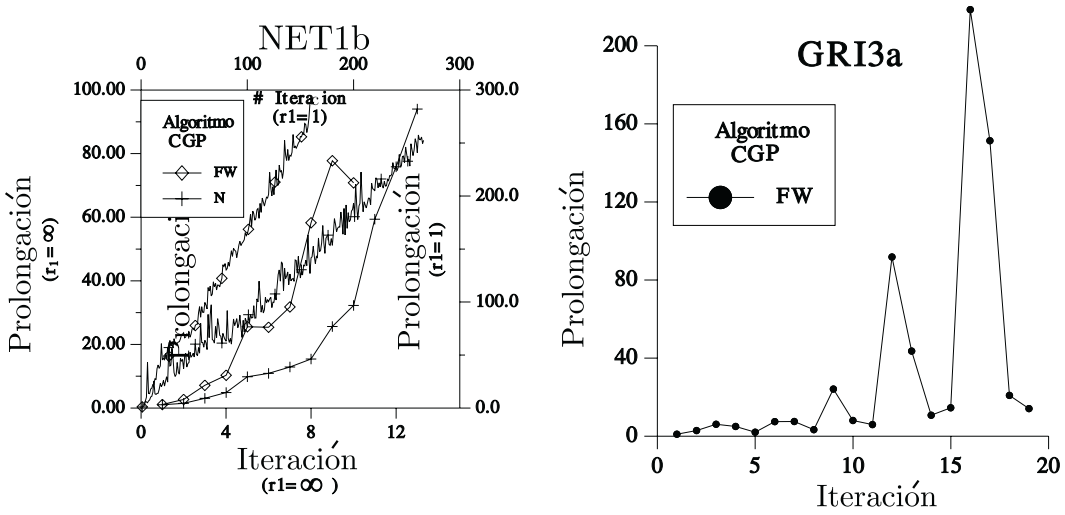


Figura 3.15: Evolución de la prolongación a la frontera en función del número de iteraciones

DEMOSTRACIÓN. Empleando los mismos argumentos de la demostración del teorema 2.4.13 podemos garantizar la existencia de un número entero τ_1 para el que $X^t \subset F^*$ para todo $t > \tau_1$. Esto implica que para cualquier punto $\tilde{\mathbf{y}}^t = \mathbf{x}^t + \ell_t(\hat{\mathbf{y}}^t - \mathbf{x}^t)$, con $\ell_t \geq 1$ se cumple que $\tilde{\mathbf{y}}^t \in \mathtt{aff}\,(F^*)$, donde $\mathtt{aff}\,(F^*)$ es la envoltura afin de F^*. Sea $\tilde{X}^t$ el conjunto obtenido al reemplazar $\mathbf{y}^t$ en $\mathcal{P}^t$ con una columna del tipo $\tilde{\mathbf{y}}^t$. Luego se cumple que $\tilde{X}^t$ es un subconjunto de $\mathtt{aff}\,(F^*)$ para todo $t > \tau_1$.

Ahora demostraremos que existe un entero τ verificando que, si la columna $\mathbf{y}^t$ es reemplazada por $\tilde{\mathbf{y}}^t$ para definir X^t con $t \geq \tau$, la solución de este RMP se alcanzará en un punto de X.

Como $\mathbf{x}^t \to \mathbf{x}^*$ y $\mathbf{x}^* \in \mathtt{rint}\,(F^*)$, entonces existe un τ_2 cumpliendo para todo $t > \tau_2$

$$f(\mathbf{x}^t) < \text{minimizar}_{\mathbf{x} \in \mathtt{rfro}\,(F^*)}\ f(\mathbf{x})$$

Sea $\tau = \max\{\tau_1,\ \tau_2\}$ probaremos que $\text{SOL}(f, \tilde{X}^t) \subset X$ para todo $t > \tau$ por reducción al absurdo. Supongamos que $\mathbf{x}^{t+1} \notin X$. Por hipótesis $\mathbf{x}^{t+1} \in \tilde{X}^t \subset \mathtt{aff}\,(F^*)$, y entonces existe un punto $\mathbf{z} \in [\mathbf{x}^t, \mathbf{x}^{t+1}] \cap \mathtt{rfro}\,(F^*)$ cumpliendo $f(\mathbf{x}^t) \leq f(\mathbf{z})$ y $f(\mathbf{z}) \geq f(\mathbf{x}^{t+1})$ y esto contradice la pseudoconvexidad de f. □

EXPERIMENTO 3.4: geometría de la cara óptima

La proposición anterior asegura que existen relaciones entre la geometría de la cara óptima y la operación de prolongar a la frontera relativa.

En este experimento dibujamos el tamaño de la prolongación ℓ_t en cada iteración t. Hemos considerado $\text{N}_\infty^{10,1}$, $\text{FW}_\infty^{10,10}$, $\text{N}_1^{10,1}$, $\text{FW}_1^{10,10}$ para NET1b y $\text{FW}_\infty^{50,4}$ para la red GRI3a. La primera conclusión es que $\ell_t > 1$ para toda iteración t. Esto significa que tanto la iteración como la columna están en la misma cara.

Los dos dibujos muestran dos situaciones diferentes. En el primero, la solución óptima está localizada en una cara donde las columnas están distribuidas uniformemente a lo largo de su frontera relativa con respecto a la solución óptima. Por esta razón, el valor $\ell_t = \frac{\|\mathbf{y}^t - \mathbf{x}^t\|}{\|\hat{\mathbf{y}}^t - \mathbf{x}^t\|} \to \infty$. En este ejemplo, el numerador, que indica la distancia de la actual iteración a un punto de la frontera relativa, es más o menos constante, mientras que el denominador $\|\hat{\mathbf{y}}^t - \mathbf{x}^t\| \to 0$. En el otro ejemplo, la localización de la solución óptima es diferente. Existen puntos muy cercanos a la frontera relativa, mientras que otros están muy lejos.

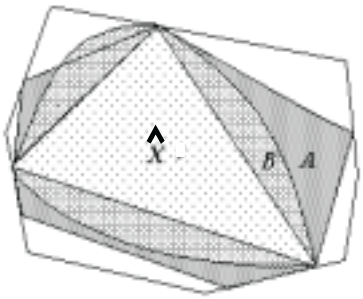


Figura 3.16: Regiones factibles del RMP para los métodos CG/SD

Bloque 4: el papel de X^t

La clase CG/SD permite regiones factibles más generales para los problemas RMP que los algoritmos SD, RSD o NSD. La elaboración de estos nuevos conjuntos se basa en las columnas retenidas, que denominaremos por $\hat{\mathcal{P}}$. El conjunto maximal, que puede ser definido empleando $\hat{\mathcal{P}}$, es $\texttt{aff}\,(\hat{\mathcal{P}}) \cap X$, donde $\texttt{aff}\,(\hat{\mathcal{P}})$ es la envoltura afín del conjunto $\hat{\mathcal{P}}$. A priori, un modo de mejorar los algoritmos CG/SD respecto a los algoritmos SD, RSD o NSD; es considerar una región mayor para el RMP que la clásica región $\hat{X}$, definida como la envoltura convexa de un conjunto de puntos. Esto es, considerar un conjunto convexo y compacto $\tilde{X}$ cumpliendo que $\hat{X} \subset \tilde{X} \subset \texttt{aff}\,(\hat{X}) \cap X$. Esta elección produciría un mayor descenso en el RMP que en la forma estándar empleadas por los métodos SD, RSD or NSD para definir $\hat{X}$.

La desventaja es que también crecería el esfuerzo computacional. Los posibles conjuntos para $\tilde{X}$ deben ser conjuntos poliedrales con un mayor número de puntos extremos que la cardinalidad del conjunto $\hat{\mathcal{P}}$, por la minimalidad de los símplices[2] (esta es la situación A indicada en la figura 3.16), o un conjunto no poliedral (es el caso B de figura 3.16). En ambos casos, el RMP tiene una mayor complejidad computacional. En el primer caso esta mayor complejidad computacional se traduce por un incremento en el número de variables del problema, que coincide con el número de puntos retenidos. En el otro caso, la mayor complejidad radica en pasar de un problema linealmente restringido a uno no lineal.

La anterior discusión previa conduce a la siguiente pregunta: ¿Son ventajosas estas mejores aproximaciones interiores para definir los RMP, teniendo en cuenta que poseen un mayor coste computacional?

En nuestra experiencia computacional, en la mayoría de las iteraciones (aproximadamente 87% para problemas del tipo NET) el RMP no elimina columnas. Este hecho implica que la solución truncada para el RMP (y probablemente la solución óptima) pertenece a $\texttt{rint}\,(X^t)$. Empleando un similar argumento que en el teorema 2.3.6 se puede demostrar que $\mathbf{x}^{t+1}$ resuelve también CDP$(f, \tilde{X}^t)$. Esto significa que el descenso sería el mismo para cualquier otra aproximación interior que contuviese a X^t y estuviera contenida en $\texttt{aff}\,(X^t)$, pero la complejidad computacional sería mayor. Estas consideraciones podrían indicar que la definición de los conjuntos X^t como la envoltura convexa sea la adecuada.

Bloque 5: comparativa de métodos

EXPERIMENTO 5.1: comparativa de los métodos CG/SD, NSD y SD

[2]Si el RMP está resuelto de forma exacta es un símplice y empleando el hecho de que los puntos extremos están en la frontera de un conjunto convexo, este conjunto es minimal respecto al número de puntos extremos

Tabla 3.9: Clasificación de los algoritmos

Tipo	SNFPP	TAP-M
I	FW_1^{8,n_c^t}	$\mathrm{FW}_1^{n_r,n_c}$, $\mathrm{E}_1^{n_r,n_c}$
II	SD, $\mathrm{N}_\infty^{8,1}$, $\mathrm{GLP}_\infty^{8,n_c^t}$	$\mathrm{FW}_{100}^{n_r,1}$, $\mathrm{E}_{100}^{n_r,1}$, $\widehat{\mathrm{NE}}_\infty^{n_r,1}$
III	$\mathrm{FW}_\infty^{8,n_c^t}$	$\mathrm{FW}_\infty^{n_r,n_c}$, $\mathrm{E}_\infty^{n_r,n_c}$
$\mathcal{A}_c$	N, GLP,	FW, E $\widehat{\mathrm{NE}}$

La clase de algoritmos CG/SD generalizan los métodos de direcciones factibles mediante su utilización en el CGP. Decimos que es una extensión ya que el algoritmo original se obtiene tomando $r = 1$ y $n_c = 1$. Nosotros hemos investigado tres formas de mejorar un algoritmo $\mathcal{A}_c$ dado:

I. PARTAN generalizada ($n_c \geq 2$ y $r = 1$). Esta extensión efectúa n_c iteraciones con el algoritmo $\mathcal{A}_c$ y realiza entonces una nueva búsqueda unidimensional en la dirección definida por la actual iteración y la columna generada. Estos métodos son nuevos, exceptuando el caso $n_c = 2$ que se conoce con el nombre de métodos de las tangentes paralelas PARTAN.

II. Búsquedas multidimensionales de la clase NSD ($n_c = 1$ y $r \geq 2$). La elección de $n_c = 1$ y $r \geq 2$ es una forma de generalizar los métodos de búsquedas unidimensionales a búsquedas sobre símplices. Ejemplos de estos algoritmos son los de la clase NSD.

III. Búsquedas multidimensionales de la clase CG/SD. ($n_c \geq 2$ y $r \geq 2$). Estos algoritmos son nuevos y constituyen propiamente la clase CG/SD.

Este experimento ha sido diseñado para comparar los algoritmos de direcciones factibles con sus extensiones del tipo I, II y III. Para este fin, hemos resuelto los problemas SNFP y TAP-M empleando los algoritmos mostrados en la tabla 3.9. Algunos de estos métodos tienen disponibles cotas inferiores a la solución del problema y existen diferencias importantes entre la calidad de éstas. Para eliminar este factor en la medida de la eficiencia de los algoritmos, hemos empleado una cota inferior común para todos ellos. La precisión empleada para los problemas de tipo SNFPP se muestra en la tabla 3.3 y para los problemas del tipo TAP-M hemos usado dos niveles de precisión: ($1.D-03$ y $1.D-04$).

Hemos empleado $n_r = 8$ proyecciones para resolver los RMP para los problemas del tipo NET y el valor $n_r = 25$ para las otra clase de problemas del tipo SNFP. Hemos usado la actualización dinámica para todos los algoritmos empleados en los problemas SNFP, incluso hemos hecho la modificación oportuna para incluir el caso $r = 1$. Por otro lado, hemos utilizado valores fijos de n_c para los problemas del tipo TAP-M. Hemos empleado los valores $n_c = 15$ para Sif2 y el valor $n_c = 5$ para los problemas Hul2 y NgD2. Hemos elegido $n_r = 10$ proyecciones para resolver el RMP.

Los problemas cuadráticos, que aparecen en los algoritmos NSD $\widehat{\mathrm{NE}}_\infty^{n_r,1}$ y $\widehat{\mathrm{NE}}$, se resolvieron con el algoritmo FW, realizando el mismo número de iteraciones que para los algoritmos CG/SD, esto es, n_c.

Los resultados obtenidos para los problemas SNFP se muestran en la tabla 3.10 y para los problemas del tipo TAP-M en la tabla 3.11. En la tabla 3.10 se observa el tiempo de CPU y el número de columnas retenidas en el último RMP y en la tabla 3.11 se muestra, en el primer bloque, los métodos de direcciones factibles y en el segundo, sus generalizaciones a búsquedas sobre símplices. Los resultados de este experimento son el tiempo de CPU, el número de iteraciones principales, el número de columnas retenidas en el último RMP y el número total de iteraciones realizadas por el algoritmo $\mathcal{A}_c$.

Analizaremos los resultados derivados de la tabla 3.10 y de la tabla 3.11 en función del tipo de algoritmo (I, II y III).

I. No es posible comparar el algoritmo de FW con sus extensiones FW_1^{8,n_c^t} o $\mathrm{FW}_1^{n_r,n_c}$ debido a que el tiempo de CPU empleado por el FW para resolver los problemas de prueba es prohibitivo. Si comparamos la descomposición simplicial para los problemas NET-SNFP y la descomposición simplicial restringida $\mathrm{FW}_{100}^{10,1}$ para los problemas TAP-M, entonces la media armónica de los ratios de los cocientes de tiempo de CPU es 17.2 y 1.5 respectivamente, favorable a la extensión unidimensional del método de FW. Por otro lado, es posible comparar el algoritmo $\mathrm{E}_1^{n_r,n_c}$ con su algoritmo de referencia E. La media armónica de los ratios es de 3.4 para la precisión de 10^{-3} y de 5.6 para la precisión 10^{-4}. También $\mathrm{E}_1^{n_r,n_c}$ mejora la descomposición simplicial con subproblemas de Evans, esto es, $\mathrm{E}_{100}^{10,1}$. La tasa media de mejora es de 1.9 para la precisión de 10^{-3} y 1.2 para la precisión de 10^{-4}.

II. Hemos calculado la media armónica de los ratios entre los algoritmos unidimensionales y sus extensiones NSD. Hemos considerado los algoritmos N y GLP para los problemas SNFP y $\widehat{\mathrm{NE}}$ para los problemas TAP-M. Esta media para los problemas del tipo NET es 5.6 y 6.4 para N y GLP respectivamente, y 1.1 y 1.8 para los problemas GRID-AUT. En este tipo de problemas, existen ejemplos donde las búsquedas multidimensionales no mejoran los métodos de direcciones factibles. Esto se debe a dos factores, el primero es la gran cantidad de cotas activas en las soluciones de estos ejemplos (ver la tabla 3.3) y esto hace ineficiente la expansión de los símplices. El segundo factor es la inadecuada precisión elegida para resolver los RMP, que es demasiado grande.

   El $\widehat{\mathrm{NE}}$ se mejora 3.8 veces su tiempo de CPU para una precisión 10^{-3} y alrededor de 20.35 para la precisión 10^{-4}.

   En general, los métodos CG/SD mejoran su versión unidimensional, siendo mayor para precisiones altas.

III. Los algoritmos de tipo III, $\mathrm{FW}_\infty^{8,n_c^t}$, $\mathrm{FW}_\infty^{n_r,n_c}$ y $\mathrm{E}_\infty^{n_r,n_c}$ han presentado el mejor comportamiento computacional para los problemas del tipo NET y TAP-M . En la tabla 3.10 se ve que el tiempo de CPU del algoritmo SD se mejora por el algoritmo $\mathrm{FW}_\infty^{8,n_c^t}$ mediante un factor de 113.2 para los problemas del tipo NET. Este factor es de 10.3 para los problemas TAP-M. Si comparamos $\mathrm{FW}_{100}^{10,1}$ (descomposición simplicial restringida) con $\mathrm{E}_\infty^{n_r,n_c}$ se obtiene que este factor es de 24.9. En realidad, deberíamos comparar el algoritmo $\mathrm{E}_\infty^{n_r,n_c}$ con $\mathrm{E}_{100}^{10,1}$, siendo este factor de 4.3. El método SD es satisfactorio para precisiones medianas o pequeñas, pero es de mucha exigencia computacional cuando la precisión demandada es alta, haciendo que los procedimientos sean imposibles de aplicar en un tiempo razonable. Los métodos CG/SD son robustos para este efecto.

EXPERIMENTO 5.2: cotas inferiores para los métodos CG/SD.

El experimento anterior es una imagen de la convergencia de los métodos para un nivel de precisión dado. Un método podría ser mejor que otro al comienzo de las iteraciones y transcurridas unas cuantas, la situación inicial podría ser invertida.

La convergencia de los métodos se monitoriza por las cotas inferiores a lo largo del proceso. Éstas son el valor óptimo de los subproblemas de Frank-Wolfe o Evans.

En el experimento anterior, todos los algoritmos operan con la misma cota inferior. Éste está diseñado para ilustrar estas dos cuestiones: la evolución de las cotas generadas por los métodos y su convergencia.

La figura 3.17 muestra la evolución de las cotas inferiores y de la sucesión $f(\mathbf{x}^t)$ generada por los métodos FW, E, $\mathrm{FW}_\infty^{10,1}$, $\mathrm{E}_\infty^{10,1}$, $\mathrm{FW}_\infty^{10,n_c}$, $\mathrm{E}_\infty^{10,n_c}$ frente al número de iteraciones y tiempo de CPU en el problema Sif2.

La principal conclusión es que los métodos $\mathrm{E}_\infty^{10,n_c}$, $\mathrm{FW}_\infty^{10,n_c}$ son mejores que los otros (tanto frente a número de iteraciones como tiempo empleado de CPU) y generan también mejores cotas inferiores a lo largo del proceso. Los métodos E y $\mathrm{E}_\infty^{10,1}$ son mejores que sus equivalentes FW y $\mathrm{FW}_\infty^{10,1}$, pero éstos solamente generan mejores cotas al principio, posteriormente la tendencia es invertida.

Tabla 3.10: Resultado experimentales de los problemas SNFP

Red	SD	FW_1^{8,n_c^t}	N	GLP	FW_∞^{8,n_c^t}	N_∞^{8,n_c^t}	GLP_∞^{8,n_c^t}
NET1a	260.9[a]	20.2	7.5	33.6	18.7	10.5	20.2
	34[b]	2	2	2	4	3	6
NET1b	2056.3	585.8	607.1	1624.6	37.1	56.4	104.2
	88	2	2	2	6	9	16
NET1c	4686.1	544.1	=	663.0	31.6	=	45.2
	120	2	FW_1^{8,n_c^t}	2	9	FW_∞^{8,n_c^t}	13
NET2a	> 11000	1106.8	4003.7	1434.3	92.5	80.5	111.9
	> 100	2	2	2	8	7	8
NET2b	> 44000	1685.5	2360.4	3552.3	93.4	207.6	293.5
	> 130	2	2	2	8	12	17
NET3a	6407.9	136.5	=	264.0	21.2	=	24.9
	111	2	FW_1^{8,n_c^t}	2	13	FW_∞^{8,n_c^t}	13
NET3b	665.9	21.6	=	103.0	13.3	=	52.9
	49	2	FW_1^{8,n_c^t}	2	10	FW_∞^{8,n_c^t}	10
NET4a	> 17110	471.7	=	922.6	98.5	=	110.5
	> 83	2	FW_1^{8,n_c^t}	2	15	FW_∞^{8,n_c^t}	16
NET4b	> 15550	154.6	=	1442.6	81.6	=	918.7
	> 100	2	FW_1^{8,n_c^t}	2	11	FW_∞^{8,n_c^t}	27
Red		FW_1^{25,n_c^t}	N	GLP	FW_∞^{25,n_c^t}	N_∞^{25,n_c^t}	GLP_∞^{25,n_c^t}
GRI2a		227.1	266.2	1604.8	102.6	126.2	94.6
		2	2	2	10	8	11
GRI2b		93.7	116.2	256.6	37.2	62.8	35.4
		2	2	2	11	13	10
GRI3a		76.6	43.0	263.7	59.7	297.9	1668.0
		2	2	2	4	8	17
AUT1		29.8	24.9	22.9	12.8	6.8	6.6
		2	2	2	7	5	5
AUT2		45.0	92.3	47.1	59.5	81.8	71.3
		2	2	2	5	9	8
AUT3		31.9	18.7	24.3	33.3	20.3	27.5
		2	2	2	5	3	4

[a]Tiempo de CPU.
[b]Número de puntos retenidos en el último RMP.

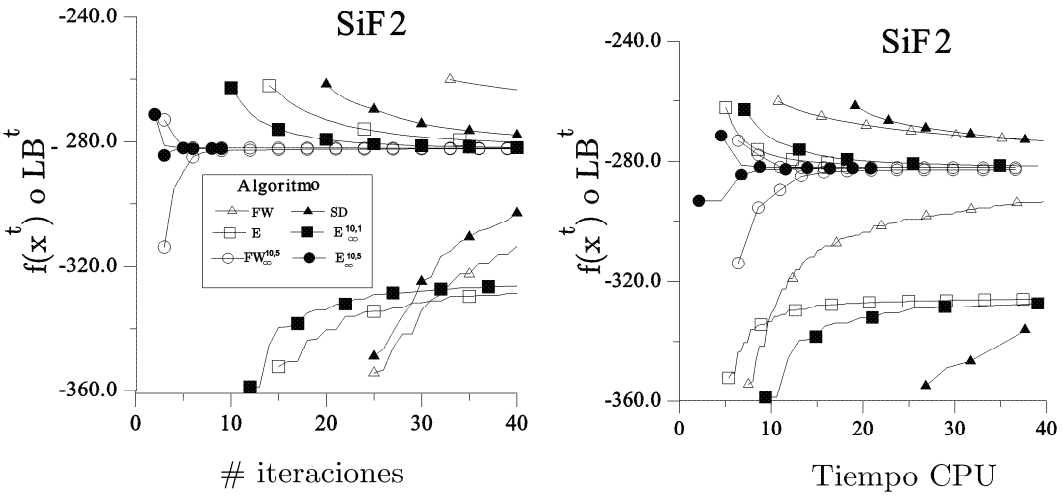


Figura 3.17: Evolución de $f(\mathbf{x}^t)$ y de la cota inferior LB^t frente al número de iteraciones

Tabla 3.11: Resultados experimentales para los problemas TAP-M

Precisión	Red	FW	E	$\widehat{\text{NE}}$	FW_1^{10,n_c}	E_1^{10,n_c}	$\text{FW}_{100}^{10,1}$	$\text{E}_{100}^{10,1}$	$\widehat{\text{NE}}_\infty^{10,1}$	$\text{FW}_\infty^{10,n_c}$	E_∞^{10,n_c}
	NgD2	3.62[a]	1.37	0.33	0.60	0.28	0.33	0.22	0.11	0.22	0.11
		445[b]	167	29	29	15	21	16	9	9	6
		–[c]	–	–	–	–	16	14	8	8	6
		445[d]	167	145	145	75	21	16	45	45	30
0.1%	SiF2	>72.5	49.8	310.3	313.1	11.3	> 1616.9	56.2	34.9	13.4	11.7
		>2000	124	141	136	5	>100	72	93	70	8
		–	–	–	–	–	>78	31	82	65	3
		>2000	124	141	136	75	>100	72	1395	1050	120
	Hul2	89.2	16.6	20.9	22.5	8.6	44.1	16.8	10.2	10.1	6.2
		231	47	23	20	7	32	19	10	10	5
		–	–	–	–	–	28	17	9	9	2
		231	47	115	100	35	32	17	50	35	25
	NgD2	144.56	39.32	15.22	24.44	6.59	0.60	1.04	0.55	0.77	0.72
		14531	4009	1073	1119	294	30	36	22	22	21
		–	–	–	–	–	23	27	21	21	20
		14531	4009	5365	5595	1470	30	36	110	110	105
0.01%	SiF2	?	394.9	?	>2346.5	48.5	?	222.3	1682.7	942.6	18.9
			848		>1000	21	72	93	70	8	
		?	–	?	–	–	?	31	82	65	3
			848		>1000	315		72	1395	1050	120
	Hul2	?	639.1	990.3	1291.2	185.5	> 30000	423.4	67.8	67.3	25.9
			1472	1067	1053	143	> 1000	83	36	36	17
		?	–	–	–	–	?	54	31	31	12
			1472	5215	5085	715	> 1000	83	180	180	85

[a] CPU.
[b] Número de iteraciones principales.
[c] Número de puntos retenidos en el último RMP.
[d] Número total de iteraciones realizadas por el algoritmo $\mathcal{A}_c$.

Capítulo 4

Calibración de parámetros y estimación de matrices origen destino en modelos combinados

Resumen

En las últimas dos décadas se han realizado importantes avances en la formulación y análisis de modelos combinados. Para poder usar muchos de estos modelos se requiere conocer una matriz de demanda origen-destino (O-D) y un vector de parámetros. Tradicionalmente, la obtención de esta información ha conducido a dos problemas independientes. El primero es el problema de calibración de los parámetros del modelo y el segundo el problema de estimación de matrices O-D.

En este capítulo se ha analizado el problema de la calibración los parámetros del modelo combinado TAP-M. Esta estimación se formula en un contexto binivel y se analiza el problema de *sobrespecificación de los parámetros*. Se ha realizado una experiencia numérica para analizar diversas posibilidades para la métrica de la función objetivo del nivel superior.

El modelo de calibración desarrollado ha sido generalizado para poder estimar simultáneamente las matrices O-D y calibrar los parámetros. Este modelo, abreviadamente denominado CDAM, emplea, como datos de entrada, un subconjunto de aforos de la red multimodal, una matriz O-D desactualizada y/o los resultados de una encuesta de movilidad. Se ha demostrado que el CDAM tiene solución incluso cuando las observaciones (en los arcos o en las matrices O-D) son inconsistentes o incompletas.

Se ha desarrollado un marco para la elaboración de algoritmos heurísticos de resolución del CDAM basado en la generación de una secuencia de problemas de optimización binivel, mucho más fáciles de resolver que el original. Se ha demostrado que el límite de esta sucesión, en caso de tener convergencia finita, es un óptimo local del problema CDAM.

Palabras clave:

Estimación de matrices O-D, calibración de parámetros, modelos combinados en redes de equilibrio, programación matemática binivel, algoritmos heurísticos.

4.1 Introducción

La promoción de los viajes combinados requiere de herramientas que auxilien la toma de decisiones. Los modelos de equilibrio con modos combinados son herramientas adecuadas en una toma de decisiones cuyo horizonte temporal sea el de la planificación táctica. Estos modelos no han recibido mucha atención en la literatura, en parte por la dificultad de modelar simultáneamente la elección del modo de transporte, ruta y nodos de transferencia modal. En el capítulo 1 se ha desarrollado un modelo que aúna estas decisiones, el denominado TAP-M. En este capítulo desarrollaremos una metodología para estimar la matriz de viajes O-D y el vector de parámetros del TAP-M. Hemos considerado que la función de costes en los arcos tenga una matriz Jacobiana simétrica.

Una de las principales tareas en las aplicaciones, es desarrollar una metodología para obtener los datos necesarios para la utilización del modelo. Éstos son un conjunto de parámetros y una matriz de demanda de viajes O-D para el total de las alternativas consideradas por el TAP-M. Asumimos que la red de transporte y sus funciones de costes son conocidas. Tradicionalmente la obtención de esta información ha conducido a dos problemas independientes. El primero es el problema de la calibración del modelo y el segundo es el problema de estimación de la matriz de demanda O-D (DAM).

Ortúzar y Willumsem [189] consideraron dos metodologías básicas para el problema de calibración de los modelos de transporte: no estructurada y estructurada.

- ◇ La metodología no estructurada se basa en los métodos de estimación de la estadística clásica, tales como: la estimación de máxima verosimilitud, la maximización de la entropía, la estimación mínimo cuadrática, etc. Estos modelos requieren de información adicional obtenida mediante la realización de encuestas y ajustan el valor de los parámetros con las observaciones muestrales, ayudados de ciertas métricas.
- ◇ La metodología estructurada restringe la región factible de los parámetros imponiendo que exista una estructura especial, asumiendo usualmente que existe un modelo de transporte definido por este conjunto de parámetros. La formulación del problema de calibración mediante este enfoque se basa en la programación matemática binivel. En el nivel superior se plantea un problema clásico de estimación (como en la metodología anterior) y en el nivel inferior, definido por un modelo de transporte, se obtiene la información necesaria (predicciones del modelo de transporte) para realizar la estimación en el nivel superior. La estructura binivel para el problema de calibración está presente en los modelos de Boyce [30], Zhang [249], Abrahamsson y Lundqvist [5] y Ottomanelli [190] (entre otros).

En este trabajo se ha asumido un enfoque estructurado para el problema de calibración del TAP-M. Un problema presente en ambas metodologías es el llamado *problema de sobrespecificación* de los parámetros, que se basa en la existencia de más parámetros del modelo de los necesarios y esto produce infinitas soluciones al problema de calibración. Hemos caracterizado este problema en la calibración del TAP-M, esto es debido al uso de un modelo logit anidado como modelo de demanda y a la estructura de los costes combinados. La sobrespecificación debe ser evitada debido a que la velocidad de convergencia depende críticamente de este fenómeno, incluso para los métodos más robustos. La sobrespecificación produce que la matriz Hessiana de las métricas sea singular. Bierlaire y otros [26] han analizado el problema de sobrespecificación en la estimación máximo verosímil de modelos logit anidados. Estos autores analizaron la relación entre dos estrategias arbitrarias para evitar la sobrespecificación y mostraron que una debía ser una transformación lineal de la otra. Algunos métodos de optimización, como el método de Newton o los métodos de quasi-Newton de la familia de Broyden con búsquedas lineales, son invariantes frente a transformaciones lineales. Si empleásemos cualquiera de estos métodos en la resolución del problema (no estructurado) de calibración, sería irrelevante el modo en que se haya eliminado el efecto de la sobrespecificación. Daganzo y Kusnic [62] sugirieron igualar un parámetro a cero para cada conjunto involucrado en cada fuente de sobrespecificación y entonces estimar el resto de los parámetros. Se han caracterizado tres fuentes de sobrespecificación para el TAP-M y una simple generalización de este principio se puede aplicar para eliminar la sobrespecificación.

El otro dato necesario para emplear el TAP-M es una matriz de demanda O-D para las alternativas

presentes. La matriz O-D es la suma de una matriz O-D para el modo coche, *park'n ride* y metro (que son las tres alternativas que se han considerado en este trabajo). Los métodos convencionales para la obtención de esta información se basan en la realización de encuestas de movilidad, las cuales son económicamente costosas, o mediante la obtención del potencial de atracción o generación de viajes de las zonas del estudio (*fase de generación*), a partir de bases de datos socioeconómicos y posteriormente mediante otros modelos matemáticos (por ejemplo mediante la maximización de la entropía) se determina la matriz O-D (*fase de distribución*). En la actualidad, varios modelos de programación matemática binivel han sido empleados para la estimación de estas matrices O-D en redes congestionadas, para un mayor detalle consúltese las revisiones bibliográficas de Chen y Florian [48] y Barceló [11]. Estos esquemas integran en un único proceso el modelo convencional de estimación y el modelo de asignación en equilibrio. Los datos empleados por estos modelos son aforos de tráfico (obtenidos mediante líneas cordón) y una matriz O-D desactualizada u obtenida por otro procedimiento

En este capítulo, se han unificado dos modelos de programación matemática binivel, uno de ellos utilizado en la calibración y el otro en la estimación de matrices de demanda O-D, para abordar ambos problema simultáneamente. Este modelo se denomina CDAM. Si la matriz O-D fuese conocida, el CDAM define un nuevo modelo para la calibración de modelos combinados. La ventaja del CDAM es que emplea mayor información que los esquemas clásicos como son los aforos en la red multimodal. Si el vector de parámetros fuese conocido el modelo es transformado al problema de estimación/actualización de matrices O-D como el presentado en Lundgren [153], Chen y Florian [48], etc. aplicado al TAP-M. La discusión realizada se circunscribe al modelo TAP-M pero fácilmente es exportable a otros modelos combinados.

El CDAM, decide en el nivel superior el valor de los parámetros y la matriz O-D; y en el nivel inferior se generan los flujos, costes y la correspondiente partición modal para la situación de equilibrio del TAP-M, en función de las variables del nivel superior. El nivel superior compara estas predicciones realizadas por el TAP-M con los aforos observados, la partición modal y la matriz O-D de referencia.

Los problemas de programación matemática binivel son en general difíciles de resolver debido a su no convexidad y no diferenciabilidad. Además, una evaluación de la función objetivo del nivel superior requiere de la resolución de un problema de optimización de gran escala en el nivel inferior. Debido a que no existen algoritmos de resolución para problemas binivel de grandes dimensiones se ha recurrido a diversos heurísticos. El más extendido consiste en iterar entre el nivel superior y el nivel inferior, considerando en cada nivel las variables del otro nivel fijas. Este tipo de algoritmos ha sido aplicado al problema de diseño de redes o para el problema de control de tráfico (Gartner [110]). Estos métodos no siempre conducen a la solución óptima del problema como ilustró Tan y otros [225] y demostró teóricamente Marcotte [157]. Este tipo de algoritmo ha sido propuesto en este trabajo para resolver el modelo binivel de calibración planteado.

En la literatura, varios algoritmos han sido desarrollados para el problema de estimación de matrices O-D (DAM) para redes de tráfico congestionadas. Spiess [219] formula un modelo basado en mediciones de volúmenes de tráfico y desarrolla un algoritmo heurístico para su resolución. Este algoritmo se basa en la suposición de que la proporción de usuarios en cada arco y para cada demanda es (localmente) constante. Esta proporción es generada implícitamente durante la resolución del modelo de equilibrio a través de los caminos generados y de las búsquedas lineales obtenidas. Este algoritmo ha sido aplicado a problemas de grandes dimensiones. Los parámetros del TAP-M producen una adaptación no trivial de este algoritmo. Yang y otros [245] introducen una matriz desactualizada en la formulación del problema de estimación/actualización de matrices O-D y proponen dos modelos, uno basado en la métrica de mínimos cuadrados generalizados empleada por Cascetta y Nguyen en [40] (GLS) y el otro modelo basado en la maximización de la entropía. Yang y otros desarrollan un algoritmo heurístico de tipo optimización-asignación. Florian y Chen [81] proponen un algoritmo heurístico que no emplea explícitamente la información de los caminos óptimos, evitando los requerimientos de enumerar caminos y proporciones. Se han reportado experimentos numéricos con redes de grandes dimensiones en Florian y Chen [81] y Chen [47]. Yang [241] presenta dos algoritmos heurísticos similares al algoritmo de optimización-asignación, en el sentido de que se resuelve iterativamente un problema de asignación y luego uno de optimización para determinar un nuevo valor para la matriz O-D, pero

existe una diferencia significativa. Si consideramos este problema como un caso particular del juego de Stackelberg (y su estructura líder-seguidores), estos algoritmos tienen en cuenta en el problema de estimación de la demanda una aproximación de la reacción de los seguidores a las decisiones del líder.

Codina y Barceló [53] desarrollan una adaptación del algoritmo de Wolfe para programación matemática no diferenciable al problema DAM. Este método ha reportado mejores propiedades de convergencia que los algoritmos de Spiess y del método del descenso más rápido.

En este capítulo, se ha propuesto un marco para la elaboración de algoritmos para el CDAM y se muestra que los dos algoritmos de Yang [241] pertenecen a esta clase. Se ha dado una condición suficiente para garantizar que el límite de la sucesión generada por el algoritmo es un óptimo local para el CDAM.

4.1.1 Los datos del TAP-M

En este apartado introducimos la notación que usaremos en este capítulo para denotar los datos empleados por el TAP-M y centramos el problema objeto de estudio.

Consideramos que los usuarios eligen tres alternativas de transporte: (a) en vehículo privado (coche), (b) en transporte público (metro-cercanías) o (c) en modo combinado (*park'n ride*). Recordar que la partición modal se producía mediante funciones logit que nos daban las distintas proporciones de la manera siguiente

$$G_\omega^k(\mathbf{U}_\omega^*) = \frac{\exp -\left(\alpha^k + \beta_1 U_\omega^{k*}\right)}{\sum_{k' \in \{a,b,c\}} \exp -\left(\alpha^{k'} + \beta_1 U_\omega^{k'*}\right)}, \quad k \in \{a,b,c\} \tag{4.1}$$

donde U_ω^{c*} es igual "log-suma" del coste de viaje en modo combinado a través de todos los nodos de transferencia T_ω y éste se calcula por

$$U_\omega^{c*} = -(1/\beta_2) \ln \left(\sum_{t \in T_\omega} \exp -\beta_2 \{\alpha_t^c/\beta_2 + U_{\omega,t}^{c*}\} \right), \quad \omega \in W. \tag{4.2}$$

El modelo, explícitamente, tiene en cuenta la elección del nodo de transferencia mediante otra función logit adicional que produce la desagregación de los viajes combinados, a través de los intercambiadores, de acuerdo con la expresión

$$G_{\omega,t}^c(\mathbf{U}_\omega^{c*}) = \frac{\exp -\left(\alpha_t^c + \beta_2 U_{\omega,t}^{c*}\right)}{\sum_{t' \in T_\omega} \exp -\left(\alpha_{t'}^c + \beta_2 U_{\omega,t'}^{c*}\right)} \tag{4.3}$$

donde $U_{\omega,t}^{c*}$ representa la percepción del coste generalizado de la demanda ω mediante viaje combinado a través del nodo de transferencia t. El modelo de demanda es un modelo logit anidado donde los costes generalizados para el *park'n ride* se calculan como "log-suma" del coste combinado para todos los nodos de intercambio.

En esencia el modelo TAP-M, empleado en este capítulo, coincide con el desarrollado por Fernández y otros [73], pero se ha añadido la alternativa pura modo *transporte público* donde los usuarios van a pie a una parada de transporte público y emplean una o varias líneas hasta la parada de destino, luego completan el viaje andando. El interés por incluir esta alternativa, no es sólo por la importancia para modelar la realidad, sino porque es necesario para poder utilizar las observaciones de flujo en los arcos de la red de transporte público, ya que no es posible observar (sin realizar encuestas) sólo a los usuarios en la red de transporte público que realizan viajes combinados.

El modelo incluye los parámetros θ_a y θ_b para homogeneizar los costes en cada red de transporte. Los costes percibidos en la red de tráfico privado y en la red de transporte público son:

$$U_\omega^{a*} = \theta_a C_\omega^{a*} \text{ y } U_\omega^{b*} = \theta_a C_\omega^{b*}$$

donde C_ω^{a*} y C_ω^{b*} son los costes de equilibrio mediante coche y transporte público respectivamente. Para los usuarios de viajes combinados este coste es

$$U_{\omega,t}^{c*} = \theta_a C_{\omega,t}^{a*} + \theta_b C_{t,j}^{b*}$$

donde $C_{\omega,t}^{a*}$ es el coste del viaje de i a t para el par $\omega = (i,j)$. Bajo la hipótesis

$$\alpha^k > 0 \text{ y } 0 < \beta_1 < \beta_2 \tag{4.4}$$

se demuestra, en el apéndice III del capítulo 1, que la situación de equilibrio puede ser representada como la solución del siguiente problema de optimización

$$\begin{array}{ll} \text{minimizar} & Z(\mathbf{f},\mathbf{g}) = S(\mathbf{f},\Theta) + R(\mathbf{g},\Theta) \\ \text{sujeto a} & (\mathbf{f},\mathbf{g}) \in \tilde{\Omega}(\bar{\mathbf{g}}), \end{array} \qquad \text{[TAP-M]}$$

donde

$$S(\mathbf{f},\Theta) = \theta_a \sum_{l\in A} \int_0^{f_l} c_l(x)dx + \theta_b \sum_{l\in B} \int_0^{f_l} c_l(x)dx$$

es el coste asociado con el flujo en los arcos y

$$\begin{aligned} R(\mathbf{g},\Theta) = \quad & (1/\beta_1) \sum_{k\in\{a,b,c\}} \sum_{\omega\in W} g_\omega^k (\ln g_\omega^k - 1 + \alpha^k) - (1/\beta_2) \sum_{\omega\in W} g_\omega^c (\ln g_\omega^c - 1) \\ & + (1/\beta_2) \sum_{\omega\in W} \sum_{t\in T_\omega} g_{\omega,t}^c (\ln g_{\omega,t}^c - 1 + \alpha_t^c) \end{aligned}$$

es el coste asociado con el modelo de partición de la demanda. $\tilde{\Omega}(\bar{\mathbf{g}})$, que fue definido en la página 64, es la región factible (espacio de flujo en los arcos y desagregación de la demanda) parametrizada respecto a la matriz de demanda O-D $\bar{\mathbf{g}}$. Denotamos Θ el vector de los parámetros de las funciones logit y los parámetros de ponderación de costes θ_a y θ_b en la función objetivo. El problema de calibración estima este vector Θ y el problema de estimación de la matriz de demanda de viajes O-D el vector $\bar{\mathbf{g}}$.

4.2 Sobre la calibración del TAP-M

4.2.1 El problema de calibración

En el esquema propuesto en este apartado para realizar la calibración del TAP-M, se asume que se ha realizado una encuesta de movilidad y se han observado aleatoriamente un total de n viajes, de los cuales n_ω^k son para el par ω en el modo $k \in \{a,b\}$ y $n_{\omega,t}^c$ son para el par ω en el modo combinado c, vía nodo de transferencia t. Suponemos que las tasas de ocupación γ_ω y la demanda total $\bar{g}_\omega$ son conocidas para todos los pares ω.

El modelo se formula como un modelo de programación matemática binivel. En el nivel superior, donde los costes de equilibrio son conocidos, se establece un problema clásico (no estructurado) de estimación. En el nivel inferior, el TAP-M genera los costes de equilibrio para el vector de parámetros decidido en el nivel superior. El modelo puede ser expresado por

Nivel superior→ **Problema de estimación:** los costes de equilibrio $\mathbf{C}_\omega$ son conocidos, genera Θ

Nivel inferior→ **Modelo TAP-M**: Θ es conocido, genera $C_\omega^a(\Theta)$, $C_\omega^b(\Theta)$, $C_{it}^a(\Theta)$, $C_{tj}^b(\Theta)$

Los métodos clásicos de estimación, como son el de máxima verosimilitud (ML), el de mínimos cuadrado (NLLS), el de mínimos cuadrados ponderados (WNLLS) y maximización de la entropía (ME)

conducen a las siguientes funciones objetivo del nivel superior

(ML) :

$$\max\ \ln\ L = \sum_{\omega} \left(\sum_{k\in\{a,b,c\}} n_{\omega}^{k} \ln G_{\omega}^{k} + \sum_{t\in T_{\omega}} n_{\omega,t}^{c} \ln G_{\omega,t}^{c} \right)$$

(NLLS):

$$\min\ S_1 = \sum_{\omega} \left(\sum_{k\in\{a,b\}} \left(n_{\omega}^{k} - n_{\omega} G_{\omega}^{k}\right)^2 + \sum_{t\in T_{\omega}} \left(n_{\omega,t}^{c} - n_{\omega}^{c} G_{\omega}^{c}\right)^2 \right)$$

(WNLLS):

$$\min\ S_2 = \sum_{\omega} \left(\sum_{k\in\{a,b\}} \frac{\left(n_{\omega}^{k} - n_{\omega} G_{\omega}^{k}\right)^2}{n_{\omega}^{k}} + \sum_{t\in T_{\omega}} \frac{\left(n_{\omega,t}^{c} - n_{\omega,t}^{c} G_{\omega}^{c}\right)^2}{n_{\omega}^{c}} \right)$$

(ME):

$$\max\ S_3 = -\sum_{\omega} \left(\sum_{k\in\{a,b\}} n_{\omega} G_{\omega}^{k} \ln \frac{n_{\omega} G_{\omega}^{k}}{n_{\omega}^{k}} - n_{\omega} G_{\omega}^{k} + \sum_{t\in T_{\omega}} n_{\omega} G_{\omega}^{c} G_{\omega,t}^{c} \ln \frac{n_{\omega} G_{\omega}^{c} G_{\omega,t}^{c}}{n_{\omega,t}^{c}} - n_{\omega} G_{\omega}^{c} G_{\omega,t}^{c} \right)$$

donde $n_{\omega} = n_{\omega}^{a} + n_{\omega}^{b} + \sum_{t\in T_{\omega}} n_{\omega,t}^{c}$, y los valores G_{ω}^{k} y $G_{\omega,t}^{c}$ dependen de Θ, C_{ω}^{a}, C_{ω}^{b}, C_{it}^{a} y C_{tj}^{b}.

4.2.2 Sobre la sobrespecificación de los parámetros

Cualquier métrica razonable empleada en el nivel superior, tal como las ilustradas anteriormente, está basada en el conocimiento los valores G_{ω}^{k} y $G_{\omega,t}^{c}$. En este apartado, mostramos que no existe solución única al problema de calibración para dicho caso. Esto es debido a la sobrespecificación de los parámetros del modelo.

Bierlaire y otros [26] han analizado la sobrespecificación de los parámetros del modelo logit anidado en la estimación de máxima verosimilitud, probaron que la función de verosimilitud asociada con el modelo logit anidado es constante en un subespacio de dimensión $|\mathcal{S}| + 1$ donde $\mathcal{S}$ es el conjunto de las alternativas que poseen subalternativas. En el modelo TAP-M sólo la alternativa *park'n ride* posee subalternativas (elección del nodo de transferencia) y por tantos este subespacio tiene dimensión 2.

Las proposiciones 4.2.1 y 4.2.2 caracterizan este subespacio. En la primera proposición se muestra que la suma de una misma cantidad a las utilidades de todas las alternativas de modo de transporte, no afecta al valor de la métrica, suponiendo que está basada en las cantidades G_{ω}^{k} y $G_{\omega,t}^{c}$, y en particular no afecta a la verosimilitud de la muestra. La proposición 4.2.2 caracteriza la segunda fuente de sobrespecificación que se debe a la función de utilidad de los viajes combinados definida en (4.2). Se muestra que si una cantidad se suma a todas las funciones de las utilidades de las subalternativas y es restada de la utilidad de la alternativa en modo combinado, entonces la utilidad de la alternativa de viaje combinado no varía y por tanto la métrica es constante en este espacio de transformaciones.

PROPOSICIÓN 4.2.1 *Sean α^a, α^b, α^c valores reales. Si reemplazamos estos valores por: $\alpha'^a = \alpha^a + \gamma$, $\alpha'^b = \alpha^b + \gamma$ y $\alpha'^c = \alpha^c + \gamma$ entonces las funciones (4.1) y (4.3) mantienen su valor.*

DEMOSTRACIÓN. Denotado por Θ el vector de parámetros y por Θ' el mismo vector que Θ con la excepción de que las coordenadas α^a, α^b, α^c han sido reemplazadas por α'^a, α'^b, α'^c.

$$G_{\omega}^{k}(\Theta') = \frac{\exp(-\gamma)\exp -\left(\alpha^k + \beta_1 U_{\omega}^{k*}\right)}{\exp(-\gamma)\sum_{k\in\{a,b,c\}}\exp -\left(\alpha^k + \beta_1 U_{\omega}^{k*}\right)} = G_{\omega}^{k}(\Theta)$$

Es obvio que $G_{\omega,t}^{c}(\Theta') = G_{\omega,t}^{c}(\Theta)$. ∎

PROPOSICIÓN 4.2.2 *Sean α_t^c y α^c valores reales. Si reemplazamos estos valores por: $\alpha'^c_t = \alpha_t^c + \gamma$ y $\alpha'^c = \alpha^c - \gamma\frac{\beta_1}{\beta_2}$ entonces las funciones logit (4.1) y (4.3) no varían sus valores.*

DEMOSTRACIÓN. Emplearemos una notación similar Θ y Θ' como en la proposición 4.2.1. Para probar que $G_\omega^k(\Theta') = G_\omega^k(\Theta)$, empleamos (4.2) para calcular

$$U'^{c*}_\omega = \frac{-1}{\beta_2}\ln\left[\exp(-\gamma)\left(\sum_{t\in T_\omega}\exp-(\alpha_t^c + \beta_2 U'^{c*}_{\omega,t})\right)\right] = \frac{\gamma}{\beta_2} + U_\omega^{c*}$$

entonces

$$\alpha'^c + \beta_1 U'^{c*}_\omega = \alpha^c - \frac{\gamma\beta_1}{\beta_2} + \beta_1\left(\frac{\gamma}{\beta_2} + U_\omega^{c*}\right) = \alpha^c + \beta_1 U_\omega^{c*}$$

□

La tercera fuente de sobrespecificación aparece en la estructura de los costes generalizados del modelo que se basan en los coeficiente θ_a y θ_b.

PROPOSICIÓN 4.2.3 *Sean $\theta_a,\ \theta_b,\ \beta_1,\ \beta_2$ y dado un $\gamma \neq 0$, si reemplazamos estos valores por $\theta'_a = \gamma\theta_a,\ \theta'_b = \gamma\theta_b,\ \beta'_1 = \frac{1}{\gamma}\beta_1,\ \beta'_2 = \frac{1}{\gamma}\beta_2$ entonces las funciones logit (4.1) y (4.3) no varían sus valores.*

DEMOSTRACIÓN. Emplearemos la notación Θ y Θ' como en las proposiciones anteriores 4.2.1. Tenemos

$$\alpha^k + \beta'_1 U'^{k*}_\omega = \alpha^k + \frac{1}{\gamma}\beta_1\gamma\theta_k C_\omega^{k*} = \alpha^k + \beta_1 U_\omega^{k*} \quad k \in \{a, b\}$$

$$U'^{c*}_\omega = \frac{-1}{\beta'_2}\ln\left(\sum_{t\in T_\omega}\exp-(\alpha_t^c + \beta'_2 U'^{c*}_{\omega,t})\right) = \gamma\frac{-1}{\beta_2}\ln\left(\sum_{t\in T_\omega}\exp-(\alpha_t^c + \frac{1}{\gamma}\beta_2\{\gamma\theta_a C_{\omega,t}^{a*} + \gamma\theta_b C_{tj}^{b*}\}\right) = \gamma U_\omega^{c*},$$

entonces,

$$\alpha^c + \beta'_1 U'^{c*}_\omega = \alpha^c + \frac{1}{\gamma}\beta_1\gamma U_\omega^{c*} = \alpha^c + \beta_1 U_\omega^{c*}$$

Esto muestra que $G_\omega^k(\Theta') = G_\omega^k(\Theta)$. Ahora probaremos que $G_{\omega,t}^c(\Theta') = G_{\omega,t}^c(\Theta)$.

$$\alpha_t^c + \beta'_2 U'^{c*}_{\omega,t} = \alpha_t^c + \frac{\beta_2}{\gamma}\left(\gamma\theta_a C_{\omega,t}^{a*} + \gamma\theta_b C_{tj}^{b*}\right) = \alpha_t^c + \beta_2 U_{\omega,t}^{c*}$$

□

La razón para eliminar la sobrespecificación está motivada en el comportamiento computacional de los algoritmos de resolución. Cuando se elimina la sobrespecificación, la bondad del ajuste no varía, pero la rapidez de convergencia de los métodos se incrementa. De hecho este efecto fue detectado numéricamente cuando se comprobó que el algoritmo de Hooke-Jeves no convergía en la calibración del modelo y posteriormente se realizó el análisis teórico descrito en las proposiciones 4.2.1-4.2.3. El conjunto de soluciones óptimas al problema de calibración es no acotado y el procedimiento convergía a soluciones no acotadas, que producían errores numéricos cuando se intentaba calcular la exponencial de valores demasiado grandes.

El método de Daganzo y Kusnic [62] o la adaptación del método de Bierlaire y otros [26] pueden ser empleados para evitar la sobrespecificación. Asumiremos la eliminación de la sobrespecificación y lo expresaremos diciendo que el vector de parámetros pertenece a un conjunto C

$$\Theta \in C \subset \mathbb{R}^n\,. \tag{4.5}$$

Asumiremos que este conjunto C puede contener alguna nueva restricción que refleje alguna información adicional sobre el vector de parámetros, por ejemplo, la no negatividad, etc.

Las consideraciones que hemos realizado sobre el problema de sobrespecificación serán tenidas en cuenta en la próxima sección, donde nos plantearemos calibrar el modelo bajo la hipótesis de que la matriz O-D también es desconocida. Es decir, supondremos que el vector de parámetros debe verificar la restricción (4.5).

4.2.3 Un algoritmo heurístico para el problema de calibración

En este apartado proponemos un método heurístico para la resolución del problema de calibración basado en los algoritmos iterativos de optimización-asignación. Primeramente el algoritmo resuelve un problema TAP-M para un valor dado del vector de parámetros que proporciona los costes de transporte en equilibrio. En la fase siguiente y considerando estos valores, se plantea un problema clásico de estimación de los parámetros. Este proceso iterativo se continúa hasta que se alcance el criterio de convergencia. El esquema de este algoritmo se muestra en la tabla 4.1.

Tabla 4.1: Algoritmo heurístico para la calibración del TAP-M

0. *(Inicialización).* Determinar un valor inicial para el vector de parámetros Θ^0. Tomar $t = 0$.
1. *(Problema del nivel inferior).* Resolver el problema de equilibrio con modos combinados para el vector Θ^t.
2. *(Problema del nivel superior).* Determinar Θ^{t+1} por medio de un problema estadístico de estimación y usando los costes en equilibrio obtenidos en la etapa anterior.
3. *(Criterio de convergencia).* Si se alcanza el criterio de convergencia entonces parar. En caso contrario tomar $t = t + 1$ e ir al paso 1.

Este procedimiento requiere de un vector inicial de parámetros. La eficiencia de este método puede estar fuertemente influenciada por la proximidad a la solución óptima del vector inicial. Quizás una buena elección sería tomar los parámetros α^k, α_t^c proporcionales a la partición modal de la demanda observada. El resto de valores se pueden elegir iguales a 1.

El método de Abrahamsson y Lundqvist [5] requiere sobre diez iteraciones principales para obtener tres cifras significativas en la estimación de los parámetros de un modelo combinado aplicado a la región de Estocolmo. El algoritmo empleado es similar al aquí descrito, lo que indica que sería recomendable la utilización de la clase de algoritmos CG/SD en la paso 1 (ver capítulos 2 y 3) en la resolución de los modelos de equilibrio, aunque el procedimiento también sería viable para los clásicos algoritmos de Evans o de la descomposición simplicial restringida.

En el paso 2, se requiere analizar dos cuestiones fundamentales: la elección del método de resolución y la métrica para la función objetivo del nivel superior.

Un primer método computacional, para resolver el problema de optimización, se obtiene de la resolución de las condiciones de optimalidad del problema, que se formula mediante un sistema de ecuaciones no lineales que se obtiene igualando a cero las derivadas parciales respecto a los parámetros de la función métrica y puede ser resuelto, por ejemplo, con el paquete GAMS (ver Castillo y otros [41]).

Una técnica específica para la estimación de los parámetros de un modelo logit anidado (NL) es la denominada *estimación paso a paso* de Daly y Zachary [63] y Sobel [217]). La técnica consiste en estimar los parámetros para un modelo logit en cada nivel y emplear estos parámetros en el nivel superior. En nuestro caso obtendríamos en el nivel de elección del nodo de transferencia los parámetros α_t^c y β_2. Con estas estimaciones determinaríamos la utilidad compuesta del modo c ("log-suma" de las utilidades) que sería aplicado en el nivel de elección de modo. Existen procedimientos complicados para corregir las estimaciones con vista a reducir los errores estándares de las estimaciones, pero lo más común, en la práctica, es considerar que éstos son pequeños como lo hacen Ben-Akiva y Lerman

[14]. La estimación en cada nivel se puede obtener mediante el procedimiento de máxima verosimilitud (ver McFadden [169]) o cualquier otro procedimiento de estimación.

El tercer método consiste en resolver directamente el problema de optimización y es el que hemos seguido. Hemos empleado el método de Hooke-Jeeves (o método de Powell) debido a que este procedimiento usa solamente evaluaciones funcionales durante el proceso de optimización y evita por tanto la utilización del gradiente cuya expresión analítica es tediosa.

El cambio relativo entre dos iteraciones sucesivas Θ^{t+1} y Θ^t se puede emplear como criterio de parada en el paso 4.

4.2.4 Algunos resultados computacionales para la fase de estimación

En este apartado se realiza una pequeña experiencia computacional que intenta contestar la siguiente cuestión: ¿Qué métrica es recomendable elegir en la fase de estimación (paso 2 del algoritmo descrito en la tabla 4.1)?.

En el experimento se calculan los costes de equilibrio en una red multimodal, mediante la resolución del TAP-M para un valor conocido del vector de parámetros. Empleando estos costes y la partición modal que genera (que hace el papel de la encuesta de movilidad en las aplicaciones reales), se intenta recuperar los parámetros verdaderos (que son conocidos) mediante un problema de estimación. Se han empleado las métricas ML, NLLS, WNLLS y ME. Hemos considerado la red GaM (ver capítulo 1 sección 1.5). Esta red tiene 44 arcos, 13 nodos, 3 centroides y 4 pares de demanda. Las alternativas consideradas son: *coche privado, park'n ride* y *metro*. Se han distinguido en los viajes combinados el tiempo empleado en la primera parte del viaje, realizada en coche del centroide al intercambiador, y la segunda parte, realizada en metro del intercambiador al destino.

La alternativa *metro* tiene un primer coste asociado con el tiempo empleado andando, desde el origen a la parada del metro y un segundo coste empleado en la red de metro-pedestre para completar el viaje. En la experiencia computacional hemos añadido un nuevo parámetro, $\theta_{\tilde{b}}$, para homogeneizar el tiempo empleado andando con el tiempo empleado en los vehículos del metro. El coste generalizado para la alternativa *metro* es calculado por $U_\omega^{b*} = \theta_{\tilde{b}} C_\omega^{*\tilde{b}} + \theta_b C_\omega^{*b}$, donde $C_\omega^{\tilde{b}}$ es el tiempo andando y C_ω^b es el tiempo en vehículos de metro para el par ω.

La base de datos empleada está mostrada en la tabla 4.2. La columna cuarta es el número de usuarios en cada alternativa. Las cantidades obtenidas del modelo TAP-M han sido redondeadas para tomar valores enteros tal y como ocurre en los datos derivados de la realización de una encuesta. La columna quinta es el tiempo empleado en la primera parte del viaje, es decir, el tiempo empleado en coche para la alternativa *park'n ride* y el tiempo andando hasta la parada para la alternativa *metro*. La columna sexta muestra el tiempo empleado para ir de la parada de metro hasta el destino. La herramienta computacional empleada ha sido *Numerical Recipes Software* de Press y otros [203]. Esta librería de subrutinas contiene el algoritmo de Hooke-Jeeves. El método para realizar las búsquedas lineales se basa en ajustes cuadráticos. Este algoritmo requiere un punto inicial para comenzar a generar la sucesión de iteraciones. En el experimento hemos considerado cinco puntos iniciales: $P_1, \ldots, P_5$ que son mostrados en la tabla 4.3.

La primera experiencia computacional realizada con las cuatro métricas tuvo errores computacionales. Si la sobrespecificación de los parámetros no es eliminada, la cara óptima no es acotada y el método va encontrando grandes valores del vector de parámetros que nos llevan a calcular la exponencial de valores demasiado grandes, produciendo los mencionados errores computacionales. Eliminamos la sobrespecificación mediante el método de Daganzo y Kusnic que fija un valor de los parámetros involucrados en cada una de las tres fuentes de sobrespecificación caracterizada anteriormente. Hemos fijado estos valores al valor verdadero del parámetro. El vector verdadero de los parámetros se denota por P^* y se muestra en la tabla 4.3.

Los resultados obtenidos se muestran en la tabla 4.4. La primera conclusión es que los verdaderos valores de los parámetros no han sido recuperados (observar el parámetro $\theta_{\tilde{b}}$ y α^b). Esto es debido

Tabla 4.2: Base de datos para la fase de estimación

Modo	Par Demanda	Intercambiador	Usuarios observados	Coste viaje (1ª Parte)	Coste viaje (2ª Parte)
	1	–	3023	98.21	–
Coche privado	2	–	3070	118.96	–
C_ω^{a*}	3	–	3230	119.77	–
	4	–	2240	70.95	–
	1	12	260	7.88	88.70
	1	13	25	130.39	69.60
	2	12	175	7.88	118.65
Park'n ride	2	13	63	130.39	46.80
C_{it}^{a*}, C_{tj}^{b*}	3	12	75	127.65	47.30
	3	13	457	11.43	119.90
	4	12	8	127.65	88.70
	4	13	274	11.43	69.60
	1	–	1195	24.80	97.70
Metro	2	–	193	24.80	136.85
$C_\omega^{\tilde{b}*}$, C_ω^{b*}	3	–	241	24.80	136.10
	4	–	480	24.80	76.60

Tabla 4.3: Valores iniciales y óptimo de los parámetros

Punto	θ_a	θ_b	$\theta_{\tilde{b}}$	β_1	β_2	α^a	α^b	α^c	α_{12}^c	α_{13}^c
	Fijado					Fijado			Fijado	
P^*	1.0	1.0	1.0	0.10	0.030	1.0	0.5	0.5	1.0	0.5
P_1	1.0	1.1	1.0	0.09	0.021	1.0	1.0	-0.6	1.0	0.4
P_2	1.0	-1.0	2.0	0.09	0.021	1.0	1.0	1.6	1.0	0.4
P_3	1.0	-2.0	2.0	0.09	0.021	1.0	1.0	1.6	1.0	0.4
P_4	1.0	-2.0	2.0	0.09	0.021	1.0	1.0	-1.6	1.0	0.4
P_5	1.0	2.0	2.0	0.09	0.021	1.0	-1.0	-1.6	1.0	0.4

a que existen infinitas soluciones óptimas del problema de calibración y se ha obtenido una de ellas, distinta al verdadero valor de los parámetros.

Una vez más aparece un problema de sobrespecificación debido a que los tiempos empleados para llegar andando al intercambiador son iguales para todos los pares de demanda. Más formalmente, dado un valor del vector de parámetros Θ es suficiente probar que las utilidades respecto al nuevo vector Θ' no varían. Sea Θ un vector de parámetros dado y como en el ejemplo $C_\omega^{\tilde{b}} = 24.8 \quad \forall \omega \in \{1,2,3,4\}$, podemos definir

$$\begin{array}{rcl} \theta'_{\tilde{b}} & = & \theta_{\tilde{b}} + \gamma \\ \alpha'^b & = & \alpha^b - \beta_1 \gamma C_\omega^{\tilde{b}} \end{array} \tag{4.6}$$

donde $\gamma \neq 0$. El resto de las componentes del vector Θ' coinciden con Θ. Notar que el valor de α'^b es independiente de $\omega \in W$. Calcularemos la utilidad de la alternativa b con respecto al valor del parámetro Θ'.

$$\alpha'^b \quad + \quad \beta_1 U'^b_\omega = \alpha^b - \beta_1 \gamma C_\omega^{\tilde{b}} + \beta_1 \left((\theta_{\tilde{b}} + \gamma) C_\omega^{\tilde{b}} + \theta_b C_\omega^b \right) = \alpha^b + \beta_1 U_\omega^b$$

El resto de utilidades no depende de los parámetros α^b y θ^b obteniendo el resultado deseado.

Empleando (4.6) podemos calcular otras estimaciones con el mismo valor de la función objetivo en el problema de estimación, basta con dar valores al parámetro γ. Por ejemplo, para la maximización de la función de verosimilitud las estimaciones medias (para los cinco puntos iniciales) de los parámetros $\theta_{\tilde{b}}$ y α^b son 0.1298 y 1.5738 respectivamente. Para el valor de $\gamma = 0.1298$ la relación (4.6) da los valores

$\theta'_{\tilde{b}} = 1.$ y $\alpha'^{b} = 0.4979$ que constituyen buenas aproximaciones al verdadero valor de los parámetros. Este es un ejemplo de cómo el método de estimación reproduce los parámetros originales.

La sobrespecificación produce problemas mal condicionados y este ejemplo puede ser un buen test de prueba para evaluar la complejidad computacional de las distintas métricas. Los resultados muestran que para el punto P_2 únicamente el método de estimación máximo verosímil (ML) converge. Además, este método requiere, en general, menor número de evaluaciones de la función objetivo. Por otro lado la estimación ML es preferible basándose en propiedades teóricas.

Tabla 4.4: Fase de estimación de los parámetros del TAP-M. Resultados

Método		Iter.	N[a]	Z^*	θ_b	$\theta_{\tilde{b}}$	β_1	β_2	α^c	α^b	α^c_{13}
	P^*				1.000	1.000	0.100	0.030	0.500	0.500	0.500
	P_1	18	3493	10304.52	1.010	0.133	0.096	0.029	0.426	1.564	0.506
	P_2	18	3114	10304.52	1.010	0.117	0.096	0.029	0.417	1.608	0.506
ML	P_3	21	3781	10304.52	1.010	0.128	0.096	0.029	0.412	1.580	0.506
	P_4	31	5574	10304.54	1.010	0.103	0.096	0.029	0.417	1.645	0.503
	P_5	29	5457	10304.56	1.012	0.168	0.095	0.029	0.412	1.472	0.504
	P_1	45	8026	149.71	1.001	-0.017	0.099	0.030	0.519	2.007	0.492
	P_2			No converge							
NLLS	P_3	19	3258	251.84	1.006	0.140	0.099	0.030	0.508	1.572	0.489
	P_4	41	6893	150.70	1.001	-0.006	0.099	0.030	0.523	1.976	0.492
	P_5	60	10208	149.70	1.001	-0.014	0.099	0.030	0.521	2.000	0.492
	P_1	35	6128	4.97	1.007	0.069	0.097	0.030	0.457	1.749	0.502
	P_2			No converge							
WNLLS	P_3	17	3108	5.026	1.008	0.144	0.097	0.030	0.457	1.742	0.501
	P_4	45	8007	4.98	1.007	0.071	0.097	0.030	0.457	1.742	0.501
	P_5	37	6485	4.98	1.007	0.069	0.097	0.030	0.458	1.746	0.501
	P_1	24	4209	3.02	1.008	0.098	0.097	0.029	0.442	1.663	0.504
	P_2			No converge							
ME	P_3	16	2913	3.019	1.009	0.116	0.097	0.029	0.445	1.686	0.504
	P_4	39	7004	3.02	1.008	0.090	0.097	0.029	0.445	1.686	0.503
	P_5	34	5918	3.02	1.008	0.091	0.097	0.029	0.449	1.685	0.503

[a]Número de evaluaciones de la función objetivo.

4.3 Un modelo binivel para la estimación de matrices O-D y calibración de los parámetros

El problema de ajuste de matrices de demanda de viajes O-D en redes congestionadas, ha sido formulado mediante la programación matemática binivel como se indicón en la introducción de este capítulo. El nivel superior minimiza la suma de la discrepancia entre los flujos observados y los obtenidos como solución del modelo de equilibrio, más la diferencia entre la matriz ajustada y la de referencia. En el nivel inferior el modelo de equilibrio asigna la matriz de demanda O-D a la red de transporte, obteniéndose el correspondiente vector de flujo en los arcos.

En esta sección, se extiende el modelo binivel anterior para poder estimar simultáneamente la matriz O-D $\bar{\mathbf{g}}$ y el vector de parámetros del modelo Θ del TAP-M. El nivel superior decide conjuntamente la matriz O-D y el vector de parámetros, y en el nivel inferior se produce el flujo y la partición modal en equilibrio para el par $(\bar{\mathbf{g}}, \Theta)$ obtenidos en el nivel superior.

Los datos básicos del modelo son:

◇ Una matriz O-D para la demanda total en todas las alternativas consideradas. Esta matriz

puede ser una matriz desactualizada, una matriz de referencia obtenida por otro procedimiento, o simplemente no se dispone ninguna información sobre la misma.

- ◇ Una partición modal de referencia. Esta es la información externa al modelo TAP-M, que emplea el modelo binivel de la sección anterior para efectuar la calibración. Esta información se puede obtener mediante encuestas domiciliarias o de estudios realizados en otras áreas urbanas.
- ◇ Un conjunto de flujos en equilibrio en la red multimodal. Este constará de medidas de aforo en la red de tráfico y número de usuarios en los arcos de la red de transporte público. La obtención de esta información en cada red requiere de métodos distintos.

El modelo de calibración y ajuste de la demanda se formula

$$\begin{array}{ll} \text{minimizar} & F(\bar{\mathbf{g}},\Theta) = \mu_1 F_1(\bar{\mathbf{g}},\hat{\mathbf{g}}) + \mu_2 F_2(\mathbf{f}(\bar{\mathbf{g}},\Theta),\hat{\mathbf{f}}) + \mu_3 F_3(\mathbf{g}(\bar{\mathbf{g}},\Theta),\tilde{\mathbf{g}}) \\ \text{sujeto a} & 0 \le \bar{\mathbf{g}} \le \mathbf{b} \\ & \Theta \in C \subset \mathbb{R}^{\ell} \\ & (\mathbf{f},\mathbf{g}) = \text{arg minimizar } S(\mathbf{y},\Theta) + R(\mathbf{q},\Theta) \\ & \qquad\qquad \text{sujeto a} \\ & \qquad\qquad (\mathbf{y},\mathbf{q}) \in \tilde{\Omega}(\bar{\mathbf{g}}) \end{array} \qquad \text{[CDAM]}$$

donde la función F_1 puede ser cualquier métrica y proporciona la distancia entre la demanda total estimada $\bar{\mathbf{g}}$ y la de referencia $\hat{\mathbf{g}}$; F_2 es otra métrica que mide la discrepancia entre el flujo observado $\hat{\mathbf{f}}$ y la solución de equilibrio del TAP-M; F_3 es otra métrica que mide la distancia entre la partición modal estimada por el TAP-M, $\mathbf{g}$, y la observada $\tilde{\mathbf{g}}$; el vector $\mathbf{b}$ es una cota superior para la demanda y suponemos que el vector de parámetros Θ pertenece al conjunto C que cumple las siguientes condiciones:

(i) Es un conjunto compacto.

(ii) Añadimos una nueva restricción en los parámetros para asegurar que el problema TAP-M sea convexo y su solución defina la situación de equilibrio. Esta condición se garantiza, ver el apéndice III del capítulo 1, cuando

$$0 < \beta_1 \le \beta_2.$$

Si $\beta_1 = \beta_2$ el modelo logit anidado se transforma en un modelo multinomial. Cada alternativa de viaje combinado se transforma en una nueva alternativa de elección de modo cuyo parámetro asociado es $\alpha^c + \alpha^c_t$. Varios estudios muestran que el comportamiento de los usuarios cuando eligen modo de transporte se ve inversamente influenciado por la percepción de los costes generalizados. Esto hace que en las aplicaciones los valores $\beta_1 \le 0$ y $\beta_2 \le 0$ no sean posibles. Supondremos que existe $\varepsilon > 0$ cumpliendo $0 < \varepsilon \le \beta_1 \le \beta_2$, y

$$C \subset \{\Theta \ / \ \ 0 < \varepsilon \le \beta_1 \le \beta_2\} \tag{4.7}$$

(iii) La sobrespecificación de los parámetros se ha eliminado. La discusión planteada en la sección anterior establece que se deben fijar tres valores de los parámetros tal y como indican las proposiciones 4.2.1, 4.2.2 y 4.2.3

La condición (i) es necesaria para probar la existencia de soluciones para el CDAM; la condición (ii) garantiza que la solución del TAP-M caracteriza la situación de equilibrio, la condición (iii) no restringe la calidad del ajuste de los parámetros pero mejora el comportamiento computacional de los algoritmos.

El CDAM puede ser interpretado en el marco de la programación matemática multiobjetivo, donde los coeficientes μ_1, μ_2 y μ_3 son los correspondientes pesos que expresan la importancia relativa de las

tres medidas, que pueden ser relacionados con la fiabilidad de las observaciones. Por ejemplo, la ponderación de una matriz antigua de demanda O-D desactualizada $\bar{\mathbf{g}}$, puede tener menor importancia que unas mediciones de flujos $\hat{\mathbf{f}}$ actuales. Notar que, para poder transferir la fiabilidad de las observaciones en los pesos de las métricas, es necesario que el rango de las métricas F_i sean similares. Si esto no fuese posible se puede considerar una muestra aleatoria sobre un entorno que contuviese a priori la solución de CDAM, $\{(\bar{g}^j, \Theta^j)\}_{j\in J}$. Después, calcularíamos la desviación estándar de $\{F_i(\bar{g}^j, \Theta^j)\}_{j\in J}$, denotado por s_i, y reemplazaríamos la funciones métricas $\tilde{F}_i$ por $\frac{1}{s_i}F_i$. Con esta transformación la varianza muestral de las tres transformaciones es 1 y podrían ser comparables.

Se ha realizado un considerable esfuerzo de investigación en los aspectos teóricos de la calibración de los modelos de transporte urbano. Esto es debido a que la bondad de estos modelos depende, aunque no exclusivamente, de la exactitud en la estimación de estos parámetros. Si la matriz de demanda O-D fuese conocida el CDAM podría ser empleado para calibrar el TAP-M. Esta nueva formulación conduce a un marco para la calibración de modelos combinados en los que, además de la información derivada de las encuestas, también se emplea información de los aforos de la red.

Muchos métodos de calibración se basan en la estimación máximo verosímil de los parámetros. Estos métodos conducen a elegir los parámetros de tal modo que reproduzcan el tiempo medio de viaje observado en la red. Esta metodología ha sido empleada en una amplia gama de modelos. Un ejemplo que podría ser visto como un predecesor del TAP-M es dado por Florian y Los [83] que consideraron el problema de determinar matrices O-D de la primera componente de viajes combinados tipo *park'n ride*, sugirieron que este modelo podría ser calibrado como una generalización de los métodos basados en reproducir el tiempo medio observado en la red. Este método también puede ser situado como caso particular del CDAM. Otro ejemplo, donde el problema de calibración está explícitamente definido mediante un modelo binivel, está dado por Ottomanelli [190] y Abrahamsson y Lundqvist [5]. Ambos métodos caen dentro del CDAM. El primero considera $\mu_1 = \mu_3 = 0$ y basa la calibración sobre observaciones de flujo. Alternativamente, el segundo basa la calibración en encuesta de movilidad y toma $\mu_1 = \mu_2 = 0$, y F_3 es la función de verosimilitud.

El CDAM tiene la ventaja de usar toda la información disponible. No es obligatorio disponer de una partición modal de referencia porque ésta está contenida implícitamente en el nivel de servicio de los arcos, información que es más económica de obtener. En caso contrario, si esta información estuviera disponible se podría definir la métrica F_3 (empleando por ejemplo el principio de máxima verosimilitud) y se podría calibrar los parámetros teniendo en cuenta las interacciones existentes entre elección de ruta, modo e intercambiador. Por otro lado, si los parámetros del modelo fuesen conocidos, el CDAM se convertiría en un problema de ajustar matrices O-D pero sobre una red multimodal.

4.3.1 Existencia de soluciones para el CDAM

Esta sección está dedicada a mostrar que el CDAM tiene, al menos, una solución independientemente del tipo de datos, bajo las suposiciones de continuidad de las métricas F_1, F_2 y F_3; y de las funciones de coste en los arcos. Esto implica que el modelo es aplicable también a situaciones en los que solamente se dispone de medidas de flujo sobre un subconjunto de arcos en la red multimodal, o solamente unas cuantas entradas de la matriz de referencia O-D son conocidas, o incluso, cuando no se dispone de una partición modal de referencia. En estos casos, las métricas estarán definidas sobre el conjunto de datos observados. El CDAM puede ser aplicado, no solamente cuando los datos sean incompletos, sino cuando éstos sean incorrectos, por ejemplo cuando los flujos son inconsistentes, esto es, cuando no existe ninguna matriz O-D cuya asignación en equilibrio reproduzca las observaciones de flujo.

Emplearemos un argumento similar al usado en el trabajo de Chen y Florian [48] para probar la existencia de soluciones del DAM.

DEFINICIÓN 4.3.1 Sean $\Omega_h(.)$ y $\tilde{\Omega}(.)$ las aplicaciones punto-conjunto de las restricciones expresadas en función del flujo en los caminos y en los arcos respectivamente. Éstas están definidos por

$$\Omega_h(\bar{\mathbf{g}}) = \{\mathbf{h} \;/\; \delta^{\bar{\mathbf{g}}}\mathbf{h} = \bar{\mathbf{g}},\ \mathbf{h} \geq \mathbf{0}\}, \quad \text{con} \quad \mathbf{0} \leq \bar{\mathbf{g}} \leq \mathbf{b},$$
$$\tilde{\Omega}(\bar{\mathbf{g}}) = \{(\mathbf{y}, \mathbf{q}) \;/\; \delta^{\bar{\mathbf{g}}}\mathbf{h} = \bar{\mathbf{g}},\ \delta^{\mathbf{f}}\mathbf{h} - \mathbf{y} = \mathbf{0},\ \delta^{\mathbf{g}}\mathbf{h} - \mathbf{q} = \mathbf{0},\ \mathbf{h} \geq \mathbf{0}\}, \quad \text{con} \quad \mathbf{0} \leq \bar{\mathbf{g}} \leq \mathbf{b}$$

donde $\delta^{\bar{\mathbf{g}}}$ es la matriz de incidencia caminos-pares de demanda, $\delta^{\mathbf{f}}$ la matriz de incidencia ruta-arco y $\mathbf{b}$ es una cota superior para la matriz de demanda. □

DEFINICIÓN 4.3.2 Sean $\Omega_h^*(.)$ y $\tilde{\Omega}^*(.)$ las aplicaciones punto-conjunto de soluciones del TAP-M en función de flujo en los caminos y en los arcos respectivamente, definidas por

$$\begin{aligned}\Omega_h^*(\bar{\mathbf{g}},\Theta) &= \{\mathbf{h} \ / \ \mathbf{h}\in\Omega_h(\bar{\mathbf{g}}),\ \mathbf{h} \text{ es un flujo en equilibrio para el TAP-M para los valores } \bar{\mathbf{g}} \text{ y } \Theta\}\\ \tilde{\Omega}^*(\bar{\mathbf{g}},\Theta) &= \left\{(\mathbf{f},\mathbf{g}) \ / \ \mathbf{h}\in\Omega_h^*(\bar{\mathbf{g}},\Theta),\ \delta^{\bar{\mathbf{g}}}\mathbf{h}=\bar{\mathbf{g}},\ \delta^{\mathbf{f}}\mathbf{h}-\mathbf{f}=\mathbf{0},\ \delta^{\mathbf{g}}\mathbf{h}-\mathbf{g}=\mathbf{0},\ \mathbf{h}\geq\mathbf{0}\right\}\end{aligned}$$

con $(\bar{\mathbf{g}},\Theta)\in[\mathbf{0},\mathbf{b}]\times\mathbf{C}$ □

DEFINICIÓN 4.3.3 Definimos los siguientes conjuntos:

$$\begin{aligned}\Omega_h([\mathbf{0},\mathbf{b}]) &= \bigcup_{\bar{\mathbf{g}}\in[\mathbf{0},\mathbf{b}]}\Omega_h(\bar{\mathbf{g}})\\ \Omega_h^*([\mathbf{0},\mathbf{b}]\times C) &= \bigcup_{(\bar{\mathbf{g}},\Theta)\in[\mathbf{0},\mathbf{b}]\times C}\Omega_h^*(\bar{\mathbf{g}},\Theta)\\ \tilde{\Omega}^*([\mathbf{0},\mathbf{b}]\times C) &= \bigcup_{(\bar{\mathbf{g}},\Theta)\in[\mathbf{0},\mathbf{b}]\times C}\tilde{\Omega}^*(\bar{\mathbf{g}},\Theta)\end{aligned}$$

□

LEMA 4.3.4 $\Omega_h([\mathbf{0},\mathbf{b}])$ *es un conjunto compacto no vacío*

DEMOSTRACIÓN. Es inmediata. □

El siguiente lema establece la continuidad de la aplicación de restricciones $\tilde{\Omega}(.)$.

LEMA 4.3.5 *La función multievaluada* $\tilde{\Omega}(.)$ *es una aplicación continua en su dominio* $[\mathbf{0},\mathbf{b}]$

DEMOSTRACIÓN. Demostraremos que $\Omega_h(.)$ es continua en $\bar{\mathbf{g}}\geq\mathbf{0}$ usando la nota 6.3.1 de Shimizu y otros [212].

Sea $H(\bar{\mathbf{g}},\mathbf{h})=\delta^{\bar{\mathbf{g}}}\mathbf{h}-\bar{\mathbf{g}}$ y $G(\bar{\mathbf{g}},\mathbf{h})=-\mathbf{I}\mathbf{h}$, donde $\mathbf{I}$ es la matriz identidad, estas funciones cumplen:

(i) H y G son funciones lineales y por tanto continuas en $\{\bar{\mathbf{g}}\}\times\mathbb{R}^m$.

(ii) H es lineal y por tanto continuamente diferenciable respecto a $\mathbf{h}$ en $N(\bar{\mathbf{g}})\times\mathbb{R}^m$, donde $N(\bar{\mathbf{g}})$ es un entorno de $\bar{\mathbf{g}}$ y el rango $(\nabla_h H(\bar{\mathbf{g}},\mathbf{h}))=$ rango $(\delta^{\bar{\mathbf{g}}})=|W|$ para todo $(\bar{\mathbf{g}},\mathbf{h})$.

(iii) Se cumple que $\Omega_h(\bar{\mathbf{g}})\neq\emptyset$ y

$$\text{Cl}\ \left(\{\mathbf{h} \ / \ \mathbf{h}>\mathbf{0},\ \delta^{\bar{\mathbf{g}}}\mathbf{h}=\bar{\mathbf{g}}\}\right)=\Omega_h(\bar{\mathbf{g}})$$

donde Cl es la clausura de un conjunto.

Bajo estas hipótesis la aplicación $\Omega_h(.)$ es continua en $[\mathbf{0},\mathbf{b}]$.

La aplicación multievaluada $\Omega(.)$ puede expresarse

$$\tilde{\Omega}(.)=\left[(\delta^{\mathbf{f}},\delta^{\mathbf{g}})\ \text{o}\ \Omega_h\right](.),$$

donde $\tilde{\Omega}(.)$ es la composición de una función multievaluada continua con una aplicación continua y el resultado que se obtiene es también continuo. □

TEOREMA 4.3.6 *Si todas las funciones de coste $c_l(.)$ son continuas, entonces la aplicación punto-conjunto de soluciones del TAP-M, $\Omega_h^*(.)$, es semicontinua superiormente en $(\bar{\mathbf{g}}, \Theta) \in [\mathbf{0}, \mathbf{b}] \times C$.*

DEMOSTRACIÓN. Como la función objetivo del TAP-M es continua y $\tilde{\Omega}(\bar{\mathbf{g}})$ es un conjunto compacto para cualquier $\bar{\mathbf{g}} \in [\mathbf{0}, \mathbf{b}]$ y $\Theta \in C$, entonces $\tilde{\Omega}^*(\bar{\mathbf{g}}, \Theta)$ es no vacío.

Considerar cualquier $(\tilde{\mathbf{g}}, \tilde{\Theta}) \in [\mathbf{0}, \mathbf{b}] \times C$, sucesión convergente $\{(\bar{\mathbf{g}}^j, \Theta^j)\}$, tal que $(\bar{\mathbf{g}}^j, \Theta^j) \to (\tilde{\mathbf{g}}, \tilde{\Theta})$ cuando $j \to \infty$. Considérese una sucesión $\{\mathbf{h}^j\}$ tal que $\mathbf{h}^j \in \Omega_h^*(\bar{\mathbf{g}}^j, \Theta^j)$ y $\mathbf{h}^j \to \tilde{\mathbf{h}}$. Probaremos que $\tilde{\mathbf{h}} \in \Omega_h^*(\tilde{\mathbf{g}}, \tilde{\Theta})$.

Como $\mathbf{h}^j \in \Omega_h^*(\bar{\mathbf{g}}^j, \Theta^j) \subset \Omega_h(\bar{\mathbf{g}}^j)$ y $\Omega_h(.)$ es continuo, como ya se vió en la demostración del lema 4.3.5, $\tilde{\mathbf{h}} \in \Omega_h(\tilde{\mathbf{g}})$.

Si existe un $\omega \in W$ cumpliendo $\tilde{g}_\omega = 0$, entonces $\lim_j \bar{g}_\omega^j = 0$. Como $0 \le h_p^j \le \bar{g}_\omega^j$ para todo $p \in P_\omega$, entonces $\lim_j h_p^j = \tilde{h}_p = 0$ que es el único valor posible de flujo que satisface la demanda ω. Sin pérdida de generalidad asumiremos que $\tilde{g}_\omega > 0,\ \forall \omega \in W$.

Ahora probaremos que $\tilde{\mathbf{h}}$ es un flujo en equilibrio mediante el teorema 1.2.1. Como el número de caminos es finito entonces existe un índice j_0 cumpliendo que para todo $j > j_0$

$$\text{si } \tilde{h}_p > 0 \text{ entonces } h_p^j > 0,$$

y empleando las condiciones de optimalidad para el TAP-M, teorema 1.2.1, existen unos coeficientes ${\lambda_\omega^*}^i$ cumpliendo

$$ {\lambda_\omega^*}^i = \begin{cases} \bar{C}_p(\mathbf{h}^j, \Theta^j) + \frac{\ln g_\omega^k(\mathbf{h}^j) + (\alpha^k)^j}{(\beta_1)^j}, & \text{si } p \in P_\omega^k,\ k \in \{a, b\}; \\ \bar{C}_p(\mathbf{h}^j, \Theta^j) + \frac{\ln g_\omega^c(\mathbf{h}^j) + (\alpha^k)^j}{(\beta_1)^j} + \frac{-\ln g_\omega^c(\mathbf{h}^j) + \ln g_{\omega,t}^c(\mathbf{h}^j) + ({\alpha_t}^c)^j}{(\beta_2)^j}, & \text{si } p \in P_{\omega,t}^c. \end{cases} \tag{4.8}$$

Notar que las funciones $\bar{C}_p(.)$ son continuas por serlo $c_l(.)$ para todo $l \in A \cup B$ y que $g_\omega^a(.),\ g_\omega^b(.),\ g_{\omega,t}^c(.)$ son siempre continuas. Empleando la condición (4.7) obtenemos que se cumple $\lim_j (\beta_s)^j = \tilde{\beta}_s \ge \varepsilon > 0, \quad s = 1, 2.$

Demostraremos, por reducción al absurdo, que $g_{\omega,t}^c(\tilde{\mathbf{h}}) > 0$, para todo $t \in T_\omega$, y $g_\omega^k(\tilde{\mathbf{h}}) > 0,\ k \in \{a, b\}$ para todo $\omega \in W$. Si uno de ellos tuviera el valor cero, como

$$\begin{aligned} &\ln g_{\omega,t}^c(\mathbf{h}^j) \to \ln g_{\omega,t}^c(\tilde{\mathbf{h}}), \quad t \in T_\omega, \quad \omega \in W; \\ &\ln g_\omega^k(\mathbf{h}^j) \to \ln g_\omega^k(\tilde{\mathbf{h}}), \quad \omega \in W, \quad k \in \{a, b\}, \end{aligned}$$

existiría un ω cumpliendo ${\lambda_\omega^*}^i \to -\infty$. Como $\bar{C}_p(.)$ es continua en $\Omega_h([\mathbf{0}, \mathbf{b}]) \times C$, el valor mínimo está acotado, entonces la única posibilidad es que el mínimo se alcance en $-\infty$ y es

$$\begin{aligned} g_{\omega,t}^c(\mathbf{h}^j) &\to 0, \quad \omega \in W, \quad t \in T_\omega; \\ g_\omega^k(\mathbf{h}^j) &\to 0, \quad \omega \in W, \quad k \in \{a, b\}. \end{aligned}$$

Por continuidad de $g_{\omega,t}^c(\tilde{\mathbf{h}}) = 0$ y $g_\omega^k(\tilde{\mathbf{h}}) = 0,\ k \in \{a, b\}$; y esto contradice la suposición de que $\tilde{g}_\omega > 0$.

Tomando límites en j a ambos lados de la igualdad (4.8) obtenemos

$$ \tilde{\lambda}_\omega^* = \begin{cases} \bar{C}_p(\tilde{\mathbf{h}}, \tilde{\Theta}) + \frac{\ln g_\omega^k(\tilde{\mathbf{h}}) + \tilde{\alpha}^k}{\tilde{\beta}_1}, & \text{si } p \in P_\omega^k, \quad k \in \{a, b\}; \\ \bar{C}_p(\tilde{\mathbf{h}}, \tilde{\Theta}) + \frac{\ln g_\omega^c(\tilde{\mathbf{h}}) + \tilde{\alpha}^k}{\tilde{\beta}_1} + \frac{-\ln g_\omega^c(\tilde{\mathbf{h}}) + \ln g_{\omega,t}^c(\tilde{\mathbf{h}}) + \tilde{\alpha}_t^c}{\tilde{\beta}_2}, & \text{si } p \in P_{\omega,t}^c \end{cases} $$

En caso contrario, sea p tal que $\tilde{h}_p = 0$, entonces para todo $j \in \mathbb{N}$ se cumple

$$ {\lambda_\omega^j}^i \le \begin{cases} \bar{C}_p(\mathbf{h}^j, \Theta^j) + \frac{\ln g_\omega^k(\mathbf{h}^j) + (\alpha^k)^j}{(\beta_1)^j}, & \text{si } p \in P_\omega^k, \quad k \in \{a, b\}; \\ \bar{C}_p(\mathbf{h}^j, \Theta^j) + \frac{\ln g_\omega^c(\mathbf{h}^j) + (\alpha^k)^j}{(\beta_1)^j} + \frac{-\ln g_\omega^c(\mathbf{h}^j) + \ln g_{\omega,t}^c(\mathbf{h}^j) + ({\alpha_t}^c)^j}{(\beta_2)^j}, & \text{si } p \in P_{\omega,t}^c \end{cases} $$

y tomando límites en j a ambos lados de la anterior desigualdad, obtenemos que el límite satisface el teorema 1.2.1, probando que $\tilde{\mathbf{h}} \in \Omega_h^*(\tilde{\mathbf{g}}, \tilde{\Theta})$ □

LEMA 4.3.7 *$\Omega_h^*([\mathbf{0}, \mathbf{b}] \times C)$ es un conjunto compacto.*

DEMOSTRACIÓN. Por el teorema 4.3.6 y por la compacidad de $[\mathbf{0}, \mathbf{b}] \times C$ se garantiza que $\Omega_h^*([\mathbf{0}, \mathbf{b}] \times C)$ es un conjunto cerrado. Además $\Omega_h^*([\mathbf{0}, \mathbf{b}] \times C) \subset \Omega_h([\mathbf{0}, \mathbf{b}])$ es un conjunto acotado, entonces $\Omega_h^*([\mathbf{0}, \mathbf{b}] \times C)$ es un conjunto compacto. □

TEOREMA 4.3.8 *Si las funciones $c_l(.)$ son continuas en todos los arcos l; F_1, F_2 y F_3 son continuas en sus dominios , y C es un conjunto compacto entonces el CDAM tiene al menos una solución*

DEMOSTRACIÓN. Por el lema 4.3.7, $\Omega_h^*([\mathbf{0}, \mathbf{b}] \times C)$ es un conjunto compacto.

Sea

$$F : \Omega_h^*([\mathbf{0}, \mathbf{b}] \times C) \rightarrow \mathbb{R}^{|A \cup B|} \times \mathbb{R}^n$$
$$\mathbf{h} \rightarrow (\delta^{\mathbf{f}}\mathbf{h}, \delta^{\mathbf{g}}\mathbf{h}),$$

donde $|.|$ es el cardinal de un conjunto. F es una aplicación lineal y por esa razón es continua, entonces $F(\Omega_h^*([\mathbf{0}, \mathbf{b}] \times C))$ es un conjunto compacto. El conjunto de restricciones del CDAM puede ser reemplazado por $(\mathbf{f}, \mathbf{g}) \in F(\Omega_h^*([\mathbf{0}, \mathbf{b}] \times C))$ y la variable $\bar{\mathbf{g}}$ tiene una dependencia continua de $\mathbf{g}$, entonces el problema CDAM es la optimización de una función continua sobre un conjunto compacto y la demostración está garantizada por el teorema de Weierstrass. □

4.3.2 CDAM frente a la metodología secuencial: un ejemplo numérico

La finalidad de esta sección es motivar el uso del modelo CDAM, que aborda ambos problemas simultáneamente, frente a la metodología secuencial descrita anteriormente. La ventaja no proviene del tipo de datos empleados en ambas metodologías, ya que ambos procesos se pueden adaptar al mismo conjunto de datos, sino a los resultados de la estimación, que pueden ser significativamente diferentes. Para ilustrar esta afirmación considerar la red de prueba de la figura 4.1. Esta red de transporte tiene solamente un par de demanda y dos modos de transporte: en coche (a) o en metro (b) y solamente un camino por cada modo. Consideraremos el modelo de equilibrio que incluye la partición modal mediante un modelo logit. Este modelo está descrito en Marín [164] o el modelo P_2 de Fernández y otros [73].

Hemos asumido que los parámetros θ_a y θ_b están incluidos en los costes generalizados C_a y C_b, y que estos valores son conocidos. Para evitar la sobrespecificación de este modelo consideramos que el parámetro α^a, que define la cuota de mercado del modo coche, ha sido fijado $\alpha^a = 0$, y estimamos respecto a él el parámetro α^b.

Hemos empleado ambas metodologías: una secuencial y una simultánea para calibrar el vector de parámetros y la matriz O-D (que consta de un sólo par). Hemos asumido conocida una matriz O-D desactualizada y un vector de flujo actual, obtenido por observaciones directas. Esta información se muestra en la tabla 4.5

La simplicidad de la red de prueba se emplea para describir explícitamente las soluciones del modelo de equilibrio mediante cuatro ecuaciones. Ambas metodologías las hemos basado en observaciones de flujo y hemos empleado dos métricas para la función F_2. La primera es la NLLS y la segunda una del tipo WNLLS.

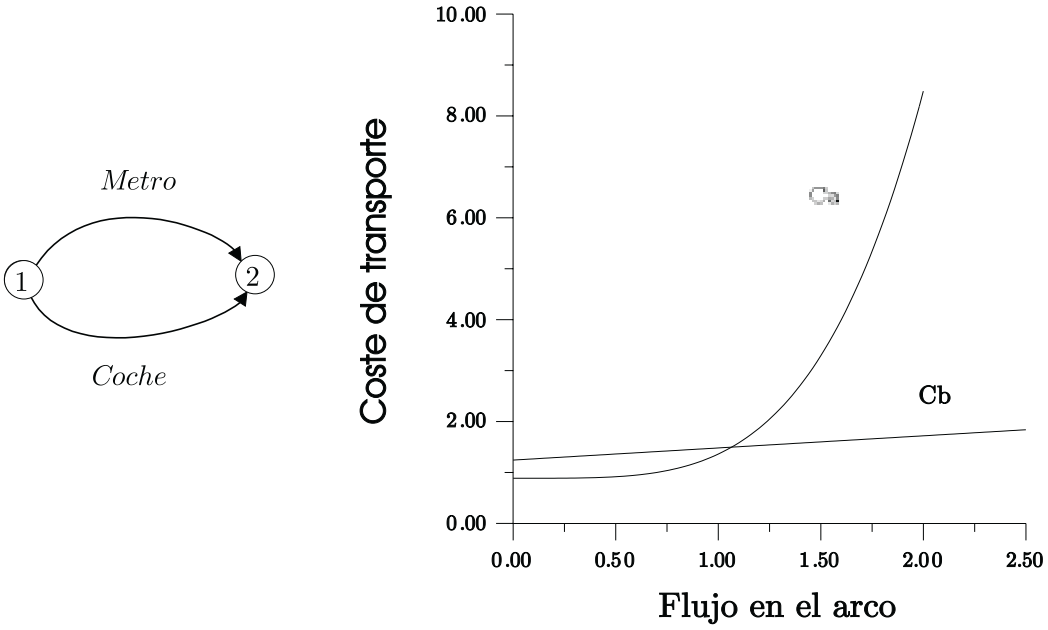


Figura 4.1: Red de prueba para el CDAM

Tabla 4.5: Base de datos para la estimación-calibración secuencial del TAP-M

Parámetros verdaderos	$\bar{g}$	β_1	α^b
	4.0	0.5011	0.5967
Datos	$\hat{f}_a$	$\hat{f}_b$	$\hat{g}_{desac.}$
	1.6	2.4	1.0
Coste de viajes	$c_a(f_a) = 0.8859 + 0.4751(f_a)^4$		
	$c_b(f_b) = 1.4285 + 0.2380 f_b$		

El problema de calibración es:

$$\begin{aligned} \text{minimizar}_{\,\beta_1,\alpha^b}\ & Z = F_2(f_a, f_b, \hat{f}_a, \hat{f}_b) \\ \text{sujeto a}\quad & g_a + g_b = \hat{g}_{desac.} \\ & g_a = f_a \\ & g_b = f_b \\ & C_a(f_a) + \tfrac{\ln g_a}{\beta_1} = C_b(f_b) + \tfrac{\ln g_b + \alpha^b}{\beta_1}. \end{aligned}$$

El problema de estimación de la matriz O-D es:

$$\begin{aligned} \text{minimizar}_{\,\bar{g}}\ & Z = F_2(f_a, f_b, \hat{f}_a, \hat{f}_b) \\ \text{sujeto a}\quad & g_a + g_b = \bar{g} \\ & g_a = f_a \\ & g_b = f_b \\ & C_a(f_a) + \tfrac{\ln g_a}{\hat{\beta}_1} = C_b(f_b) + \tfrac{\ln g_b + \hat{\alpha}^b}{\hat{\beta}_1}. \end{aligned}$$

donde $\hat{\beta}_1$ y $\hat{\alpha}^a$ son las estimación de los parámetros obtenidos en el problema de calibración.

La tabla 4.6 muestra los resultados y la interacción entre patrón de viaje y congestión. Un usuario elige el patrón de viaje (en este caso modo de transporte) en función de dos tipos de factores. El primer tipo está representado en el coste generalizado y tiene en cuenta factores como el tiempo de

viaje, tarifas, etc. y el otro tipo obedece a factores sociales como número de licencias de automóvil, distribución demográfica, etc. En este modelo el parámetro α^b tiene en cuenta este tipo de factores y el parámetro β_1 pondera la importancia relativa de las dos clases de factores. El hecho de que se asuma una matriz desactualizada, implica implícitamente un nivel de congestión en la red, por tanto, si este nivel es incorrecto quedaría una variabilidad en el patrón de viaje atribuible al factor congestión, que intentaría ser explicado por el otro factor, representado por α^b. En los flujos observados, el número de usuarios en metro es mayor que el de coche privado, $\hat{f}_b = 2.4$ frente a $\hat{f}_a = 1.6$. Esta diferencia se explica en el TAP-M, por la gran diferencia entre los costes generalizados en cada alternativa, pero en la estimación secuencial se explica porque los usuarios prefieren la alternativa metro frente a la del coche para el mismo coste de viaje. Obsérvese en la tabla 4.6 que el signo del parámetro α^b es negativo, que es un signo contrario al real. Estos errores en la estimación del modelo producen más tarde una estimación por defecto de la matriz O-D, debido a que si la cantidad de viajes es incrementada, ésta es erróneamente distribuida por modos produciendo flujos diferentes a los reales.

Tabla 4.6: Metodología secuencial de calibración-estimación

Métrica	Fase	g_a	g_b	$\bar{g}$	β_1	α^b
NLLS	Calibración	0.1009	0.9901	Fijo	1.2149	-3.1057
	Estimación	0.4956	2.7078	3.2035	Fijo	Fijo
WNLSS	Calibración	0.3996	0.6003	Fijo	0.8127	-0.9542
	Estimación	1.1739	2.5119	3.6858	Fijo	Fijo

Para concluir, observar que los valores verdaderos del modelo son la solución óptima del CDAM. Por lo que la dificultad está en elaborar algoritmos eficaces para resolver el CDAM. Puede ocurrir que no existiesen algoritmos para resolver el CDAM, o que si los hubiera, la aproximación obtenida fuese peor que la solución encontrada por la metodología secuencial. Por esta razón proponemos un algoritmo heurístico para el CDAM basado en que es posible aplicar la metodología secuencial, es decir, que existen algoritmos para resolver el problema de calibración y el de estimación de la matriz O-D. Este algoritmo se basa en repetir iterativamente la metodología secuencial, es decir, después de encontrar una nueva matriz O-D reiniciamos el procedimiento. Los resultados obtenidos con este algoritmo son ilustrados en la tabla 4.7. Notar que el procedimiento converge al verdadero valor de la matriz y del parámetro α^b pero su convergencia parece bastante lenta. Otra observación es que la velocidad de convergencia depende del tipo de métrica para F_2. La métrica WNLSS obtiene los mejores resultados en ambas metodologías, en la metodología secuencial produce mejores estimaciones y usada en el CDAM conduce a mejores comportamientos numéricos.

Tabla 4.7: Algoritmo heurístico calibración-estimación para el CDAM

.

Métrica	Fase	g_a	g_b	$\bar{g}$	β_1	α^b
NLLS	Calibración	0.1009	0.9901	Fijo	1.2149	-3.1057
1	Estimación	0.4956	2.7078	3.2035	Fijo	Fijo
2	Calibración	1.2013	2.20021	Fijo	1.7041	-0.5613
	Estimación	1.2561	2.4388	3.6950	Fijo	Fijo
3	Calibración	1.4779	2.2170	Fijo	1.1648	0.9887
	Estimación	1.4951	2.4089	3.9041	Fijo	Fijo
4	Calibración	1.5520	2.3520	Fijo	0.9011	1.0753
	Estimación	1.5567	2.4038	3.9600	Fijo	Fijo
WNLSS	Calibración	0.3996	0.6003	Fijo	0.8127	-0.9542
1	Estimación	1.1739	2.5119	3.6858	Fijo	Fijo
2	Calibración	1.4738	2.2119	Fijo	1.1770	0.9746
	Estimación	1.4919	2.4139	3.9058	Fijo	Fijo
3	Calibración	1.5623	2.3434	Fijo	0.8734	1.1057
	Estimación	1.5677	2.4042	3.9270	Fijo	Fijo
4	Calibración	1.5888	2.3832	Fijo	0.8065	1.1414
	Estimación	1.5904	2.4012	3.9917	Fijo	Fijo

4.4 Algoritmos heurísticos para el CDAM

En el ejemplo numérico anterior se plantea un algoritmo heurístico para la resolución del CDAM. Este algoritmo es conceptual ya que se basa en la existencia de algoritmos para resolver los problemas de calibración y de estimación de la matriz O-D, en problemas reales de grandes dimensiones. En esta sección, profundizamos en el modo de obtener algoritmos operativos para problemas grandes dimensiones. Desarrollaremos una clase de algoritmos heurísticos para el CDAM basada en la suposición de que la función objetivo del nivel inferior (el TAP-M) es estrictamente creciente[1]. Bajo esta hipótesis, existe un único flujo en equilibrio y una única partición modal de la matriz O-D para cada valor de las variables de decisión del nivel superior. Esto permite definir las llamadas *funciones de respuesta* o *funciones de reacción* $\mathbf{f} = \Phi(\bar{\mathbf{g}}, \Theta)$ y $\mathbf{g} = \Psi(\bar{\mathbf{g}}, \Theta)$ para cada matriz O-D $\mathbf{0} \leq \bar{\mathbf{g}} \leq \mathbf{b}$ y para cada valor de los parámetros $\Theta \in C$. Si estas funciones fuesen conocidas explícitamente, el problema binivel podría ser transformado a un problema de optimización de un sólo nivel, mediante el reemplazamiento del problema de asignación TAP-M por sus funciones de reacción Φ y Ψ.

El algoritmo propuesto para el resolver el CDAM tiene un esquema iterativo que se muestra en la tabla 4.8.

Tabla 4.8: Algoritmo heurístico para la resolución del CDAM

0. *(Inicialización).* Determinar un valor inicial de la matriz $\bar{\mathbf{g}}^0$ y del vector de parámetros Θ^0. Tomar $t = 0$.
1. *(Problema del nivel inferior.)* Resolver TAP-M para $\bar{\mathbf{g}}^t$ y Θ^t, obteniendo $\mathbf{f}^t = \Phi(\bar{\mathbf{g}}^t, \Theta^t)$
2. (*CDAM(t)).* El próximo valor del par $(\bar{\mathbf{g}}, \Theta)$ se obtiene como solución del siguiente problema binivel

$$\begin{array}{ll} \text{minimizar}_{\,(\bar{\mathbf{g}},\Theta)} & \tilde{F}(\bar{\mathbf{g}}, \mathbf{f}, \mathbf{g}) = \mu_1 F_1(\bar{\mathbf{g}}, \hat{\mathbf{g}}) + \mu_2 F_2(\mathbf{f}, \hat{\mathbf{f}}) + \mu_3 F_3(\mathbf{g}, \tilde{\mathbf{g}}) \\ \text{sujeto a} & 0 \leq \bar{\mathbf{g}} \leq \mathbf{b} \\ & \Theta \in C \subset \mathbb{R}^{\ell} \\ & (\mathbf{f}, \mathbf{g}) \in \arg\min\ S(\mathbf{f}^t, \Theta) + \nabla S(\mathbf{f}^t, \Theta)(\mathbf{y} - \mathbf{f}^t) + R(\mathbf{q}, \Theta) \\ & \qquad \text{sujeto a:} \\ & \qquad (\mathbf{y}, \mathbf{q}) \in \tilde{\Omega}(\bar{\mathbf{g}}) \end{array} \qquad [\text{CDAM}(t)]$$

   Denotar una solución del CDAM(t) por $(\bar{\mathbf{g}}^{t+1}, \Theta^{t+1})$
3. *(Criterio de convergencia).* Si se satisface el criterio de convergencia parar, en caso contrario tomar $t = t + 1$ y regresar el paso 1.

CDAM(t) está definido mediante la sustitución del TAP-M por una aproximación suya que coincide con el subproblema del algoritmo de Evans para obtener la dirección de descenso y se obtiene linealizando los costes en los arcos en la solución actual. Nos referiremos a esta aproximación por el nombre TAP-M(t). Este heurístico alcanza una solución aproximada del problema CDAM mediante la resolución de una sucesión de problemas binivel. Denotamos por $(\Phi_t(.), \Psi_t(.))$ la aplicación multievaluada de respuesta del problema TAP-M (t) (submodelo del CDAM(t)), esto es

$$(\Phi_t(\bar{\mathbf{g}}, \Theta), \Psi_t(\bar{\mathbf{g}}, \Theta)) = \{(\mathbf{f}, \mathbf{g}) \,/\, (\mathbf{f}, \mathbf{g}) \text{ es solución del TAP-M}(t) \text{ para}(\bar{\mathbf{g}}, \Theta)\}$$

El efecto de la congestión ha sido eliminado en los subproblemas TAP-M (t) provocando que el coste en los arcos o en los caminos sea independiente del flujo en éstos. Denotamos por $\{\mathbf{C}^*_\omega(t), \mathbf{C}^{*c}_\omega(t)\}_{\omega \in W}$ los costes de equilibrio en la $t-$ésima iteración. $\mathbf{C}^*_\omega(t)$ representa el coste de equilibrio por modos y $\mathbf{C}^{c*}_\omega(t)$ el coste de equilibrio para el modo combinado a través de los nodos de transferencia. Esta

[1] Cuestión que se garantiza cuando las funciones de coste en los arcos son estrictamente monótonas.

simplificación permite calcular la partición modal de la demanda mediante las funciones logit

$$\begin{array}{rcll} g^k_\omega & = & G^k_\omega\left(\Theta, \mathbf{C}^*_\omega(t), \mathbf{C}^{*c}_\omega(t)\right)\bar{\mathbf{g}}_\omega, & k \in \{a,b,c\} \quad \omega \in W \\ g^c_{\omega,t} & = & G^c_\omega\left(\Theta, \mathbf{C}^*_\omega(t), \mathbf{C}^{*c}_\omega(t)\right) G^c_{\omega,t}\left(\Theta, \mathbf{C}^{*c}_\omega(t)\right)\bar{\mathbf{g}}_\omega, & t \in T_\omega, \quad \omega \in W, \end{array} \tag{4.9}$$

donde $\{g^k_\omega\}$ es el número de usuarios de la demanda ω que viajan en el modo $k \in \{a,b,c\}$ y $\{g^c_{\omega,t}\}$ es el número de viajeros de la demanda ω que lo hacen mediante viaje combinado a través del nodo de transferencia t. Las relaciones (4.9) definen la función de reacción $\mathbf{g} = \Psi_t(\bar{\mathbf{g}}, \Theta)$ para el CDAM(t).

Los caminos óptimos del TAP-M para los datos actuales $(\bar{\mathbf{g}}^t, \Theta^t)$ también son óptimos para el TAP-M(t) para cualquier combinación $(\bar{\mathbf{g}}, \Theta)$ de la matriz O-D y del vector de parámetros debido a que no existe congestión. El flujo en estos caminos óptimos está restringido a satisfacer la demanda total, la partición modal y la distribución por intercambiadores de acuerdo con (4.9). Denotamos este conjunto de caminos óptimos en la iteración t por $P^*_\omega(t) := \cup_{k\in\{a,b,c\}} P^{k*}_\omega(t)$ para todo $\omega \in W$, donde $P^{k*}_\omega(t)$ es el conjunto de caminos óptimos para el modo k, para el par ω en la iteración $t-$ésima. El conocimiento de estos caminos nos permite calcular la función de reacción $\mathbf{f} \in \Phi_t(\bar{\mathbf{g}}, \Theta)$ para el CDAM(t) y obteniendo la siguiente expresión

$$\Phi_t(\bar{\mathbf{g}}, \Theta) = \left\{\mathbf{f} \,/\, \mathbf{f} = \delta^{\mathbf{f}}\mathbf{h}, \bar{\mathbf{g}} = \delta^{\bar{\mathbf{g}}}\mathbf{h} \text{ y } \mathbf{h} \in \Omega(t)\right\},$$

donde $\Omega(t) = \{\mathbf{h} \,/\, h_p = 0 \text{ para todo } p \notin \cup_{\omega\in W} P^*_\omega(t)\}$.

La ventaja de cada CDAM(t) con respecto al CDAM es que las funciones de reacción se pueden calcular explícitamente transformando el CDAM(t) en un problema de optimización de un solo nivel. El siguiente teorema muestra que si el punto $(\bar{\mathbf{g}}, \Theta)$ resuelve el subproblema CDAM(.) obtenido en el mismo punto $(\bar{\mathbf{g}}, \Theta)$, entonces $(\bar{\mathbf{g}}, \Theta)$ es un mínimo local para el CDAM. Esto garantiza que si la iteración generada por el algoritmo llega a un punto fijo (o equivalentemente converge en un número finito de iteraciones), entonces se obtiene un mínimo local. Bajo ciertas condiciones de continuidad, quizás, se pudiera demostrar la convergencia de la heurística a este tipo de puntos fijos y por tanto se podría demostrar la convergencia de este algoritmo a mínimos locales del CDAM.

Teorema 4.4.1 (Condición suficiente de óptimo local para el CDAM) *Sea* $(\bar{\mathbf{g}}^*, \Theta^*)$ *un elemento de* $(\mathbf{0}, \mathbf{b}) \times C$*, sea* $\mathbf{h}^*$ *el flujo de equilibrio en los caminos para el par* $(\bar{\mathbf{g}}^*, \Theta^*)$*, y CDAM(*$*$*) denota el problema CDAM(t) en el punto* $(\bar{\mathbf{g}}^t, \Theta^t) = (\bar{\mathbf{g}}^*, \Theta^*)$*. Si* $\mathbf{h}^*$ *es una solución no degenerada y* $(\bar{\mathbf{g}}^*, \Theta^*)$ *es un mínimo local para el CDAM(*$*$*) entonces* $(\bar{\mathbf{g}}^*, \Theta^*)$ *también es un mínimo local para el CDAM.*

Demostración. Daremos la demostración para el modelo de equilibrio TAP-M, pero este resultado también es válido para otros modelos combinados. Para abordar la demostración en una forma general, consideraremos que se está aplicando un modelo de equilibrio cuyas condiciones de equilibrio pueden formularse del siguiente modo: un vector de flujo en los caminos $\mathbf{h}$ está en equilibrio si existe unos coeficientes de coste por par λ_ω con $\omega \in W$ cumpliendo que

$$\begin{array}{l} \text{si } h_p > 0 \text{ entonces } \lambda_\omega = \lambda_p \\ \text{si } h_p = 0 \text{ entonces } \lambda_\omega \le \lambda_p \end{array} \quad \forall p \in P_\omega,\ \forall \omega \in W.$$

Por ejemplo, para el modelo TAP-M estos coeficiente λ_p para todo p, son los *costes extendidos* de los caminos y tienen en cuenta la elección de la ruta, modo y nodo de intercambio. En otros modelos, por ejemplo el TAP, representarían los costes generalizados.

Sea

$$\begin{aligned} \lambda : (\mathbf{0}, \mathbf{b}) \times C &\to \mathbb{R}^{\bar{c}} \\ (\bar{\mathbf{g}}, \Theta) &\to \lambda(\bar{\mathbf{g}}, \Theta), \end{aligned}$$

donde $\bar{c}$ es el cardinal del conjunto de caminos en la red multimodal. Las funciones coordenadas $\lambda_p(\bar{\mathbf{g}}, \Theta)$ $p \in P_\omega,\ \omega \in W$ son los costes extendidos de la ruta p en el equilibrio para el par de datos $(\bar{\mathbf{g}}, \Theta)$. Estas funciones son continuas en $(\bar{\mathbf{g}}^*, \Theta^*) \in (0, \mathbf{b}) \times C$ para el TAP-M suponiendo que las funciones del coste en los arcos lo sean. La demostración es fácil y se basa en la representación

explícita de las condiciones de equilibrio. Esta es la hipótesis principal que debe satisfacer el modelo de equilibrio.

Por hipótesis $\mathbf{h}^*$ es un punto no degenerado, lo que significa que

$$\min\{\lambda_p(\bar{\mathbf{g}}^*,\Theta^*) - \lambda_\omega(\bar{\mathbf{g}}^*,\Theta^*) \,/\; \mathbf{h}_p^* = 0,\ p \in P_\omega\} > \varepsilon > 0,$$

donde $\lambda_\omega(\bar{\mathbf{g}}^*,\Theta^*) = \lambda_p(\bar{\mathbf{g}}^*,\Theta^*)$ para todo $p \in P_\omega^*$. Esta relación implica que para todo $\omega \in W$ y para todo $p \in P_\omega - P_\omega^*$

$$\lambda_\omega(\bar{\mathbf{g}}^*,\Theta^*) + \frac{\varepsilon}{2} < \lambda_p(\bar{\mathbf{g}}^*,\Theta^*) - \frac{\varepsilon}{2}. \tag{4.10}$$

Sea $\frac{\varepsilon}{2} > 0$, debido a que λ es continua en $(\bar{\mathbf{g}}^*,\Theta^*)$, existe un entorno V de $(\bar{\mathbf{g}}^*,\Theta^*)$ cumpliendo:

$$\|\lambda(\bar{\mathbf{g}},\Theta) - \lambda(\bar{\mathbf{g}}^*,\Theta^*)\|_\infty < \frac{\varepsilon}{2}, \quad \forall(\bar{\mathbf{g}},\Theta) \in V,$$

donde $\|\cdot\|_\infty$ es la norma del supremo. Entonces todas la componentes $p \in P_\omega$ cumplen

$$|\lambda_p(\bar{\mathbf{g}},\Theta) - \lambda_p(\bar{\mathbf{g}}^*,\Theta^*)| < \frac{\varepsilon}{2}, \quad \forall(\bar{\mathbf{g}},\Theta) \in V,$$

en forma equivalente se cumple la siguiente relación

$$\lambda_p(\bar{\mathbf{g}},\Theta) - \tfrac{\varepsilon}{2} < \lambda_p(\bar{\mathbf{g}}^*,\Theta^*) < \lambda_p(\bar{\mathbf{g}},\Theta) + \tfrac{\varepsilon}{2}, \quad \forall(\bar{\mathbf{g}},\Theta) \in V, \quad \forall p \in \cup_{\omega\in W} P_\omega \tag{4.11}$$

Sea $p^* \in P_\omega^*$, $p' \in P_\omega - P_\omega^*$, para cualquier $\omega \in W$, empleando (4.10) y (4.11) se cumple la siguiente relación

$$\lambda_{p^*}(\bar{\mathbf{g}},\Theta) < \lambda_{p^*}(\bar{\mathbf{g}}^*,\Theta^*) + \frac{\varepsilon}{2} < \lambda_{p'}(\bar{\mathbf{g}}^*,\Theta^*) - \frac{\varepsilon}{2} < \lambda_{p'}(\bar{\mathbf{g}},\Theta), \quad \forall(\bar{\mathbf{g}},\Theta) \in V.$$

Esto demuestra que si el TAP-M es perturbado en el entorno V, entonces la solución de equilibrio podría obtenerse empleando únicamente los caminos pertenecientes a P^*, que también resuelven el problema original. Esto justifica la siguiente relación:

$$(\Phi(\bar{\mathbf{g}},\Theta),\Psi(\bar{\mathbf{g}},\Theta)) \in (\Phi_*(\bar{\mathbf{g}},\Theta),\Psi_*(\bar{\mathbf{g}},\Theta)), \quad \forall(\bar{\mathbf{g}},\Theta) \in V, \tag{4.12}$$

donde (Φ_*,Ψ_*) son las funciones de reacción para TAP-M$(*)$. Por otro lado, como $(\bar{\mathbf{g}}^*,\Theta^*)$ es un mínimo local de CDAM$(*)$ entonces existe un entorno U cumpliendo

$$\tilde{F}(\bar{\mathbf{g}},\mathbf{f},\mathbf{g}) \geq \tilde{F}(\bar{\mathbf{g}}^t,\mathbf{f}^t,\mathbf{g}^t), \quad \forall(\bar{\mathbf{g}},\Theta) \in U \cap ((0,\mathbf{b}) \times C), \quad \forall(\mathbf{f},\mathbf{g}) \in (\Phi_*(\bar{\mathbf{g}},\Theta),\Psi_*(\bar{\mathbf{g}},\Theta)) \tag{4.13}$$

Empleando las relaciones (4.12) y (4.13) obtenemos

$$F(\bar{\mathbf{g}},\Theta) = \tilde{F}(\bar{\mathbf{g}},\Phi(\bar{\mathbf{g}},\Theta),\Psi(\bar{\mathbf{g}},\Theta)) \geq \tilde{F}(\bar{\mathbf{g}}^t,\mathbf{f}^t,\mathbf{g}^t) = F(\bar{\mathbf{g}}^*,\Theta^*), \quad \forall(\bar{\mathbf{g}},\Theta) \in V \cap U \cap ((0,\mathbf{b}) \times C),$$

y prueba que $(\bar{\mathbf{g}}^*,\Theta^*)$ es un mínimo local para el CDAM. □

4.4.1 Aproximaciones al CDAM(t) mediante funciones de selección

El CDAM(t) se puede formular empleando las variables de flujo en los arcos y con ayuda de las funciones de reacción por

$$\min_{(\bar{\mathbf{g}},\Theta)} \{\tilde{F}(\bar{\mathbf{g}},\mathbf{f},\mathbf{g}) \,/\; \mathbf{f} \in \Phi_t(\bar{\mathbf{g}},\Theta),\ \mathbf{g} = \Psi_t(\bar{\mathbf{g}},\Theta),\ \Theta \in C \text{ y } \mathbf{0} \leq \bar{\mathbf{g}} \leq \mathbf{b}\}$$

Existen dos motivos para no resolver de forma exacta el CDAM(t). La primera es mantener un equilibrio entre la precisión alcanzada y el coste computacional de hacerlo. La segunda es que, para describir el conjunto $\Phi_t(\mathbf{g}, \Theta)$, es obligatorio almacenar todos los caminos con coste positivo en el equilibrio alcanzado en la iteración t. En el siguiente apartado se analizan distintas estrategias empleadas para la aproximación del CDAM(t) mediante la *funciones de selección*.

Una estrategia para aproximar el problema CDAM(t) consiste en seleccionar una función de la aplicación multievaluada Φ_t y resolver exactamente este "fácil" problema aproximado. Una función $\mathbf{f} : \mathbb{R}^s \to \mathbb{R}^{|A \cup B|}$ es una selección de $\Phi_t(.)$ si cumple $\mathbf{f}(\bar{\mathbf{g}}, \Theta) \in \Phi_t(\bar{\mathbf{g}}, \Theta), \forall(\bar{\mathbf{g}}, \Theta)$.

En otras aplicaciones se han empleado la *selección optimista* definida por

$$\mathbf{f}_o(\bar{\mathbf{g}}, \Theta) \in \arg \ \text{minimizar}_{\mathbf{f}}\{F_2(\mathbf{f}, \hat{\mathbf{f}}) \,/\, \mathbf{f} \in \Phi_t(\bar{\mathbf{g}}, \Theta)\},$$

y *la selección pesimista*

$$\mathbf{f}_p(\bar{\mathbf{g}}, \Theta) \in \arg \ \text{maximizar}_{\mathbf{f}}\{F_2(\mathbf{f}, \hat{\mathbf{f}}) \,/\, \mathbf{f} \in \Phi_t(\bar{\mathbf{g}}, \Theta)\}$$

Ambos problemas están bien definidos debido a que $\Phi_t(\bar{\mathbf{g}}, \Theta)$ es un poliedro compacto (politopo) y F_2 es una función continua. Notar que si F_2 es estrictamente convexa, el problema tiene solución única.

A priori una selección interesante es la sugerida por Dempe en [66], la cual considera una regularización de la función objetivo del nivel inferior. Para el CDAM(t) esta selección se formula

$$\mathbf{f}(\bar{\mathbf{g}}, \Theta) \in \arg \text{minimizar}_{(\mathbf{y},\mathbf{q})} \left\{ S(\mathbf{f}^t, \Theta) + \nabla S(\mathbf{f}^t, \Theta)(\mathbf{y} - \mathbf{f}^t) + R(\mathbf{q}, \Theta) + \alpha_t \tilde{F}(\bar{\mathbf{g}}, \mathbf{f}, \mathbf{g}) : (\mathbf{y}, \mathbf{q}) \in \tilde{\Omega}(\bar{\mathbf{g}}) \right\},$$

donde $\tilde{F}$ es la función objetivo del nivel superior, $\{\alpha_t\}$ es una sucesión de números reales cumpliendo que $\alpha_t \to 0$ cuando $t \to \infty$. La relevancia de esta función de selección es que requiere de un conjunto pequeño de camino con flujo en el equilibrio, quizás un camino por cada par de demanda $\omega \in W$.

Empleando la selección $\mathbf{f}_t(\bar{\mathbf{g}}, \Theta)$, el problema CDAM($t$) será aproximado por el siguiente

$$\min_{(\bar{\mathbf{g}}, \Theta)} \{\tilde{F}(\bar{\mathbf{g}}, \mathbf{f}_t(\bar{\mathbf{g}}, \Theta), \mathbf{g}) \,/ \quad \mathbf{g} = \Psi_t(\bar{\mathbf{g}}, \Theta), \ \Theta \in C \text{ y } \mathbf{0} \leq \bar{\mathbf{g}} \leq \mathbf{b}\}$$

Cada selección $\mathbf{f}_t(\bar{\mathbf{g}}, \Theta)$ conduce a diferentes aproximaciones del CDAM(t) y dada la generalidad de $\mathbf{f}_t(\bar{\mathbf{g}}, \Theta)$, origina diferentes algoritmos heurísticos. Una selección está caracterizada por un subconjunto arbitrario del conjunto de caminos óptimos de TAP-M en $(\bar{\mathbf{g}}^t, \Theta^t)$ y un conjunto de reglas para asignar el flujo a estos caminos.

Para concluir esta discusión consideraremos un caso especial donde la selección es una función lineal $\mathbf{f}_t(\bar{\mathbf{g}}, \Theta)$. Algoritmos de este tipo han sido desarrollados por Yang en [241] para el problema de estimar matrices O-D en redes de tráfico. Una primera selección está definida mediante la proporción de flujo en los arcos en función de la matriz O-D. En nuestro contexto multimodal esta selección vendría definida por

$$f_l = \sum_{\omega \in W} \left[\sum_{k \in \{a,b\}} z^k_{l,\omega} g^k_\omega + \sum_{t \in T_\omega} z^c_{l,\omega,t} g^c_{\omega,t} \right], \quad l \in A \cup B$$

donde $\mathbf{g} = \Psi_t(\bar{\mathbf{g}}, \Theta)$,

$$z^k_{l,\omega} = \frac{\sum_{p \in P^{k*}_\omega(t)} \delta_{l,p} h_p(t)}{g^{k*}_\omega(t)}, \quad k \in \{a,b\}, \ \omega \in W, \quad l \in A \cup B;$$
$$z^c_{l,\omega,t} = \frac{\sum_{p \in P^{c*}_{\omega,t}(t)} \delta_{l,p} h_p(t)}{g^{c*}_{\omega,t}(t)}, \quad \omega \in W, \quad t \in T_\omega, \quad l \in A \cup B.$$

Las variables $h_p(t), g^{k*}_\omega(t),$ y $g^{c*}_{\omega,t}(t)$ son los valores óptimos del TAP-M para el $(\bar{\mathbf{g}}^t, \Theta^t)$. En esta selección los caminos son empleados proporcionalmente al uso tenido en la solución $(\bar{\mathbf{g}}^t, \Theta^t)$, con

respecto a cada modo y a cada intercambiador. Los coeficientes $\mathbf{Z} = [z^k_{l,\omega}, z^c_{l,\omega,t}]$ se denominan *factores de influencia.*

Un segundo algoritmo se obtiene definiendo los factores de influencia $\mathbf{Z}$ como las derivadas del flujo en los arcos respecto a las variables de demanda. En el caso aquí desarrollado obtendríamos $\mathbf{Z} = [\tilde{z}^k_{l,\omega}, \tilde{z}^c_{l,\omega,t}]$ donde $\tilde{z}^k_{l,\omega}, \tilde{z}^c_{l,\omega,t}$ es calculado por

$$\tilde{z}^k_{l,\omega} = \frac{\partial f_l}{\partial g^k_\omega}, \quad k \in \{a,b\}, \quad \omega \in W,$$
$$\tilde{z}^c_{l,\omega,t} = \frac{\partial f_l}{\partial g^c_{\omega,t}}, \qquad t \in T_\omega, \quad \omega \in W,$$

donde las derivadas se calculan usando análisis de sensibilidad para la solución en equilibrio. Ver Tobin y Friesz [226] para el cálculo. No es obvio reconocer este algoritmo como un caso particular de función de selección.

Tobin y Friesz en [226] mostraron, bajo la condición de complementariedad estricta y no degeneración para $\mathbf{h}(t)$, que si el problema TAP es perturbado respecto a la matriz $\bar{\mathbf{g}}$, esto es, si la reemplazamos por $\bar{\mathbf{g}} + \epsilon$, donde ϵ es una perturbación de la demanda, entonces el problema perturbado puede también ser resuelto, si restringiésemos el problema original a los caminos con flujo positivo para la demanda $\bar{\mathbf{g}}$. Por tanto, se cumple la siguiente relación

$$\sum_{p \in P^*_\omega(t)} h_p(\bar{\mathbf{g}} + \epsilon) = g_\omega + \epsilon_\omega = \sum_{k \in \{a,b,c\}} g^k_\omega + \epsilon_\omega,$$

donde $h_p(\bar{\mathbf{g}} + \epsilon)$ es el flujo en equilibrio para el camino p de la demanda $\bar{\mathbf{g}} + \epsilon$. Tomando derivadas, en ambos lados de la igualdad respecto a las variables g^k_ω, obtenemos

$$\sum_{p \in P^{*k}_\omega(t)} \frac{\partial h_p}{\partial g^k_\omega}(\bar{\mathbf{g}} + \epsilon) = 1.$$

Por otro lado, el flujo en los caminos (debido a que el coste es monótono creciente respecto a los flujos) es creciente respecto a las demandas g^k_ω y se justifica la relación $0 \leq \frac{\partial h_p}{\partial g^k_\omega}(\bar{\mathbf{g}} + \epsilon)$ para todo $p \in P^{*k}_\omega(t)$.

Las últimas dos relaciones dan una interpretación de los coeficientes $\frac{\partial h_p}{\partial g^k_\omega}$ como los pesos de una combinación convexa. Por tanto, la selección de la función asigna las demandas g^k_ω a los caminos $p \in P^{*k}_\omega(t)$ de acuerdo a estos coeficientes, esto es, $h_p = \frac{\partial h_p}{\partial g^k_\omega}(\bar{\mathbf{g}}) g^k_\omega$.

Usando el hecho de que las derivadas de los flujos en los arcos son independientes del conjunto de puntos extremos considerados $\mathbf{h}^*$ (teorema 6. de Tobin y Friesz [226]), obtenemos

$$\frac{\partial f_l}{\partial g^k_\omega} = \sum_{p \in P^{k*}_\omega(t)} \delta_{l,p} \left(\frac{\partial h_p}{\partial g^k_\omega} \right)$$

donde δ_{lp} toma el valor 1 si el arco l está en el camino p y 0 en otro caso. Esta relación justifica que esta selección conduce al segundo algoritmo de Yang.

Notar que los algoritmos basados en los factores de influencia $\mathbf{Z}$ requieren, para poder calcularlos, las rutas y la distribución de la demanda en ellas y esta información se debe actualizar de una iteración a otra. En el cálculo del $\nabla_{\mathbf{g}}\mathbf{f}$, se debe invertir una matriz cuyas dimensiones vienen definidas por el número de coordenadas positivas de $\mathbf{h}^*(t)$. El número de caminos puede ser demasiado grande para ser computacionalmente aplicable a grandes dimensiones.

En el caso de emplear funciones de selección lineales, las métricas F_1, F_2 y F_3 juegan un papel importante para poder aplicar algoritmos eficientes de resolución. Tradicionalmente se han aplicado formas cuadráticas o basadas en la función de entropía. Dos alternativas importantes consisten en considerar las métricas derivadas de las normas $\|.\|_1$ o $\|\|_\infty$ que conducen a funciones lineales. Benitez

[16] emplea esta clase de métricas en aplicaciones reales con 1350 arcos, 750 nodos y 12,470 pares O-D, desarrolla un algoritmo similar al de Yang y otros [245] quienes emplean los factores de influencia basados en proporciones. [16] considera la $\|.\|_1$ para calcular la distancia entre la variable observada $\hat{\mathbf{x}}$ y la predicción $\mathbf{x}$, obteniendo

$$G(\mathbf{x}, \hat{\mathbf{x}}) = \sum_{i=1}^{n} |x_i - \hat{x}_i|$$

Para evitar los valores absolutos, este autor introduce $2n$ variables x_i^+ y x_i^-, para $i = 1, \ldots, n$ que representan respectivamente la sobrestimación y subestimación en las estimaciones. Las relaciones entre las variables es

$$\begin{aligned} x_i + x_i^- - x_i^+ &= \hat{x}_i, \quad i = 1, \ldots, n \\ x_i^+, x_i^- &\geq 0, \quad i = 1, \ldots, n \end{aligned} \tag{4.14}$$

Además estas variables deben satisfacer las siguientes condiciones

$$\begin{aligned} &\text{si } x_i^+ > 0 \rightarrow x_i^- = 0, \quad i = 1, \ldots, n \\ &\text{si } x_i^- > 0 \rightarrow x_i^+ = 0, \quad i = 1, \ldots, n \end{aligned} \tag{4.15}$$

entonces empleando (4.14) se puede calcular G por

$$G(\mathbf{x}, \hat{\mathbf{x}}) = \sum_{i=1}^{n} |x_i - \hat{x}_i| = \sum_{i=1}^{n} \left| x_i^+ - x_i^- \right| = \sum_{i=1}^{n} \left(x_i^+ + x_i^- \right) \tag{4.16}$$

Si deseamos minimizar la función G, podemos reemplazar G por la expresión dada en (4.16), entonces el mínimo de $\sum_{i=1}^{n} \left(x_i^+ + x_i^- \right)$ sujeto a (4.14) es alcanzado en valores positivos cumpliendo (4.15) y la sustitución se realiza correctamente. Si empleásemos la métrica inducida por $\|.\|_1$ entonces CDAM(t) puede ser formulado por

$$\begin{aligned} \text{minimizar}_{\,(\bar{\mathbf{g}},\Theta)} \; & F(\bar{\mathbf{g}}, \Theta) = \mu_1 \textstyle\sum_{\omega \in W} (g_\omega^+ + g_\omega^-) + \mu_2 \sum_{l \in A \cup B} (f_l^+ + f_l^-) \\ & + \mu_3 \textstyle\sum_{\omega \in W} \sum_{k \in \{a,b\}} (g_\omega^{k+} + g_\omega^{k-}) + \mu_3 \sum_{\omega \in W} \sum_{t \in T_\omega} (g_{\omega,t}^{c+} + g_{\omega,t}^{c-}) \\ \text{sujeto a} \quad & 0 \leq \bar{\mathbf{g}} \leq \mathbf{b}, \\ & \Theta \in C \subset \mathbb{R}^n, \\ & \mathbf{f} = \mathbf{f}_t(\bar{\mathbf{g}}, \Theta), \\ & \mathbf{g} = \Psi_t(\bar{\mathbf{g}}, \Theta), \\ & \bar{g}_\omega + g_\omega^- - g_\omega^+ = \hat{g}_\omega, \quad \forall \omega \in W, \\ & f_l + f_l^- - f_l^+ = \hat{f}_l, \quad \forall l \in A \cup B, \\ & g_\omega^k + g_\omega^{k-} - g_\omega^{k+} = \tilde{g}_\omega^k, \quad \forall \omega \in W, \quad k \in \{a, b\}, \\ & g_{\omega,t}^c + g_{\omega,t}^{c-} - g_{\omega,t}^{c+} = \tilde{g}_{\omega,t}^c, \quad \forall t \in T_\omega, \quad \forall \omega \in W, \\ & \mathbf{f}^+, \mathbf{f}^-, \mathbf{g}^+, \mathbf{g}^-, \mathbf{g}^{k+}, \mathbf{g}^{k-} \geq \mathbf{0}, \quad k \in \{a, b, c\}. \end{aligned} \tag{$\text{CDAM}_{\|\|_1}(t)$}$$

El $\text{CDAM}_{\|\|_1}(t)$ para el DAM es lineal y la única complicación en su resolución es la dimensión de este problema. Castillo y otros [41] desarrollan una forma alternativamente al método empleado por Benítez, para evitar el valor absoluto de la función objetivo que emplea sólo $n + 1$ variables. Esta transformación tiene ventajas computacionales evidentes, ya que la aplicación del primer método al problema de estudio condujo a unas 25.000 variables frente a las 12.500 que tendría el método descrito en [41].

Capítulo 5

Capacidad y tarifación de aparcamientos disuasorios: un problema de diseño de redes

Resumen

En este capítulo consideramos el problema de diseñar facilidades de aparcamiento para promocionar los viajes combinados de tipo *park'n ride*. Para ello se desarrolla un modelo continuo de diseño de redes para decidir las capacidades y tarifas de los aparcamientos empleados por estos viajes combinados. Este problema se sitúa en un contexto de planificación táctica y se asume que las decisiones sobre la localización de los aparcamientos así como otras decisiones sobre la topología de la red ya han sido tomadas previamente.

La metodología empleada para su modelización es la programación matemática binivel. En el nivel superior, una autoridad evalúa la eficiencia de la red de transporte para cada diseño de la misma. En el nivel inferior, un modelo de equilibrio con modos combinados (el TAP-M) genera la cuota de mercado y los costes de las rutas en cada modo de transporte, así como el nivel de utilización de los aparcamientos, en función de las variables de diseño de los aparcamientos consideradas. El objetivo del modelo es hacer un diseño y tarifación óptima de los aparcamientos, de modo que se minimice el coste de transporte en la red multimodal. Este objetivo conduce a considerar la capacidad y tarifas de los aparcamientos como las variables de diseño del modelo. En este trabajo se presenta una reformulación del modelo basada en emplear los costes generalizados en los aparcamientos como variables de diseño, en lugar de la tarifación y capacidad de los mismos. El modelo binivel ha sido resuelto mediante una adaptación del algoritmo del recocido simulado (SAA) a un problema continuo y restringido (variables no negativas). Se han realizado experimentos numéricos para ilustrar el uso del modelo, para comparar las dos formulaciones y para evaluar la aplicabilidad del SAA a problemas de tamaño mediano.

Palabras clave: Problema de diseño de redes, capacidad y tarifas en aparcamientos disuasorios, programación matemática binivel, modelos de equilibrio con modos combinados, algoritmo del recocido simulado para problemas continuos y restringidos.

5.1 Introducción

El crecimiento del tráfico está produciendo el colapso del sistema privado de transporte en muchas de las grandes áreas urbanas de los países desarrollados, ya que éste no tiene la posibilidad de absorber más volúmenes de tráfico. Una posible solución a este problema pasa por incrementar la cuota de mercado del transporte público. Con este fin, se están desarrollando ciertas políticas encaminadas a la promoción de los viajes combinados. Esta promoción está basada en el desarrollo de líneas de transporte público de alta calidad (rápidas y confortables), en conjunción con el uso de otras formas de transporte que alimenten a estas líneas principales y del desarrollo de atractivas facilidades de transferencia, tales como las facilidades de aparcamiento, la promoción de tarifas combinadas, etc. Todos estos incentivos inducen a los viajeros a realizar sus viajes empleando más de un modo de transporte.

Según Carrese y otros en [39] las principales políticas de aparcamientos son:

- ◇ Limitación del aparcamiento en ciertas calles y/o en el tiempo para reducir la congestión en los centros históricos.
- ◇ Introducción de los llamados aparcamientos disuasorios donde los usuarios aparcan su vehículo y continúan su viaje en transporte público, potenciando los viajes combinados y por tanto, reduciendo la congestión en los centros urbanos.
- ◇ Otras políticas de aparcamiento que están relacionadas con la satisfacción de las necesidades de aparcamiento de los residentes o de los clientes.

Las metodologías empleadas en el diseño de aparcamientos han tenido dos orientaciones fundamentales, por un lado se han desarrollado modelos de localización empleados en la investigación operativa y en la planificación regional y por otro lado modelos de asignación de la demanda. El primer grupo identifica el plan óptimo de localización basado (generalmente) en la optimización de los parámetros coste de aparcamiento y distancias a pie, sujeto a restricciones espaciales y/o económicas (Ver Odoni y Larson [184]). La segunda clase de modelos analiza diferentes planes para la satisfacción de las demandas de aparcamientos y elige uno entre ellos, según el criterio del planificador. Esta metodología asume la existencia de un modelo para la evaluación de la demanda en cada aparcamiento en función de las decisiones adoptadas. Por ejemplo, Hunt y Teply [131] consideran un modelo logit anidado para la desagregación de la demanda de aparcamientos o, por ejemplo, se emplean modelos de asignación en equilibrio para evaluar el nivel de servicio en los aparcamientos como en los trabajos de Florian y Los [84], Bifulco [27] o Carrese y otros [39]. Ambas metodologías emplean modelos de un solo nivel.

La programación matemática con restricciones de equilibrio es una importante herramienta en la planificación del transporte urbano. En el capítulo 4, se estudió una aplicación de este modelo al problema de estimación de matrices O-D y calibración de parámetros de modelos combinados. En este capítulo, estudiamos su aplicación al llamado *problema de diseño de redes* (NDP). Estos modelos se concentran en la modificación de la topología de la red de transporte o en la variación de su parametrización para alcanzar algún objetivo de eficiencia del sistema. Ejemplos de estos problemas son la determinación del número de carriles de las calles, la regulación semafórica de la red de tráfico, el establecimiento de cargas en los arcos de la red (tarifas de transporte público, tasas de aparcamiento, tasas de uso, etc.), la provisión de algún nuevo servicio de transporte público (representado como un nuevo conjunto de arcos), el establecimiento de las frecuencias de las líneas de transporte público o la construcción de una nueva calle o línea de metro.

El problema de diseñar los aparcamientos disuasorios, empleados en los viajes combinados *park'n ride*, se puede formular empleando la programación matemática binivel. El gestor del sistema de aparcamientos realiza las decisiones de inversión y tarifación de los aparcamientos en el nivel superior, de modo que se minimice el coste total del sistema, y en el nivel inferior los usuarios reaccionan a este diseño de la red de transporte cambiando su patrón de viaje, esto es, ajustando su ruta, eligiendo un modo de transporte alternativo, cambiando de aparcamiento, etc.

El NDP puede ser clasificado según la naturaleza de las variables de diseño en discreto (DNDP), si éstas son discretas (por ejemplo la selección de localizaciones de recursos), o continuo (CNDP), si son

continuas (por ejemplo la determinación de la capacidad óptima de un conjunto de arcos de la red). El problema que abordamos en este capítulo es del tipo CNDP porque las variables de diseño son las tarifas y capacidades de los aparcamientos, y éstas tienen naturaleza continua.

Se han propuesto varios algoritmos heurísticos para encontrar una solución aproximada al CNDP. Un primer algoritmo es el llamado algoritmo iterativo de optimización-asignación (IOA) que itera entre un problema de equilibrio y un problema de diseño en el que los flujos son considerados fijos. Este algoritmo no siempre conduce a un óptimo global en el diseño de la red, como fue ilustrado por Tan y otros en [225] y demostrado teóricamente por Marcotte en [158]. Otros detalles de este procedimiento han sido estudiados por Friesz y Harker en [90], Marcotte y Marquis en [162], etc.

Abdulaal y LeBlanc en [3] aplican el algoritmo de Hooke-Jeeves al NDP. Este método es computacionalmente intenso porque requiere de frecuentes evaluaciones del modelo de asignación de tráfico.

Mediante el análisis de sensibilidad se puede calcular la derivada del flujo respecto a la capacidad en los arcos (ver Tobin y Friesz [226] y Yang en [242]). Friesz y otros en [91], Yang y Yagar en [246], y Yang y otros en [247] han desarrollado varios algoritmos heurísticos basados en estas derivadas.

Suwansirikul y otros en [224] desarrollan el llamado esquema de descomposición del modelo de equilibrio (EDO), el cual optimiza los parámetros de diseño uno por uno, tratando el resto de parámetros al que se está optimizando como fijos. Después de que este conjunto de subproblemas de optimización haya sido resuelto, se actualiza la variable de diseño. Este algoritmo ha mostrado ser más eficiente que el algoritmo de Hooke-Jeeves.

Friesz y otros en [89] proponen una adaptación del algoritmo del recocido simulado. Este método es indicado para la búsqueda de óptimos globales. En este capítulo hemos adaptado esta metodología al nuevo NDP.

Meng y otros en [170] tratan el NDP formulándolo mediante un problema de optimización con un único nivel mediante el auxilio de una función marginal. Esta función y la capacidad de los arcos son obtenidas mediante el modelo de equilibrio. Para resolver la formulación alternativa (diferenciable pero no convexa) aplican un Lagrangiano aumentado que es localmente convergente.

Recientemente Patriksson y Rockafellar [199] han desarrollado un modelo para la gestión táctica del tráfico formulado como un MPEC. El modelo incluye las acciones de gestión: regulación semafórica, diseño de redes y tarifación de la congestión. Estos autores han presentado una reformulación alternativa a través de un problema no diferenciable linealmente restringido y establecen la convergencia de una clase de algoritmos para esta nueva formulación.

Las referencias anteriores son relativas al problema NDP para redes de tráfico. En estos modelos el nivel inferior está definido por el TAP que puede ser formulado bajo ciertas suposiciones. Yang y Bell en [243] presentan una revisión bibliográfica para este tipo de problemas NDP unimodales.

En el contexto de diseño de redes multimodales, Ferrari en [75], introduce un modelo de gestión del transporte urbano, donde los modos privados y públicos son gestionados conjuntamente por una autoridad que toma decisiones de tarifación de arcos, precios de billetes de transporte público y frecuencias del servicio de transporte público. El modelo de asignación bimodal está sujeto a restricciones físicas, operacionales del sistema, de capacidad y presupuestarias. La resolución de este modelo se basa en que el número de variables es pequeño y permite la obtención de polinomios de interpolación de las funciones de reacción para el conjunto de variables de diseño. Estas funciones reemplazan al nivel inferior del problema y se obtiene un problema de optimización de un solo nivel.

El diseño de intercambiadores multimodales urbanos puede ser considerado en un nivel de planificación táctico y estratégico. A un nivel estratégico se decide la localización de los intercambiadores y la red de alimentación de estos intercambiadores. Este es un ejemplo de diseño de redes multimodales que será estudiado en el capítulo 6. En un nivel táctico se estudian ciertas facilidades de intercambio. En este capítulo consideramos el precio y las capacidades de los aparcamientos de estos intercambiadores considerando que la topología de la red es fija y conocida. Este es un problema continuo de diseño de redes multimodales que denominaremos NDP-M. El NDP-M utiliza en el nivel inferior el modelo con modos combinados TAP-M desarrollado en el capítulo 1. Hemos asumido una formulación variacional del problema en el espacio de los flujos en los arcos.

El capítulo se organiza del siguiente modo. En la sección 2, presentamos el problema de diseño de redes (formulación estándar), y se da una formulación alternativa (formulación no-estándar) basada en tomar como variables de diseño el coste generalizado en los arcos asociados a los aparcamientos. En la sección 3, describimos la implementación del SAA para resolver este problema y finalmente, en la sección 4 desarrollamos una experiencia computacional con el algoritmo SAA aplicado a las dos formulaciones. El capítulo se concluye con un apéndice que ilustra la formulación no-estándar para costes de inversiones lineales y formas funcionales BPR para los costes de aparcamiento.

5.2 NDP-M: un problema de diseño de redes

El NDP-M está formulado como un modelo de programación matemática con restricciones de equilibrio. En esta sección primeramente describiremos el nivel inferior y posteriormente el problema de optimización del nivel superior.

5.2.1 El modelo de asignación del nivel inferior

En general, la mejora de la red de transporte induce cambios en los flujos en equilibrio sobre la red multimodal. Con el fin de poder predecir los nuevos patrones de viajes y poderlos integrar en el proceso de diseño de redes, se requiere de un modelo del comportamiento del usuario a estos cambios de la red.

Un buen modelo para describir el comportamiento de los usuarios debería recoger tres efectos macroscópicos: el efecto que produce la congestión en la redistribución de flujos y demandas entre modos de transporte, el efecto que tienen los aparcamientos en la generación de demanda y la competencia entre los aparcamientos.

El NDP-M emplea el modelo combinado de asignación multimodal TAP-M, desarrollado en el capítulo 1, para predecir los flujos y la partición modal de la demanda que resultarían de ciertas políticas de aparcamientos, tales como un incremento o decremento en la capacidad de aparcamiento o cambios en la tarifación de los mismos.

El TAP-M, explícitamente, identifica las elecciones efectuadas por un usuario que realiza un viaje combinado en *park'n ride*. Estas elecciones incluyen el modo de transporte, eligiendo entre modo combinado *park'n ride*, "vehículo privado" o "metro" y si el usuario realiza un viaje combinado tiene en cuenta explícitamente la elección del aparcamiento. El modelo describe la elección de la ruta hasta el aparcamiento y del aparcamiento al destino, dentro de la red de transporte público. Las dos primeras elecciones se modelan mediante un modelo logit anidado y la elección de la ruta empleando el primer principio de Wardrop. Los detalles específicos del modelo son analizados en el capítulo 1 y para detalles más generales sobre modelos combinados, se pueden consultar los excelentes libros de Sheffi [210], Ortúzar y Willumsem [189] y Patriksson [196].

Cada aparcamiento se representa mediante un arco de transferencia en la red multimodal y los parámetros de las funciones de coste tienen en cuenta factores como las tarifas y capacidades de los aparcamientos. Estas funciones dan el coste de aparcamiento (tomando el tiempo como unidades) que es la suma del tiempo empleado en aparcar (búsqueda de un espacio libre y tiempo andando hasta la parada de metro) y precio del aparcamiento.

Si el coste en los arcos $\mathbf{c}(\mathbf{f})$, tiene una matriz Jacobiana no simétrica y el modelo de asignación en redes de transporte público puede ser formulado en el espacio de flujos en los arcos, entonces el TAP-M se puede formular mediante la siguiente desigualdad variacional: encontrar un $(\mathbf{f}^*, \mathbf{g}^*) \in \Omega_{\mathbf{f}}^{\mathbf{g}}$ que verifique

$$\mathbf{c}(\mathbf{f}^*)^T(\mathbf{f}-\mathbf{f}^*) - \Lambda(\mathbf{g}^*)^T(\mathbf{g}-\mathbf{g}^*) \geq 0, \quad \forall(\mathbf{f},\mathbf{g}) \in \Omega_{\mathbf{f}}^{\mathbf{g}}, \qquad [\text{TAP-MVIP}(\mathbf{c}-\Lambda, \Omega_{\mathbf{f}}^{\mathbf{g}})]$$

donde $\Omega_{\mathbf{f}}^{\mathbf{g}}$ es el espacio de flujo en los arcos y demandas definido por

$$\Omega_{\mathbf{f}}^{\mathbf{g}} = \left\{(\mathbf{f},\mathbf{g}) : \exists \mathbf{h} \in \Omega_{\mathbf{h}} \text{ cumpliendo } \ \mathbf{f} = \delta^{\mathbf{f}}\mathbf{h} \text{ y } \mathbf{g} = \delta^{\mathbf{g}}\mathbf{h}\right\},$$

con

$$\Omega_{\mathbf{h}} = \left\{ \mathbf{h} \in \Re^{\bar{c}} \,/\, \sum_{k} \sum_{p \in P_{\omega}^{k}} h_p = \bar{g}_{\omega},\ \forall \omega \in W;\ \mathbf{h} \geq 0 \right\}.$$

El término $\mathbf{c}(\mathbf{f}^*)^T(\mathbf{f}-\mathbf{f}^*)$ tiene en cuenta la elección de la ruta en la red multimodal y el término $\Lambda(\mathbf{g}^*)^T(\mathbf{g}-\mathbf{g}^*)$ está asociado con el modelo de demanda que es un logit anidado. Esta función produce el equilibrio en las elecciones del modo de transporte y de aparcamiento. La región factible $\Omega_{\mathbf{f}}^{\mathbf{g}}$ para las variables $(\mathbf{f},\mathbf{g})$ está definida por las restricciones de conservación de flujo, restricciones auxiliares que relacionan el flujo en los caminos con el flujo en los arcos y la no negatividad del flujo en los caminos.

5.2.2 El problema de optimización del nivel superior

El NDP está elaborado para seleccionar el incremento óptimo de las capacidades (modelizadas como variables continuas) y de las tarifas de los aparcamientos disuasorios empleados en los viajes combinados *park'n ride*. Hemos formulado este problema mediante un problema binivel generalizado. En el nivel superior se deciden la capacidad y las tarifas de los aparcamientos y en el nivel inferior, el problema variacional TAP-MVIP produce los flujos en equilibrio en los arcos, $\mathbf{f}^*$, y la distribución de la demanda por modos y aparcamientos, $\mathbf{g}^*$, para el valor dado de las variables de diseño.

Asumimos que la autoridad que opera el sistema de aparcamientos elige el valor de las variables de diseño de modo que el coste total de transporte en un subconjunto de la red (por ejemplo, en la red de tráfico), $\hat{\mathcal{A}} \subset \mathcal{A}$, sea minimizado. Es posible elaborar funciones objetivo más sofisticadas basadas en la partición modal de la demanda, costes generalizados, costes de inversión, etc.

Supondremos que las variables de diseño están restringidas a satisfacer un presupuesto y que el dinero obtenido por las tarifas de aparcamientos es I para el período de planificación. Sea T el conjunto de arcos asociados a los aparcamientos gestionados por la autoridad, $a_t \geq 0$ el valor de la tasa impuesta e $I = \sum_{t\in T} f_t^* a_t$. Para dar una mayor flexibilidad a la formulación consideramos $I = \sum_{t\in T} \eta f_t^* a_t$ donde η es un parámetro. Esta formulación puede tener en cuenta subsidios públicos de dichas tasas.

Supondremos que el coste de operación para el período de planificación es C y que es la suma del coste de gestión más el coste de amortización de la expansión de la capacidad de aparcamientos. Este coste puede ser considerado como una función de la nueva capacidad instalada $k_t \geq 0$, entonces $C = \sum_{t\in T} s_t(k_t)$. Supondremos que los aparcamientos ya están localizados y tienen ya cierta capacidad instalada que deseamos mejorar. Esta capacidad adicional se representa por las variables de diseño k_t con $t \in T$.

El déficit por período de planificación es la diferencia entre C e I. Si B es el presupuesto público (o privado) para el período de planificación, entonces las restricciones presupuestarias se formulan $C - I = \sum_{t\in T}[s_t(k_t) - \eta f_t^* a_t] \leq B$.

El problema de diseño de redes con restricciones presupuestarias se formula por

$$\begin{array}{ll} \text{minimizar}_{\mathbf{x}=(\mathbf{a},\mathbf{k})} Z = \sum_{l\in\hat{\mathcal{A}}} c_l(\mathbf{f}^*,\mathbf{x}) f_l^* & \\ \text{sujeto a} & \sum_{t\in T}[s_t(k_t) - \eta f_t^* a_t] \leq B, \\ & a_t \geq 0, \quad \forall t \in T, \\ & k_t \geq 0, \quad \forall t \in T, \\ & \mathbf{c}(\mathbf{f}^*,\mathbf{x})^T(\mathbf{f}-\mathbf{f}^*) - \Lambda(\mathbf{g}^*)^T(\mathbf{g}-\mathbf{g}^*) \geq 0, \quad \forall (\mathbf{f},\mathbf{g}) \in \Omega_{\mathbf{f}}^{\mathbf{g}}, \end{array} \tag{5.1}$$

donde $\mathbf{x} = (\mathbf{a},\mathbf{k})$ es el vector de variables de diseño, que son respectivamente las tarifas y capacidades de los aparcamientos, $\mathbf{c}(\mathbf{f}^*,\mathbf{x})$ es el coste de viaje para un flujo en los arcos $\mathbf{f}^*$ y para las variables de diseño $\mathbf{x}$, y la desigualdad variacional, que define el nivel inferior, fue definida en el capítulo 1.

Alternativamente, la restricción presupuestaria la podemos incorporar en la función objetivo relajándola Lagrangianamente. Denotando la variable dual asociada a dicha restricción por λ, la función objetivo se formula por

$$\min_{\mathbf{x}=(\mathbf{k},\mathbf{a})} Z = \sum_{l\in\hat{\mathcal{A}}} c_l(\mathbf{f}^*,\mathbf{x}) f_l^* + \lambda \sum_{t\in T} [s_t(k_t) - \eta f_t^* a_t] - \lambda B. \tag{5.2}$$

La función objetivo puede ser vista como un compromiso entre un coste social (coste de congestión) y un coste monetario de inversión-gestión. El término $-\lambda B$ es constante y se eliminará de la formulación.

Suponiendo que la restricción presupuestaria es activa y no degenerada en la solución óptima entonces $\lambda > 0$. Si multiplicamos la función objetivo del nivel superior por la constante positiva $\theta = 1/\lambda$, la solución óptima del anterior problema de optimización no cambia. Esta transformación se efectúa para interpretar los costes en unidades monetarias y formulamos el problema de diseño por

$$\begin{array}{ll} \text{minimizar}_{\mathbf{x}=(\mathbf{k},\mathbf{a})} Z = \theta \sum_{l\in\hat{\mathcal{A}}} c_l(\mathbf{f}^*,\mathbf{x}) f_l^* + \sum_{t\in T} [s_t(k_t) - \eta f_t^* a_t] & \\ \text{sujeto a} \quad a_t \geq 0, \quad \forall t \in T, & \\ \qquad\quad k_t \geq 0, \quad \forall t \in T, & \\ \qquad\quad \mathbf{c}(\mathbf{f}^*,\mathbf{x})^T(\mathbf{f}-\mathbf{f}^*) - \Lambda(\mathbf{g}^*)^T(\mathbf{g}-\mathbf{g}^*) \geq 0, \quad \forall (\mathbf{f},\mathbf{g}) \in \Omega_{\mathbf{f}}^{\mathbf{g}}, & \end{array} \tag{NDP-M($\mathbf{x}$)}$$

5.2.3 Una formulación no-estándar del NDP-M

En este apartado discutiremos una formulación alternativa del NDP-M que emplea como variables de diseño los costes de aparcamientos.

Supongamos que el coste del aparcamiento $t \in T$ (conjunto de aparcamientos) está dado por la ecuación

$$y_t = c_t(f_t^*, \mathbf{x}_t) = c_t(f_t^*, a_t, k_t), \quad t \in T. \tag{5.3}$$

El coste generalizado de aparcamiento, y_t, depende del nivel de servicio del aparcamiento t, que es f_t^*, y de las variables de diseño $\mathbf{x}_t = (a_t, k_t)$. La función de coste en un arco genérico $l \in \mathcal{A}$ es $c_l(\mathbf{f},\mathbf{x})$ y puede ser formulada empleando las variables $(\mathbf{f},\mathbf{y})$. Si el arco l no está asociado a un aparcamiento, c_l solamente depende del flujo en el arco $\mathbf{f}$. En caso contrario, $l \in T$, el coste de aparcamiento es exactamente y_l. Reescribiendo $c_l(\mathbf{f},\mathbf{x})$ en función de las variables $(\mathbf{f},\mathbf{y})$, obtenemos

$$c_l(\mathbf{f},\mathbf{y}) = \begin{cases} y_l, & \text{si } \ l \in T, \\ c_l(\mathbf{f}), & \text{si } \ l \notin T. \end{cases} \tag{5.4}$$

Formularemos el problema NDP-M($\mathbf{x}$) en función de la nueva variable $\mathbf{y}$. Primeramente consideraremos el problema auxiliar definido por las variables $\mathbf{x}$ e $\mathbf{y}$.

$$\begin{array}{ll} \text{minimizar}_{\mathbf{x}=(\mathbf{a},\mathbf{k}),\mathbf{y}} Z(\mathbf{x},\mathbf{y}) = \theta \sum_{l\in\hat{\mathcal{A}}} c_l(\mathbf{f}^*,\mathbf{y}) f_l^* + \sum_{t\in T} \{s_t(k_t) - \eta f_t^* a_t\} & \\ \text{sujeto a} & \end{array} \tag{NDP-M($\mathbf{x},\mathbf{y}$])}$$

$$y_t = c_t(f_t^*, a_t, k_t), \quad \forall t \in T, \tag{5.5}$$

$$a_t \geq 0, \quad \forall t \in T, \tag{5.6}$$

$$k_t \geq 0, \quad \forall t \in T, \tag{5.7}$$

$$\mathbf{c}(\mathbf{f}^*,\mathbf{y})^T(\mathbf{f}-\mathbf{f}^*) - \Lambda(\mathbf{g}^*)^T(\mathbf{g}-\mathbf{g}^*) \geq 0, \quad \forall (\mathbf{f},\mathbf{g}) \in \Omega_{\mathbf{f}}^{\mathbf{g}}. \tag{5.8}$$

Notar que los problemas NDP-M($\mathbf{x},\mathbf{y}$) y NDP-M($\mathbf{x}$) son equivalentes.

Existe una ventaja computacional en considerar el NDP-M$(\mathbf{x}, \mathbf{y})$ en el $\mathbf{y}$-espacio, más que en el $(\mathbf{x}, \mathbf{y})$-espacio. Esta transformación se realiza mediante la *proyección o particionamiento* (ver el trabajo de Geoffrion [111]). Consideremos la siguiente formulación abreviada del NDP-M$(\mathbf{x}, \mathbf{y})$

$$\min_{\mathbf{x},\mathbf{y}} \; Z(\mathbf{x}, \mathbf{y}) \text{ s. a. } G(\mathbf{x}, \mathbf{y}) = 0, \; \mathbf{x} \in X, \; \mathbf{y} \in Y \tag{5.9}$$

donde $G(\mathbf{x}, \mathbf{y}) = 0$ es la restricción (5.5), X está definido por (5.6) y (5.7), e Y por (5.8).

La proyección de (5.9) en $\mathbf{y}$ es

$$\min_{\mathbf{y}} \; Z(\mathbf{y}) \text{ s. a. } \; \mathbf{y} \in Y \cap V,$$

donde

$$Z(\mathbf{y}) = \min_{\mathbf{x}} Z(\mathbf{x}, \mathbf{y}) \text{ s. a. } \mathbf{x} \in X, \; G(\mathbf{x}, \mathbf{y}) = 0$$

y

$$V = \{\mathbf{y} \; / \; G(\mathbf{x}, \mathbf{y}) = 0 \text{ para algún } \mathbf{x} \in X\}$$

Nótese que $Z(\mathbf{y})$ es un valor óptimo de NDP-M$(\mathbf{x}, \mathbf{y})$ para un valor fijo $\mathbf{y}$, que el conjunto V está formado por los valores de $\mathbf{y}$ para los que $Z(\mathbf{y})$ está definido, y que $Y \cap V$ puede ser visto como la proyección de la región factible de NDP-M$(\mathbf{x}, \mathbf{y})$ en el $\mathbf{y}-$espacio.

Además, el problema de la desigualdad variacional (5.8) no depende de la variable $\mathbf{x}$, entonces los flujos en equilibrio sólo dependen de la variable $\mathbf{y}$. Esta observación es clave para poder calcular explícitamente la función $Z(\mathbf{y})$. La interpretación de este hecho es que los cambios de las variables $\mathbf{a}$ y $\mathbf{k}$ afectan a las decisiones de los usuarios, debido a que producen cambios en los costes de aparcamiento. Si cambiáramos las variables de diseño de tal modo que el coste de aparcamiento no variase, entonces no cambiaría el flujo en equilibrio. Para remarcar este hecho denotamos el flujo en equilibrio en el arco para la variable $\mathbf{y}$ por $\mathbf{f}^*(\mathbf{y})$.

Ahora calcularemos la función $Z(\mathbf{y})$. Observamos que el primer término de la función objetivo no depende de $\mathbf{x}$ y juega el papel de una constante. Empleando la definición de $Z(\mathbf{y})$ obtenemos

$$Z(\mathbf{y}) = \theta \sum_{l \in \hat{\mathcal{A}}} c_l(\mathbf{f}^*(\mathbf{y}), \mathbf{y}) f_l^*(\mathbf{y}) + \Gamma(\mathbf{y})$$

donde $\Gamma(\mathbf{y})$ está definido por el siguiente submodelo:

$$\begin{aligned} \Gamma(\mathbf{y}) = \min_{\mathbf{x}=(\mathbf{a},\mathbf{k})} & \sum_{t \in T} \{s_t(k_t) - \eta f_t^*(\mathbf{y}) a_t\} \\ & \text{sujeto a} \\ y_t = c_t(f_t^*(\mathbf{y}), a_t, k_t), & \quad \forall t \in T, \\ a_t \geq 0, & \quad \forall t \in T, \\ k_t \geq 0, & \quad \forall t \in T. \end{aligned}$$

El problema anterior puede ser descompuesto en $|T|$ subproblemas, uno por cada aparcamiento $t \in T$,

$$\begin{aligned} \Gamma_t(y_t) = \min_{\mathbf{x}_t=(a_t,k_t)} & \; s_t(k_t) - \eta f_t^* a_t \\ & \text{sujeto a} \\ y_t = c_t(f_t^*, a_t, k_t), & \\ a_t \geq 0, & \\ k_t \geq 0. & \end{aligned}$$

El subproblema anterior puede ser resuelto en forma cerrada mediante la utilización de las condiciones de optimalidad de Karush-Kuhn-Tucker. En el apéndice I hemos desarrollado estas condiciones para un caso particular de las funciones c_t y s_t.

Para calcular el conjunto V, debemos caracterizar los costes de aparcamientos factibles. Empleando la interpretación de los parámetros a_t, k_t podemos suponer que el coste de aparcamiento es creciente

con la tarifa de aparcamiento a_t. Entonces $\mathbf{y}$, que representa el coste generalizado de aparcamiento, será factible para valores de $\mathbf{y}$ suficientemente grandes, debido a que podemos elegir tarifas tan elevadas como deseemos y por tanto podemos obtener cualquier coste generalizado de aparcamiento que sea lo suficientemente elevado. Por otro lado, los costes de aparcamientos están acotados inferiormente por el tiempo que se emplea en ir andando del aparcamiento a la parada de la línea de transporte público. Por eso el conjunto V será un intervalo no acotado superiormente.

Formalmente, bajo las suposiciones de que las funciones $c_t(\mathbf{f}, a_t, k_t)$ son continuas con respecto a a_t y k_t para todo valor $\mathbf{f} \geq \mathbf{0}$, que el $\lim_{a_t \to +\infty} c_t(\mathbf{f}, a_t, k_t) = +\infty$, que $c_t(\mathbf{f}, a_t, k_t)$ es estrictamente creciente en a_t y estrictamente decreciente en k_t, y que está acotada inferiormente para todo $\mathbf{f} > 0$, y que $\mathbf{f}^*(\mathbf{y}) > 0$ para todo $\mathbf{y} \in V$ entonces se puede demostrar que

$$V = \left\{ \mathbf{y} \,/\, y_t^{LB} < y_t,\ t \in T \right\} \text{ donde } y_t^{LB} = \lim_{k_t \to +\infty} c_t(\mathbf{f}^*(\mathbf{y}), 0, k_t) \tag{5.10}$$

Finalmente, obtenemos la formulación del NDP-M$(\mathbf{x}, \mathbf{y})$ en el $\mathbf{y}-$espacio, que denominaremos formulación no-estándar, y que viene dada por

$$\begin{array}{ll} \text{minimizar}_{\mathbf{y}} & Z(\mathbf{y}) = \theta \sum_{l \in \hat{\mathcal{A}}} c_l(\mathbf{f}^*(\mathbf{y}), \mathbf{y}) f_l^*(\mathbf{y}) + \Gamma(\mathbf{y}) \\ \text{sujeto a} & y_t^{LB} < y_t, \quad \forall t \in T, \\ & \mathbf{c}(\mathbf{f}^*, \mathbf{y})^T (\mathbf{f} - \mathbf{f}^*) - \Lambda(\mathbf{g}^*)^T (\mathbf{g} - \mathbf{g}^*) \geq 0, \quad \forall (\mathbf{f}, \mathbf{g}) \in \Omega_{\mathbf{f}}^{\mathbf{g}} \end{array} \qquad \text{[NDP-M}(\mathbf{y})\text{]}$$

Sea $\mathbf{x}^*(\mathbf{y})$ una solución de la aplicación $\Gamma(\mathbf{y})$ y sea $\mathbf{y}^*$ una solución óptima del NDP-M$(\mathbf{y})$, entonces se cumple que $(\mathbf{x}^*, \mathbf{y}^*)$ es una solución óptima del NDP-M$(\mathbf{x}, \mathbf{y})$ para cualquier punto $\mathbf{x}^* \in \mathbf{x}(\mathbf{y}^*)$ y además $\mathbf{x}^*$ es una solución óptima del NDP-M$(\mathbf{x})$.

Es importante resaltar que el NDP-M$(\mathbf{y})$ opera con unos costes en los arcos diferentes al NDP-M$(\mathbf{x})$. En el NDP-M$(\mathbf{y})$ se reemplazan las funciones de coste no lineales de los aparcamientos empleados en el NDP-M$(\mathbf{x})$ por unas funciones constantes, haciendo que el cálculo del estado de equilibrio sea computacionalmente más fácil.

5.3 El algoritmo del recocido simulado (SAA)

Friesz y otros demostraron en [89] la posibilidad de emplear un algoritmo de recocido simulado (SAA) para calcular un óptimo global del problema NDP para pequeñas redes de tráfico. Este método conduce a soluciones que son mejores que los métodos heurísticos desarrollados. Esta es la principal motivación para elegir el SAA para resolver el NDP-M.

En esta sección discutiremos la aplicación del SAA al problema de optimización continuo NDP-M$(\mathbf{y})$ o NDP-M$(\mathbf{x})$. Para abordar ambos problemas, de una manera unificada, denotamos por $\mathbf{v}$ la variable de diseño y por $Z(\mathbf{v})$ el valor de la función objetivo del nivel superior, en el punto $\mathbf{v}$.

Sea $\mathbf{v}_{\text{old}}$ una solución dada, seleccionaremos una solución candidata $\mathbf{v}_{\text{new}}$ del entorno $\mathcal{S}(\mathbf{v}_{\text{old}})$. La solución se acepta si reduce el coste. En caso contrario, $\mathbf{v}_{\text{new}}$ puede ser aceptada con la probabilidad

$$\exp - \left(\frac{Z(\mathbf{v}_{\text{new}}) - Z(\mathbf{v}_{\text{old}})}{K\mathcal{T}} \right) \tag{5.11}$$

y será rechazada con la probabilidad complementaria. $\mathcal{T}$ es una constante positiva y K es una constante que depende del coste del sistema. $\mathcal{T}$ es la denominada *temperatura* del proceso y permite definir pequeños o grandes movimientos en las variables de optimización.

Siguiendo el trabajo de Friesz y otros en [89], seleccionaremos una solución candidata del siguiente modo. Sea $\mathbf{v}_{\text{old}}$ el actual valor de las variables de diseño. Para determinar la solución candidata perturbamos aleatoriamente la variable actual del siguiente modo

$$\mathbf{v}_{\text{new}} = \mathbf{v}_{\text{old}} + \epsilon \tag{5.12}$$

donde $\epsilon = \mathbf{Qu}$ y $\mathbf{u}$ es un vector aleatorio $(\dots, u_t, \dots)$ cuyas componentes u_t son variables aleatorias independientes y aleatoriamente distribuidas en el intervalo $[-\sqrt{3}, \sqrt{3}]$. La matriz $\mathbf{Q}$ define la estrategia de muestreo del entorno $\mathcal{S}(\mathbf{v}_{\text{old}})$ de la actual variable de diseño, controlando la distribución del tamaño del paso

Una diferencia importante del NDP-M($\mathbf{v}$) respecto al NDP, aplicado a redes de tráfico, es que el problema de optimización NDP-M($\mathbf{v}$) está restringido. La variable de diseño está acotada inferiormente, esto es

$$\mathbf{v}_{\text{LB}} \leq \mathbf{v}_{\text{new}}$$

El valor de $\mathbf{v}_{\text{LB}}$ es $\mathbf{0}$ para el NDP-M($\mathbf{x}$) y está definido por (5.10) para el NDP-M($\mathbf{y}$). Nótese que en el NDP-M($\mathbf{y}$) la desigualdad es estricta, esto es $\mathbf{v}_{\text{LB}} < \mathbf{v}_{\text{new}}$. Bajo un punto de vista computacional, la cota inferior es reemplazada por $\mathbf{v}_{\text{LB}} + \delta$, para un valor de δ positivo y suficientemente pequeño y la desigualdad estricta es considerada como una desigualdad de menor o igual.

En la muestra aparecen soluciones no factibles, para evitar esta dificultad de aplicación en el SAA, las soluciones candidatas son proyectadas en la región factible. En el caso que estamos considerando, esta proyección se calcula fácilmente mediante

$$\mathbf{v}_{\text{new}} = \max\{\mathbf{v}_{\text{LB}}, \mathbf{v}_{\text{old}} + \epsilon\} \tag{5.13}$$

tomando el máximo componente a componente.

Vanderbilt y Louie en [231] observan que la máxima eficiencia del algoritmo SAA se obtiene cuando la mitad de las soluciones candidatas son aceptadas. Para este fin, es recomendable que la región de muestreo dependa de la "topografía" de la función objetivo. Estos autores sugirieron un mecanismo para la determinación de la matriz $\mathbf{Q}$ basado en el paseo aleatorio (soluciones aceptadas) en una fase de temperatura determinada. Considerar que los dos primeros momentos del paseo aleatorio se calculan por

$$B_t^{(n)} = \frac{1}{M^{(n)}} \sum_{m=1}^{M^{(n)}} v_t^{(m;n)}, \tag{5.14}$$

$$S_j^{(n)} = \frac{1}{M^{(n)}} \sum_{i=1}^{M^{(n)}} \left[v_i^{(m;n)} - B_i^{(n)}\right] \left[v_j^{(m;n)} - B_j^{(n)}\right] \tag{5.15}$$

donde $\mathbf{v}^{(m;n)}$ es el valor de $\mathbf{v}$ en el $m-$ésimo paso de la $n-$ésima "temperatura" y $M^{(n)}$ es el número de soluciones aceptada en la $n-$ésima temperatura. Nótese que estos cálculos son efectuados solamente con las soluciones aceptadas. $\mathbf{S}$ describe el paseo aleatorio del actual segmento.

La matriz de varianza-covarianza para la $(n+1)-$ésima temperatura, $\mathbf{s}^{(n+1)}$, es elegida como sigue

$$\mathbf{s}^{(n+1)} = \frac{\chi_s}{\beta M^{(n)}} \mathbf{S}^{(n)} \tag{5.16}$$

donde la constante χ_s es el "factor de crecimiento", típicamente > 1. El parámetro β es una constante que le hemos dado el valor 0.11 como se sugirió en el trabajo de Vanderbilt y Louie [231].

El paso $\mathbf{Q}^{(n+1)}$ se elegirá de modo que se obtenga la matriz de varianza-covarianza deseada, $\mathbf{s}^{(n+1)}$, y este paso puede ser obtenido factorizando la matriz $\mathbf{s}^{(n+1)}$ mediante la descomposición de Cholesky

$$\mathbf{s}^{(n+1)} = \mathbf{Q}^{(n+1)} \cdot (\mathbf{Q}^{(n+1)})^T \tag{5.17}$$

Para que sea posible calcular la descomposición de Cholesky la matriz $\mathbf{s}^{(n+1)}$ debe ser simétrica y definida positiva, por ser $\mathbf{s}^{(n+1)}$ una matriz de varianza-covarianza será siempre simétrica, pero podría no ser definida positiva. Un *procedimiento modificado de Cholesky*, para la factorización de una matriz indefinida, consiste en incrementar los elementos de la diagonal durante el proceso de fatorización de Cholesky (ver Nocedal y Wright [183]). Este algoritmo puede ser visto como la aplicación de la

factorización de Cholesky a la matriz modificada $\mathbf{P}\mathbf{s}^{(n+1)}\mathbf{P}^T + \mathbf{E}$, donde $\mathbf{E}$ es una matriz diagonal cuyos elementos son no negativos y $\mathbf{P}$ es una matriz de permutación de filas y columnas.

En este trabajo, hemos planteado una modificación alternativa de la factorización de Cholesky basada en la propiedad de que los elementos de la diagonal de $\mathbf{s}^{(n+1)}$ son no negativos (representan la varianza de variables aleatorias), la cual puede ser vista como la aplicación de la factorización de Cholesky a una aproximación definida positiva a $\mathbf{s}^{(n+1)}$. Hemos considerado dos casos: el primero aparece debido a la no convexidad del NDP-M($\mathbf{v}$) que genera aproximaciones no definidas positivas de la matriz Hessiana y el segundo caso es debido a la restricciones (5.13), que fuerzan a las variables de optimización a sus cotas inferiores, haciendo que todos los candidatos tomen el mismo valor en cierta variables. Esto provoca que la varianza muestral y todas las covarianzas asociadas con dichas variables sean cero, produciendo una fila y columna de ceros y haciendo, por tanto, que la matriz $\mathbf{s}^{(n+1)}$ sea singular. A continuación consideramos las aproximaciones empleadas en cada caso.

- ◇ Caso I: todos los elementos de la diagonal de $\mathbf{s}^{(n+1)}$ son positivos pero $\mathbf{s}^{(n+1)}$ no es definida positiva. Para este caso la aproximación elegida es

$$\mathbf{s}^{(n+1)} \approx \hat{\mathcal{A}}^{(n+1)} + \alpha(\mathbf{s}^{(n+1)} - \mathbf{D}^{(n+1)}), \text{ con } \alpha \in (0,1) \tag{5.18}$$

  donde $\mathbf{D}^{(n+1)}$ es la matriz diagonal de $\mathbf{s}^{(n+1)}$.

  Notar que si $\alpha \to 0$, la matriz del lado derecho tiende a $\mathbf{D}^{(n+1)}$ que es una matriz definida positiva. Esto implica que, para algún valor de α, la aproximación es definida positiva. Para obtener α realizamos varias pruebas hasta que (5.18) se cumpla.

- ◇ Caso II: al menos un elemento de la diagonal de $\mathbf{s}^{(n+1)}$ es cero. Para este caso podemos utilizar la modificación común de la factorización de Cholesky, que añade un múltiplo de la matriz identidad $\mathbf{I}$ a la matriz indefinida. Es decir, consideramos

$$\mathbf{s}^{(n+1)} \approx \mathbf{s}^{(n+1)} + \varepsilon\mathbf{I}, \text{ con } \varepsilon > 0 \tag{5.19}$$

  Empleando este procedimiento obtenemos una matriz definida positiva o una matriz que satisface el caso I. Se puede seleccionar un valor arbitrario de ε, o emplear la modificación de Gershgorin que incrementa la diagonal tanto como sea necesario para obtener que la matriz sea definida positiva (ver Nocedal y Wright [183]).

Analizado el método de muestreo empleado por el SAA, se requiere estudiar un conjunto de parámetros del SAA. La eficiencia computacional del SAA depende de este conjunto pero no existe una regla general para seleccionarlos. Su selección se realiza por ajuste individualizado para cada problema concreto. A continuación presentamos el conjunto de reglas empleadas en la implementación realizada del SAA.

(a) En la práctica el parámetro de la temperatura es disminuido a lo largo del proceso algorítmico. Denotamos respectivamente los valores iniciales y finales de la temperatura por $\mathcal{T}_0$ y $\mathcal{T}_f$. La regla empleada para decrecer este parámetro es $\mathcal{T} = \alpha\mathcal{T}$ donde α es el parámetro de *recoción*, cumpliendo $0 < \alpha < 1$. Cuando la temperatura es suficientemente baja (menor que $\mathcal{T}_f$) la búsqueda finaliza.

(b) Cuando el sistema está en equilibrio, la temperatura del sistema debe ser cambiada. Usualmente este equilibrio está caracterizado por dos parámetros: NAC que es el máximo número de configuraciones aceptadas y NRC que es el máximo número de configuraciones rechazadas consecutivamente.

(c) La probabilidad de aceptar una solución peor que la actual está definida por (6.16) y en el estadio inicial ésta es

$$p = \exp - \left(\frac{\Delta c}{K\mathcal{T}_0}\right)$$

donde Δc es el incremento del coste del sistema, esto es $c' - c^{(0)}$, donde c' y $c^{(0)}$ son los costes de la configuración nueva y de la inicial respectivamente. Si consideramos que esta probabilidad debe ser un cierto valor p, entonces K debe satisfacer

$$K = \frac{-\Delta c}{\mathcal{T}_0 \log(p)}$$

Esta relación se usa en la fase de inicialización, para calcular el valor de la constante K. Por ejemplo, sea $\mathcal{T}_0 = 1$ y se considera una probabilidad de $p = 0.1$ de aceptar una solución un 5% peor que la inicial $c^{(0)}$, en este caso $\Delta c = 0.05 \left|c^{(0)}\right|$, obteniendo

$$K = \frac{-0.05 \left|c^{(0)}\right|}{1 * \log(0.1)} = 0.0217 \left|c^{(0)}\right|$$

La adaptación del SAA al NDP-M($\mathbf{v}$) teniendo en cuenta las reglas anteriores está recogido en la tabla 5.1.

Tabla 5.1: El algoritmo del recocido simulado aplicado al NDP-M

1. *(Inicialización):.* Encontrar una solución inicial $\mathbf{v}^{(0)}$. Sea $c^{(0)} = Z(\mathbf{v}^{(0)})$ su coste. Tomar $c_{\text{old}} = c^{(0)}$ y $\mathbf{v}_{\text{old}} = \mathbf{v}^{(0)}$. Definir $\mathcal{T}_0$, $\mathcal{T}_f$, y α. Definir K de acuerdo con la regla (c). Seleccionar los parámetros NAC y NRC como en (b). Inicializar la solución óptima $\mathbf{v}^* = \mathbf{v}^{(0)}$ y $c^* = c^{(0)}$. Inicializar los contadores cNAC=0 y cNRC=0. Definir $\mathbf{Q}^{(0)} = \psi I$ donde $\psi > 0$. Tomar $n = 0$, y $M = 0$.
2. Generar un vector aleatorio $\mathbf{u}$ y $\mathbf{v}_{\text{new}} = \max\{\mathbf{v}_{\text{LB}}, \mathbf{v}_{\text{old}} + \mathbf{Q}^{(n)}\mathbf{u}\}$.
3. Resolver el modelo de equilibrio TAP-M para el valor $\mathbf{v} = \mathbf{v}_{\text{new}}$. Sea $c_{\text{new}} = Z(\mathbf{v}_{\text{new}})$ el coste de la solución $\mathbf{v}_{\text{new}}$.
4. Si $c_{\text{new}} < c^*$ entonces $\mathbf{v}^* = \mathbf{v}_{\text{new}}$ y $c^* = c_{\text{new}}$.
5. Si $c_{\text{new}} < c_{\text{old}}$ entonces $\mathbf{v}_{\text{old}} = \mathbf{v}_{\text{new}}$, $c_{\text{old}} = c_{\text{new}}$, $cNAC = cNAC + 1$, $cNRC = 0$, $M = M + 1$ y $\mathbf{v}^{(M)} = \mathbf{v}_{\text{new}}$. En caso contrario, tomar $\mathbf{v}_{\text{old}} = \mathbf{v}_{\text{new}}$ con probabilidad $p = \exp -[(c_{\text{new}} - c_{\text{old}})/K\mathcal{T}]$ y $\mathbf{v}_{\text{old}} = \mathbf{v}_{\text{old}}$ con probabilidad $1 - p$ e incrementar el contador, $cNRC = cNRC + 1$.
6. Si $cNAC = NAC$ o $cNRC = NRC$ entonces disminuir la temperatura $\mathcal{T} = \alpha\mathcal{T}$, tomar $cNAC = 0$ y $cNRC = 0$. Actualizar $\mathbf{Q}$ empleando la factorización de Cholesky modificada; las iteraciones $\mathbf{v}^{(i)}$ para $i = 1, \ldots, M$; y las fórmulas (5.14)-(5.19). Tomar $n = n + 1$ y $M = 0$.
7. Si $\mathcal{T} < \mathcal{T}_f$ parar. En caso contrario volver al paso 2.

5.4 Experimentos numéricos

En esta sección hemos realizado varios experimentos numéricos para evaluar las principales contribuciones de este capítulo. Estos experimentos se han agrupado en tres bloques para contestar a las siguientes preguntas.

- ◇ ¿El SAA puede ser aplicado satisfactoriamente para resolver problemas de pequeño o incluso de mediano tamaño?
- ◇ ¿Cuál de las dos formulaciones (estándar y no-estándar) del NDP-M tiene mejor comportamiento computacional?
- ◇ ¿Cómo se puede emplear el NDP-M en las aplicaciones?

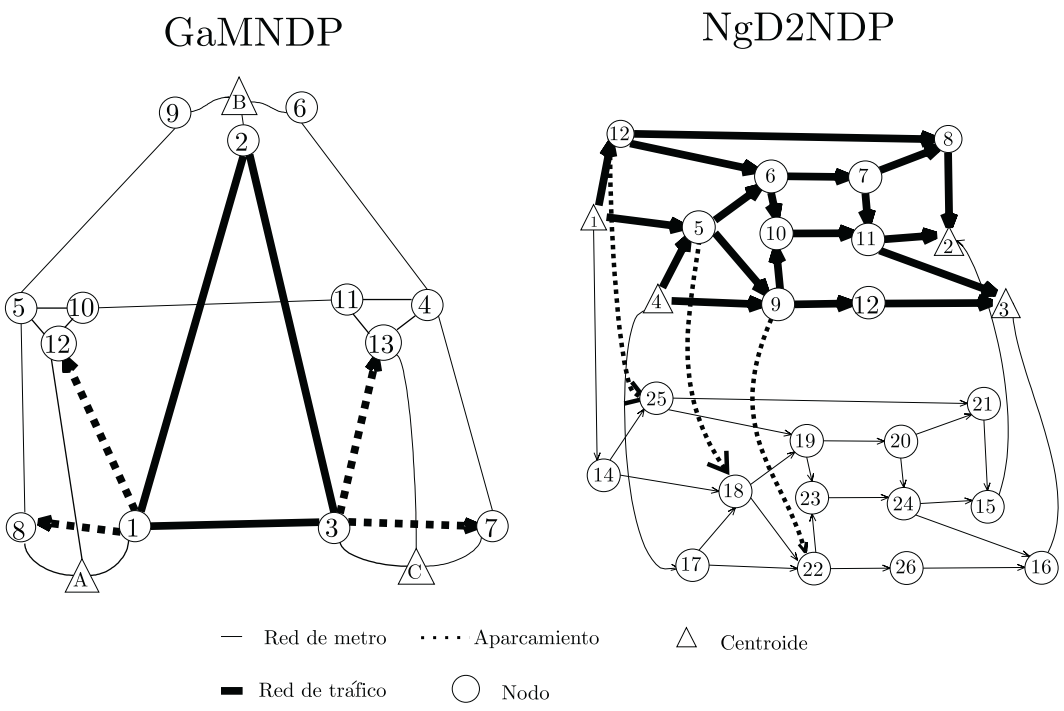


Figura 5.1: Grafo de las redes GaMNDP y NgD2NDP

Tabla 5.2: Tamaño de las redes de prueba

Problema	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{A}\rvert$	$\lvert T\rvert$	$\lvert W\rvert$	# Centroides	# demandas	# Total de variables
NgD2NDP	26	45	3	4	4	20	65
NgD2NDP1	26	45	3	4	4	20	65
GaMNDP	13	44	4	4	3	28	72
Hul2NDP	1002	1678	47	142	23	757	2435

En el experimento I validamos la adaptación del SAA para el NDP-M. En el experimento II hemos realizado una comparación entre la formulación estándar (NDP-M($\mathbf{x}$)) y no-stándar (NDP-M($\mathbf{y}$)). Este problema de diseño en redes multimodales se ha formulado para planificar las capacidades y tarifas de los aparcamientos empleados en los viajes combinados *park'n ride*. En el experimento III hemos ilustrado el uso del NDP-M para este objetivo.

En los experimentos numéricos hemos empleado cuatro redes de pruebas denominadas: NgD2NDP, NgD2NDP1, Hul2NDP y GaMNDP. Las tres primeras están definidas sobre redes de tráfico existentes (Nguyen y Dupuis [179] y Florian [80]) y la última ha sido desarrollada para este trabajo. Las redes NgD2NDP y NgD2NDP1 tienen la misma topología pero se diferencian en su parametrización. Su grafo junto el de la red GaMNDP se muestran en la figura 5.1.

El tamaño de las redes está recogido en la tabla 5.2. $\lvert\mathcal{N}\rvert$ es el número de nodos, $\lvert\mathcal{A}\rvert$ el número de arcos, $\lvert T\rvert$ el número de aparcamientos y $\lvert W\rvert$ el número de pares origen-destino.

En los experimentos numéricos, hemos considerado los modos coche (a), metro (b) y *park'n ride* (c). El número de componentes de la partición de la demanda es el número de componentes del vector $\mathbf{g} = (g^k_\omega, g^c_{\omega,t})$ donde $\omega \in W$, $k \in \{a, b, c\}$ y $t \in T$. El número de variables es la suma de variables de flujo en los arcos más el número de variables de demanda. Notar que el número de variables de diseño es $2\lvert T\rvert$ para el NDP-M($\mathbf{x}$) y $\lvert T\rvert$ para el NDP-M($\mathbf{y}$).

Tabla 5.3: Parámetros de la red GaMNDP

$$c_l(f_l) = A_l + B_l \left(\frac{f_l}{K_l}\right)^4,\ l \in \mathcal{A} - T$$

Arc l	A_l	B_l	K_l	Arc l	A_l	B_l	K_l
(2,1)	51.00	3.33	4.50	(1,2)	58.5	3.33	4.50
(3,2)	58.90	3.33	4.50	(2,3)	56.70	3.33	4.50
(1,3)	65.80	3.33	4.50	(3,1)	60.60	3.33	4.50
(5,8)	16.50	.00	1.00	(8,5)	16.50	.00	1.00
(9,5)	66.30	.00	1.00	(5,9)	66.30	.00	1.00
(4,6)	27.30	.00	1.00	(6,4)	27.30	.00	1.00
(7,4)	14.50	.00	1.00	(4,7)	14.50	.00	1.00
(10,11)	50.10	.00	1.00	(11,10)	50.10	.00	1.00
(7,3)	24.80	.00	1.00	(3,7)	24.80	.00	1.00
(3,13)	64.70	.00	1.00	(13,3)	64.70	.00	1.00
(2,6)	34.80	.00	1.00	(6,2)	34.80	.00	1.00
(2,9)	14.90	.00	1.00	(9,2)	14.90	.00	1.00
(1,12)	47.30	.00	1.00	(12,1)	47.30	.00	1.00
(1,8)	24.80	.00	1.00	(8,1)	24.80	.00	1.00
(10,12)	7.50	.00	1.00	(12,10)	15.00	.00	1.00
(13,11)	15.00	.00	1.00	(11,13)	7.50	.00	1.00
(10,5)	14.25	.00	1.00	(5,10)	16.70	.00	1.00
(12,5)	7.50	.00	1.00	(5,12)	5.00	.00	1.00
(13,4)	7.50	.00	1.00	(4,13)	5.00	.00	1.00
(11,4)	14.20	.00	1.00	(4,11)	16.70	.00	1.00

$$c_t(f_t, a_t, k_t) = A_t + a_t + B_t \left(\frac{f_t}{K_t + k_t}\right)^4,\ t \in T$$

$$Z(\mathbf{x}) = \theta \sum_{l \in \hat{\mathcal{A}}} c_l(\mathbf{f}^*, \mathbf{x}) f_l^* + \sum_{t \in T} \left[\hat{S}_t k_t - \eta f_t^* a_t\right]$$

Arc t	A_t	B_t	K_t	$\hat{S}_t$	Arc t	A_t	B_t	K_t	$\hat{S}_t$
(1,12)	7.80	6.00	0.50	100.0	(1,8)	14.60	6.00	0.50	100.0
(3,7)	16.10	6.00	0.50	100.0	(3,13)	10.70	6.00	0.50	100.0

$\theta = 1.5$ $\eta = 2.5$

Modelo de demanda				
O-D par	Demanda de viajes	Tasa de ocupación	Modelo logit anidado	
(1, 2)	5.50	1.1	$\alpha^a = 2.0$	$\beta_1 = 0.01$
(1, 3)	6.00	1.1	$\alpha^b = 1.0$	$\beta_2 = 0.05$
(3, 1)	7.50	1.1	$\alpha^c = 3.0$	$\theta_a = 1.0$
(3, 2)	8.00	1.1	$\alpha_t^c = 1.0$	$\theta_b = 1.0$

Las tablas 5.3, 5.4 y 5.5 recogen las funciones empleadas para el coste de viaje en cada arco y de inversión, además de los parámetros del modelo logit anidado. Las tablas están asociadas a las redes GaMNDP, NgD2NDP y NgD2NDP1. Hemos empleado costes de inversión lineales, esto es $s_t(k_t) = \hat{S}_t k_t$, y hemos usado expresiones del tipo BPR ([185]) para modelar los costes de aparcamiento. En el apéndice I, se muestra la solución del submodelo $\Gamma(\mathbf{y})$ de forma cerrada. Hemos omitido los parámetros y el grafo de Hul2NDP para ahorrar espacio.

Los parámetros usados por el SAA en estos ejemplos, están recogidos en la tabla 5.6. Su obtención se ha realizado mediante un laborioso proceso de ajuste a las redes de prueba. El parámetro ψ es empleado en la inicialización de la matriz $\mathbf{Q}^{(0)}$. En la formulación estándar, las variables de diseño son la capacidad y las tarifas de los aparcamientos y éstas poseen diferentes unidades. Por este motivo hemos introducido un parámetro ψ por cada tipo de variable. El primer valor del parámetro ψ está asociado a los precios y el segundo a las capacidades.

La codificación del SAA se ha realizado en FORTRAN Visual Workbench y se han empleado doble precisión. Las pruebas numéricas se han realizado en un ordenador PC con 384 megabytes de memoria RAM a 400 MHz.

Tabla 5.4: Parámetros de la red NgD2NDP

$c_l(f_l) = A_l + B_l f_l^2,\ l \in \mathcal{A} - T$

Arc l	A_l	B_l	Arc l	A_l	B_l
(1, 5)	7.0	.3250E-04	(1,12)	9.0	.2000E-04
(4, 5)	9.0	.2000E-04	(4, 9)	12.0	.1300E-03
(5, 6)	3.0	.3750E-02	(5, 9)	9.0	.3750E-02
(6, 7)	5.0	.6250E-02	(6,10)	13.0	.2500E-02
(7, 8)	5.0	.6250E-02	(7,11)	9.0	.6250E-02
(8, 2)	9.0	.6250E-02	(9,10)	10.0	.2500E-02
(9,13)	9.0	.2500E-02	(10,11)	6.0	.1250E-02
(11, 2)	9.0	.2500E-02	(11, 3)	8.0	.5000E-02
(12, 6)	7.0	.1250E-02	(12, 8)	14.0	.5000E-02
(13, 3)	11.0	.5000E-02	(14,18)	7.0	.6250E-02
(14,25)	9.0	.5000E-02	(17,18)	9.0	.5000E-02
(17,22)	12.0	.2500E-02	(18,19)	3.0	.3750E-02
(18,22)	9.0	.3750E-02	(19,20)	5.0	.6250E-02
(19,23)	13.0	.2500E-02	(20,21)	5.0	.6250E-02
(20,24)	9.0	.6250E-02	(21,15)	9.0	.6250E-02
(22,23)	10.0	.2500E-02	(22,26)	9.0	.2500E-02
(23,24)	6.0	.1250E-02	(24,15)	9.0	.2500E-02
(24,16)	8.0	.5000E-02	(25,19)	7.0	.1250E-02
(25,21)	14.0	.5000E-02	(26,16)	11.0	.5000E-02
(1,14)	.0	.0000E+00	(15, 2)	.0	.0000E+00
(16, 3)	.0	.0000E+00	(4,17)	.0	.0000E+00

$c_t(f_t, a_t, k_t) = A_t + a_t + B_t \left(\frac{f_t}{K_t + k_t}\right)^4,\ t \in T$

$Z(\mathbf{x}) = \theta \sum_{l \in \hat{A}} c_l(\mathbf{f}^*, \mathbf{x}) f_l^* + \sum_{t \in T} \left[\hat{S}_t k_t - \eta f_t^* a_t\right]$

Arc t	A_t	B_t	K_t	$\hat{S}_t$	Arc t	A_t	B_t	K_t	$\hat{S}_t$
(9,22)	10.0	5.0	100.0	10.0	(5,18)	9.0	5.0	100.0	10.0
(12,25)	11.0	5.0	100.0	10.0	$\theta = 1.5$			$\eta = 1.5$	

Modelo de demanda

O-D par	Demanda de viajes	Tasa de ocupación	Modelo logit anidado	
(1, 2)	800.0	1.0	$\alpha^a = 0.0$	$\beta_1 = 0.01$
(1, 3)	1600.0	1.0	$\alpha^b = 0.0$	$\beta_2 = 0.05$
(4, 2)	1200.0	1.0	$\alpha^c = 0.0$	$\theta_a = 0.05$
(4, 3)	400.0	1.0	$\alpha_t^c = 1.0$	$\theta_b = 0.03$

Tabla 5.5: Parámetros de la red NgD2NDP1

$c_l(f_l) = A_l f_l,\ l \in \mathcal{A} - T$							
Arc l	A_l	Arc l	A_l	Arc l	A_l	Arc l	A_l
(1, 5)	.1250E-01	(1,12)	.1000E-01	(4, 5)	.1000E-01	(4, 9)	.5000E-02
(5, 6)	.7500E-02	(5, 9)	.7500E-02	(6, 7)	.1250E-01	(6,10)	.5000E-02
(7, 8)	.1250E-01	(7,11)	.1250E-01	(8, 2)	.1250E-01	(9,10)	.5000E-02
(9,13)	.5000E-02	(10,11)	.2500E-02	(11, 2)	.5000E-02	(11, 3)	.1000E-01
(12, 6)	.2500E-02	(12, 8)	.1000E-01	(13, 3)	.1000E-01	(14,18)	.1250E-01
(14,25)	.1000E-01	(17,18)	.1000E-01	(17,22)	.5000E-02	(18,19)	.7500E-02
(18,22)	.7500E-02	(19,20)	.1250E-01	(19,23)	.5000E-02	(20,21)	.1250E-01
(20,24)	.1250E-01	(21,15)	.1250E-01	(22,23)	.5000E-02	(22,26)	.5000E-02
(23,24)	.2500E-02	(24,15)	.5000E-02	(24,16)	.1000E-01	(25,19)	.2500E-02
(25,21)	.1000E-01	(26,16)	.1000E-01	(1,14)	.0000E+00	(15, 2)	.0000E+00
(16, 3)	.0000E+00	(4,17)	.0000E+00				

$c_t(f_t, a_t, k_t) = A_t + a_t + B_t \left(\frac{f_t}{K_t + k_t}\right)^4, t \in T$

$Z(\mathbf{x}) = \theta \sum_{l \in \hat{\mathcal{A}}} c_l(\mathbf{f}^*, \mathbf{x}) f_l^* + \sum_{t \in T} \left[\hat{S}_t k_t - \eta f_t^* a_t\right]$

Arc t	A_t	B_t	K_t	$\hat{S}_t$	Arc t	A_t	B_t	K_t	$\hat{S}_t$
(9,22)	10.0	0.1	50	1.0	(5,18)	9.0	0.1	50	1.0
(12,25)	11.0	0.1	50	1.0	$\theta = 2.5$		$\eta = 2.5$		

Modelo de demanda				
O-D par	Demanda de viajes	Tasa de ocupación	Modelo logit anidado	
(1, 2)	800.0	1.0	$\alpha^a = 0.0$	$\beta_1 = 0.02$
(1, 3)	1600.0	1.0	$\alpha^b = 0.0$	$\beta_2 = 0.03$
(4, 2)	1200.0	1.0	$\alpha^c = 0.0$	$\theta_a = 1.00$
(4, 3)	400.0	1.0	$\alpha_t^c = 1.0$	$\theta_b = 1.00$

Tabla 5.6: Parámetros empleados por el SAA para los ejemplos de prueba

Parámetro	NgD2NDP	NgD2NDP1	GaMNDP	Hul2NDP
χ_s: fáctor de crecimiento en el NDP-M($\mathbf{x}$).	7.	0.5	2.0	2.0
χ_s: fáctor de crecimiento en el NDP-M($\mathbf{y}$).	7.	2.0	7.0	7.0
$\mathcal{T}_0$: temperatura inicial	1.0	1.0	1.0	1.0
$\mathcal{T}_f$: temperatura final	0.2	0.2	0.2	0.2
NAC: # configuraciones aceptadas	20	10	20	10
NRC: # configuraciones rechazadas	20	20	30	20
K: escala temperatura	0.001	0.002	0.001	0.002
α: parámetro de recocido	0.7	0.7	0.7	0.7
ψ inicialización de $\mathbf{Q}^0$ en el NDP-M($\mathbf{x}$).	5/30	2/50	10/0.05	1/25.
ψ inicialización de $\mathbf{Q}^0$ en el NDP-M($\mathbf{y}$).	10	5	5	1

5.4.1 Experimento I: validación del SAA

Existen dos cuestiones fundamentales que deben ser investigadas para validar el uso del SAA para la resolución del NDP-M:

- ◇ Convergencia a la solución óptima. Existe un resultado teórico que asegura la convergencia a la solución óptima con probabilidad uno, bajo la hipótesis de que la función objetivo $Z(\mathbf{v})$ se puede evaluar de forma exacta. En esta aplicación no es posible resolver de forma exacta el modelo de equilibrio TAP-M y por tanto la convergencia no está garantizada.
- ◇ Eficiencia computacional. El SAA genera una gran cantidad de soluciones candidatas y es por tanto obvio que el SAA sólo se puede aplicar a problemas de una importancia significicativa, donde el coste computacional está justificado. Para decidir si el SAA es un método eficiente o no deberíamos compararlo con algún otro. Esta comparación impone un número de problemas del nivel inferior que se deben resolver para alcanzar una determinada precisión. En esta situación, la aplicabilidad del SAA dependerá del número de problemas de equilibrio que se deben resolver y el tiempo empleado para resolver cada uno de ellos y éste dependerá de su tamaño y el nivel de precisión requerido en su resolución. En este experimento estamos interesados en la aplicabilidad del SAA y no en su eficiencia comparada a otros métodos.

El coste computacional está fuertemente influenciado por el algoritmo empleado para resolver el modelo de equilibrio TAP-M. Un esquema eficiente, empleado por Friesz y otros [89] cuando aplicaron el SAA al NDP para redes de tráfico, consta de dos etapas. En la primera etapa se emplea un algoritmo basado en la generación de caminos, por ejemplo el algoritmo de Frank-Wolfe, y en la segunda se utiliza el método de proyección de Bertsekas-Gafni en [21]. Este algoritmo opera directamente sobre los caminos obtenidos en la primera fase, resolviendo los problemas de equilibrio sobre el conjunto de caminos generados en la primera etapa. Este procedimiento simplifica cada problema de equilibrio a uno con restricciones de no negatividad en los caminos y proyecta en ellos la demanda. Esto hace muy económico calcular las soluciones candidatas, haciendo que el método SAA sea atractivo.

Este esquema puede ser fácilmente adaptado al NDP-M. En este trabajo solamente discutimos la primera fase del algoritmo. Hemos considerado la clase CG/SD desarrollada en los capítulos 2 y 3 como métodos a emplear en esta primera fase. Empleando la notación introducida en estos capítulos hemos empleado el algoritmo $\mathrm{FW}_{\infty}^{8,n_c}$.

El primer experimento evalúa el coste de congestión $\sum_{l\in\hat{\mathcal{A}}} c_l(\mathbf{f}^*,\mathbf{x})f_l^*$ para las variables de diseño $\mathbf{x}=\mathbf{0}$ con varias realizaciones del algoritmo CG/SD. Estos algoritmos están definidos por los valores del parámetro $n_c \in \{3,5,7,10\}$, que representan el número de iteraciones realizadas por el algoritmo de Frank-Wolfe para generar la columna en la fase CGP. El punto inicial de arranque de estos algoritmos se obtiene a partir de los costes en la red sin flujo.

En la figura 5.2 se ven los resultados obtenidos para la red de prueba Hul2NDP. La figura de la izquierda muestra la evolución del coste de congestión frente al tiempo de CPU empleado y la gráfica de la derecha muestra esta evolución frente al número total de iteraciones del algoritmo de Frank-Wolfe. Notar que en una iteración principal del algoritmo CG/SD se realizan n_c iteraciones del algoritmo de Frank-Wolfe.

La principal conclusión es que el algoritmo CG/SD tiene mejor comportamiento respecto al tiempo de CPU e indica que es recomendable incorporar este algoritmo en el SAA frente a un algoritmo de Frank-Wolfe. El tiempo de CPU empleado por el algoritmo CG/SD está comprendido en el intervalo $5-10$ segundos, dependiendo del punto inicial. Esto indicaría que el SAA podría ser aplicado sobre redes de tamaño moderado (Hul2NDP tiene 2.435 variables) debido a que tres o cuatro mil problemas de equilibrio podrían ser resueltos en un tiempo razonable.

El segundo experimento está diseñado para evaluar la convergencia del algoritmo SAA y su eficiencia en función del algoritmo CG/SD empleado en la resolución del TAP-M. Hemos comparado tres algoritmos. El primero está motivado por las conclusiones computacionales obtenidas por Friesz y otros [89] que indican que entre $3-5$ iteraciones del algoritmo de Frank-Wolfe generan soluciones bastantes precisas con un coste reducido. El segundo algoritmo incrementa a 15 el número de iteraciones

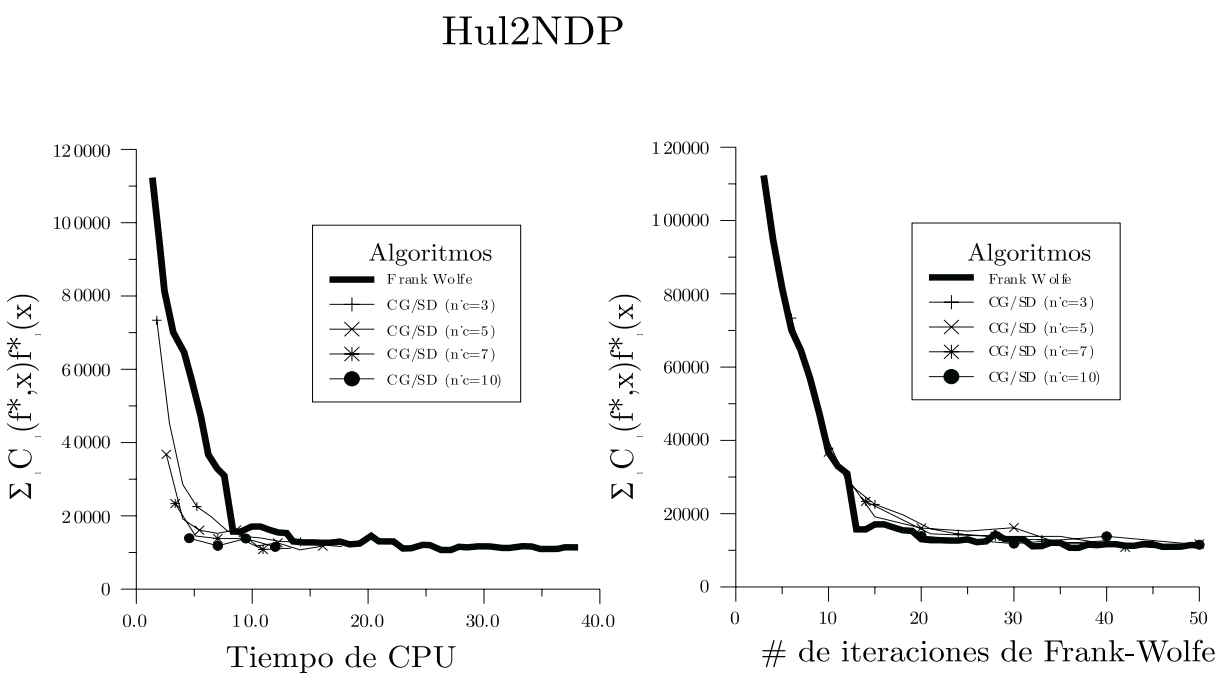


Figura 5.2: Evaluación del coste de congestión

empleadas por el algoritmo de Frank-Wolfe. El tercero es un algoritmo CG/SD con 5 iteraciones principales y el parámetro $n_c = 3$. Los dos últimos algoritmos generan soluciones mucho más precisas del TAP-M.

La figura 5.3 muestra los resultados obtenidos para la formulación estándar y la no-estándar. La formulación estándar, NDP-M($\mathbf{x}$), detecta un problema no acotado, que es una situación imposible. Este comportamiento se explica por la pérdida de la convergencia del SAA debido a la resolución no suficientemente precisa de los problemas del nivel inferior. Para ilustrar esta afirmación hemos recalculado la solución obtenida por cada uno de los algoritmos de forma muy exacta. Para este fin hemos empleado 30 iteraciones principales del algoritmo CG/SD utilizando un valor del parámetro $n_c = 10$. Los resultados son mostrados en la tabla 5.7. La primera columna, es el coste de inversión para extender la capacidad de aparcamiento, la segunda recoge los ingresos obtenidos de la tarifación de los aparcamientos, la tercera es el coste de transporte en el sistema (coste de congestión) y la última columna muestra el coste total. Observar que todas las soluciones obtenidas son superiores al valor de $10,000$. Esta evidencia muestra una gran discrepancia entre el valor aproximado calculado en la figura 5.2 y el valor exacto $Z(\mathbf{v})$.

¿Qué es lo que ha ocurrido? Lo que ha sucedido es que la convergencia del método se ha perdido. Este efecto se puede explicar del siguiente modo. Si la autoridad decide unas tarifas de aparcamiento elevadas entonces los usuarios cambiarán de aparcamiento o de modo de transporte. Es decir, los usuarios estarían incentivados a abandonar los viajes combinados *park'n ride* y utilizarían el metro o su vehículo privado para realizar su viaje. Si la precisión en la evaluación de la respuesta de los usuarios (resolución del TAP-M) es insuficiente entonces la solución aproximada del TAP-M no capta el efecto de que los usuarios dejan de utilizar los aparcamientos debido a sus tarifas elevadas. Por tanto, la solución del TAP-M contiene una sobreestimación de la demanda de aparcamiento, ya que realmente los usuarios dejan de utilizarlos. Por otro lado, las tarifas son muy elevadas y por tanto se sobrestima los ingresos mediante la tarifación de los aparcamientos, produciendo que el coste aproximado de esta configuración sea menor (aunque el coste real se mayor) que el de la solución actual y por tanto sea aceptada.

Una situación límite estaría en poner tarifas astronómicas, por ejemplo, igual al valor del coche. La respuesta del usuario haría que los aparcamientos no tendrían demanda. En la resolución del modelo de equilibrio, iteración tras iteración, los usuarios van abandonando el uso de los aparcamientos, pero

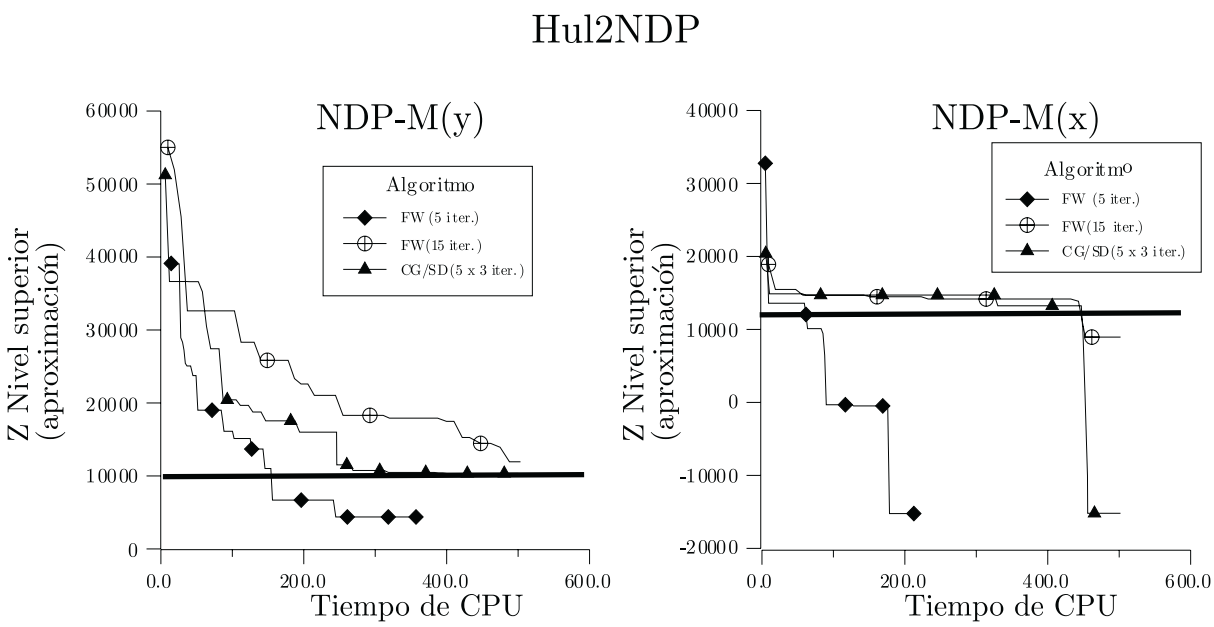


Figura 5.3: Eficacia del SAA frente al algoritmo empleado en la resolución del TAP-M

si interrumpimos demasiado pronto este proceso existiría una excesiva demanda que haría aceptar como óptimo esta política de tarifas. Este mismo efecto explica la apariencia no acotada del problema NDP-M($\mathbf{x}$).

Tabla 5.7: Evaluación "exacta" de las soluciones obtenidas

Algoritmo		Form. estándar.		
para el TAP-M	$\sum S_t(k_t^*)$	$-\sum \eta f_t^* a_t^*$	$\theta \sum c_l(\mathbf{f}^*, \mathbf{x}^*) f_l^*$	$Z(\mathbf{x}^*)$
FW, 5 iter.	10643.51	-772.80	11614.23	21484.94
FW, 15 iter	5957.49	-761.244	10485.66	15681.91
CG/SD, 5×3 iter	10359.10	-8246.04	10961.71	13074.71
Algoritmo		Form. no-estándar.		
para el TAP-M	$\sum S_t(k_t^*)$	$-\sum \eta f_t^* a_t^*$	$\theta \sum c_l(\mathbf{f}^*, \mathbf{y}^*)_l^*$	$Z(\mathbf{y}^*)$
FW, 5 iter.	6045.05	-17.02	11225.31	17253.34
FW, 15 iter	2101.84	-733.34	10393.86	11762.36
CG/SD, 5×3 iter	793.88	-748.24	10638.71	10684.36

Notar que la mejor solución encontrada se obtiene empleando el algoritmo CG/SD y la formulación no-estándar ($Z = 10684.36$). Este resultado es exactamente el opuesto al derivado del análisis de la figura 5.3.

Este ejemplo recomienda una gran precisión en la resolución del TAP-M a temperaturas elevadas. Esta es la principal motivación de considerar los algoritmos CG/SD como apropiados. Hemos empleado 15, 10 y 10 iteraciones principales del algoritmo CG/SD para valores $n_c = 6, 15$ y 6 para las redes NgD2NDP, NgD2NDP1 y GaMNDP respectivamente.

Para concluir el experimento I, realizamos unas consideraciones con el objetivo de explicar por qué es adecuado utilizar unas 5 iteraciones del algoritmo de Frank-Wolfe en el SAA aplicado al NDP y sin embargo no lo es para el NDP-M. Cuando el SAA progresa y las temperaturas son bajas, las soluciones candidatas son parecidas a la solución actual y por tanto el punto de partida para el algoritmo de Frank-Wolfe (flujos óptimos de la solución actual) está muy cerca de la verdadera solución, siendo suficiente realizar esas cinco iteraciones para obtener una gran exactitud de la solución candidata, es decir, en el límite, los problemas TAP son resueltos de forma exacta. Este efecto también es cierto

para el NDP-M, pero el hecho de que al principio la solución aproximada de la función objetivo (por resolverse aproximadamente el TAP-M) constituya una cota inferior del NDP-M, provoca la pérdida de la convergencia en la mayoría de los casos, haciendo que los problemas TAP-M no sean resueltos de forma exacta en el límite.

La diferencia sustancial del NDP con el NDP-M, que es la que produce la pérdida de la convergencia del SAA, es que los objetivos del nivel superior e inferior del NDP van a la par, esto es, que muchas[1] de las direcciones de descenso de una función objetivo también lo son para la otra función objetivo, sin embargo, esto no ocurre con el NDP-M, basta considerar direcciones donde se incremente las tarifas de los aparcamientos.

5.4.2 Experimento II: comparaciones entre las formulaciones estándar y no-estándar

El experimento II ha sido diseñado para comparar las formulaciones NDP-M($\mathbf{x}$) y NDP-M($\mathbf{y}$). Se ha empleado el mismo conjunto de parámetros para el SAA (ver la tabla 5.6) y el mismo algoritmo para resolver el TAP-M en las dos formulaciones .

Los experimentos computacionales realizados en el experimento I sugieren que la formulación no-estándar tiene mejores propiedades de convergencia. Esto puede ser debido a que la formulación no-estándar requiere un menor número de iteraciones que la formulación estándar para resolver satisfactóriamente el TAP-M. Esto puede ser debido a que la formulación no-estándar emplea costes lineales en los arcos asociados a los aparcamientos, reduciendo el grado de no linealidad del TAP-M. La precisión obtenida para estos problemas es mayor que la que se obtiene con la formulación estándar.

Los resultados obtenidos para cada formulación pueden ser explicados por las características de la propia formulación y/o por el comportamiento aleatorio del SAA, es decir, por la secuencia de números aleatorios empleados. Esto podría conducir a que los resultados obtenidos en el experimento I, donde la formulación no-estándar posee mejores propiedades de convergencia, pudieran ser debidos al azar y no a la propia formulación. Por ese motivo, se ha diseñado el primer experimento para decidir si las diferencias obtenidas entre las dos formulaciones son debidas al azar o a las características de cada formulación. En este experimento se realizan 10 pruebas para cada formulación, empleando diferentes semillas para la generación de la secuencia de números aleatorios. Con los resultados obtenidos se ha realizado el test no paramétrico *de rangos* ya que el tamaño muestral es pequeño (ver Coffin y Saltzman [54]). Los tests que han resultado significativos a un nivel de significación de $\alpha = 5\%$ son marcados con el símbolo †.

El primer test estadístico (ver el segundo bloque de la tabla 5.8) se realiza para decidir si el número total de iteraciones es igual para ambas formulaciones. El tercer bloque de la tabla 5.8 evalúa si el número de iteración en que alcanzan la mejor solución es menor en una formulación que en otra y el tercer test realizado (bloque cuarto) plantea si una formulación obtiene mejores soluciones que la otra. El valor mostrados en la tabla para los dos primeros tests es el número medio de iteraciones en las diez pruebas. Los resultados obtenidos indican que la formulación no-estándar es mejor que la estándar para los tres problemas de prueba. Para los dos primeros esta mejora es debido a que alcanza una mejor solución y en el tercer problema esta mejora es debida a que las soluciones, aunque no son mejores, son obtenidas con un número menor de iteraciones y por tanto con un coste computacional menor.

Las conclusiones anteriores no son suficientes para concluir que es preferible elegir la formulación no-estándar frente a la estándar en redes reales, ya que los experimentos son realizados sobre redes de pequeño tamaño. La complejidad computacional en redes reales puede provocar una situación en la que el SAA trabaje únicamente a elevadas "temperaturas". Por esta razón, es importante investigar el comportamiento del SAA al inicio del proceso. La figura 5.4 muestra la evolución de las dos formulaciones frente al número de iteraciones. Hemos dibujado la mejor de las diez pruebas para cada

[1] La paradoja de Braess garantiza que existen redes donde ciertas direcciones de descenso del nivel superior (asignación bajo el segundo principio de Wardrop) no son de descenso para el nivel inferior (asignación bajo el primer prinicipio de Wardrop).

Tabla 5.8: Comportamiento computacional de las dos formulaciones del NDP-M

Problema	# iter.			Iter. obtención. opt. sol.			$\bar{Z}$		
	Stan.	No-stán.	p-valor	Stán.	No-stán.	p-valor	Stán.	No-stán.	p-valor
NgD2NDP	270.0	255.4	0.918	174.7	169.4	0.683	516414.40	514759.80	0.004†
NgD2NDP1	167.1	189.0	0.153	91.1	117.4	0.308	64673.740	63759.01	0.032 †
GaMNDP	336.3	283.5	0.005†	207.9	148.2	0.010†	976.988	976.492	0.918

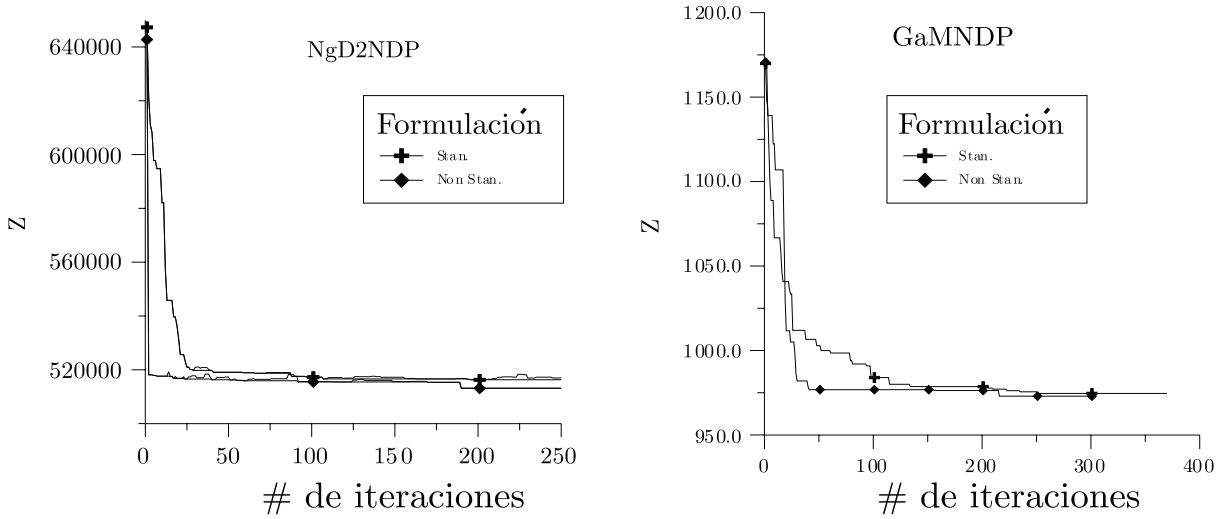


Figura 5.4: Evolución del SAA para las dos formulaciones del NDP-M

una de las formulaciones y se observa que también en las primeras iteraciones del SAA la formulación no-estándar posee mejor comportamiento.

5.4.3 Experimento III: utilización del NDP-M

La primera tarea para poder usar el NDP-M es calibrar el parámetro θ del NDP-M($\mathbf{v}$). Para calibrar θ, usualmente es necesario resolver (5.1) con varios valores de θ, para obtener una solución factible al problema de diseño con restricciones presupuestarias. No obstante, el modo más habitual es asumir un valor fijo de θ que convierta el tiempo total en el sistema de transporte en costes monetarios.

Para ilustrar la dependencia de la solución del NDP-M respecto a este parámetro θ se ha realizado el primer experimento. Hemos resuelto la red de pruebas NgN2NDP para diferentes valores de θ. Los resultados son mostrados en la figura 5.5. Se observa que para valores pequeños de θ (gráfica de la izquierda de la figura 5.5) el incremento de este parámetro (que representa una mayor valoración del el efecto de la congestión) conduce a un incremento en la oferta de aparcamientos para reducir la congestión en la red de tráfico, debido a la incentivación de los viajes combinados *park'n ride*. Notar que un incremento de este parámetro para valores grandes de θ (gráfica de la derecha de la figura 5.5) no afecta a la solución. Esta situación es equivalente a resolver el NDP-M($\mathbf{x}$) con un presupuesto elevado, que hace que las restricciones presupuestarias sean inactivas, y por tanto, un incremento del presupuesto, recogida a través del parámetro θ, no afecta a la solución.

El próximo experimento está diseñado para ilustrar las posibilidades de aplicación del NDP-M. Sobre el modelo base NDP-M se puede añadir un conjunto de restricciones para las variables de diseño de la forma $\mathbf{x} \in X$. La adaptación del SAA a este nuevo problema se realiza proyectando la solución candidata dentro del conjunto factible de soluciones X. Este conjunto tiene en cuenta ciertas restricciones para el conjunto de las políticas del planificador. A modo de ejemplo ilustrativo, hemos considerado cuatro posibles elecciones de este conjunto.

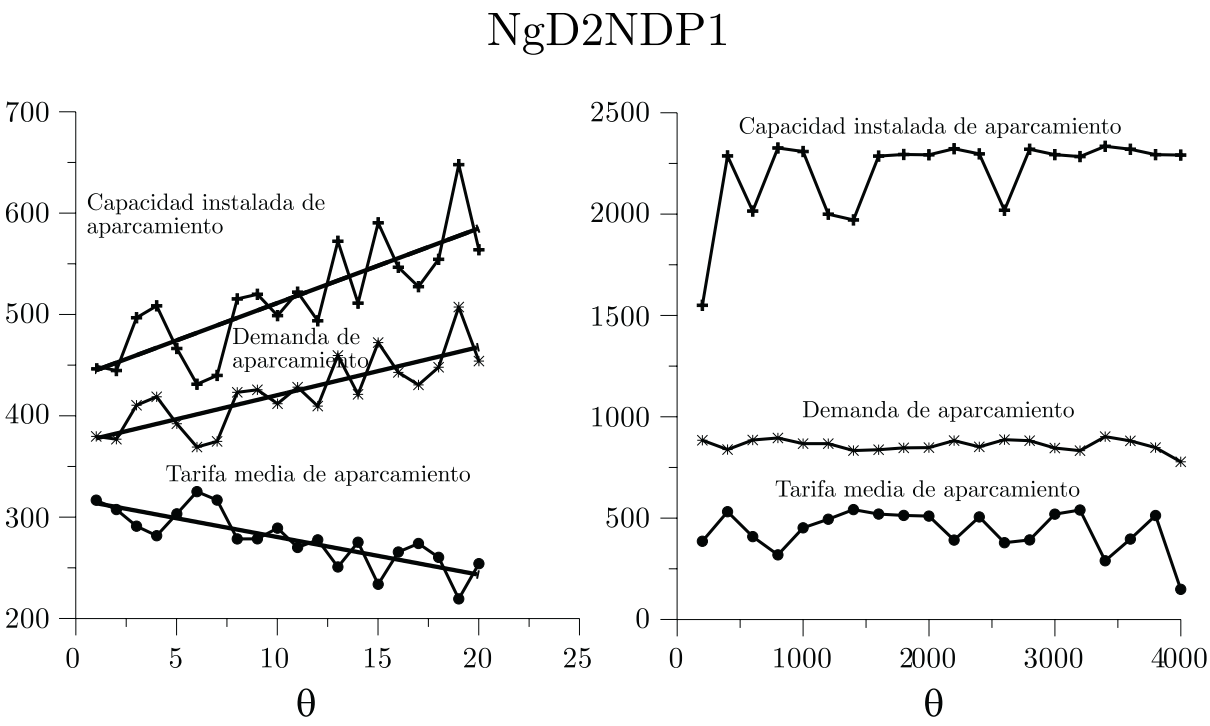


Figura 5.5: El papel del parámetro θ

- ◇ Política I. El objetivo es hacer un plan de ampliación de capacidades de aparcamientos y posible tarifación del sistema de aparcamientos. Esta situación se recoge mediante el conjunto $X = \{(a_t, k_t) \,/\, a_t \geq 0 \text{ y } k_t \geq 0\}$. Esta es la situación original formulada en el NDP-M.
- ◇ Política II. El objetivo del planificador es realizar un plan de tarifas de los aparcamientos. Asumimos que no es posible cambiar la capacidad ofertada de aparcamientos. Esta política es modelada mediante el conjunto de factibilidad $X = \{(a_t, k_t) \,/\, a_t \geq 0 \text{ y } k_t = 0\}$
- ◇ Política III. El objetivo es realizar un plan de ampliación de la capacidad de los aparcamientos. Aquí tenemos que $X = \{(a_t, k_t) \,/\, a_t = 0 \text{ y } k_t \geq 0\}$
- ◇ Política IV. El planificador desea tarifar los aparcamientos sin modificar su capacidad pero exigiendo que el precio de todos ellos sea el mismo, esta situación conduce al conjunto de factibilidad $X = \{(a_t, k_t) \,/\, a_t = a \text{ y } k_t = 0\}$

Se han considerado dos escenarios futuros definidos por dos matrices O-D. En el primer escenario la matriz O-D coincide con la matriz empleada en los experimentos anteriores y se muestra en la tabla 5.3. La matriz O-D para el segundo escenario es el doble de la matriz del primer escenario. Los resultados de las cuatro políticas anteriores aplicadas a los dos escenarios futuros se muestran en la tabla 5.9.

La política I coincide con el modelo original NDP-M, se observa que sólo en algunos aparcamientos es conveniente incrementar la capacidad de aparcamiento. Las políticas II, III y IV están asociadas solamente a una variable de diseño.

La ampliación de la capacidad de aparcamiento puede ser vista como un incentivo para la promoción de los viajes combinados. Por otro lado, las tarifas pueden ser consideradas como un desincentivo del uso de los aparcamientos disuasorios. Ambas opciones están interrelacionados en el proceso de planificación. Por ejemplo, cuando el plan se realiza empleando las dos variables simultáneamente (política I) el aparcamiento 8 es altamente tarifado, haciendo que los usuarios elijan otro aparcamiento alternativo o cambien de modo de transporte. Unos cuantos de estos usuarios serán dirigidos a través del aparcamiento 12. Si el planificador sólo puede incrementar la capacidad de aparcamientos (política

Tabla 5.9: Ilustración del uso del NDP-M sobre la red GaMNDP

Escenario	Aparcamiento	Política I			Política II		Política III		Política IV	
	t	a_t	k_t	Demanda$^{(*)}$	a_t	Demanda	k_t	Demanda	a	Demanda
	8	136.414	.000	.019	221.185	.002	.011	.464	59.248	.365
g_ω	12	2.762	.354	.553	34.156	.367	.000	.308	59.248	.172
	13	63.284	.000	.284	65.991	.311	.278	.492	59.248	.261
	7	74.295	.003	.347	95.574	.294	.009	.451	59.248	.376
	8	50.271	.728	.902	107.141	.413	1.012	1.222	99.776	.434
	12	52.017	.000	.310	97.100	.319	.288	.519	99.776	.307
$2 * g_\omega$	13	105.735	.000	.107	113.877	.366	.870	.925	99.776	.391
	7	51.391	1.651	1.483	117.701	.441	1.247	1.431	99.776	.471

$^{(*)}$ Número de usuarios (demanda) expresada en unidades de millar.

III) entonces no es posible desincentivar el uso del aparcamiento 8 y por tanto no irá al aparcamiento 12 un flujo adicional que haga recomendable ampliar su capacidad, como ocurría en la política I.

En la política IV se decide la misma tarifa para todos los aparcamientos mediante un equilibrio entre oferta-demanda. En el escenario II la demanda total se duplica, lo que hace que la capacidad de aparcamientos (ofertada) sea insuficiente y eleve por tanto los precios (tarifas) de este bien. El NDP-M permite decidir cual es la tarifa óptima en este equilibrio. Este modelo permite tener en cuenta la no linelidad del precio del aparcamiento en relación a la ley de oferta y demanda. A una duplicación de la demanda total no le corresponde la duplicación del precio del aparcamiento.

Apéndice I: cálculo de $\Gamma(\mathbf{y})$ para el caso de costes de inversión lineales y funciones del tipo BPR para representar el coste de aparcamiento

En este apéndice analizaremos un caso particular donde la función $\mathbf{x}^*(\mathbf{y})$ puede ser calculada explícitamente. Supondremos que el coste de aparcamiento está representados por una función del tipo BPR

$$y_t = c_t(f_t^*, a_t, k_t) = a_t + A_t + B_t \left(\frac{f_t^*}{k_t + K_t} \right)^{N_t}, \tag{5.20}$$

donde A_t es un coste fijo que representa el tiempo necesario para acceder a la parada de transporte público, a_t es la tarifa de aparcamiento y el término $B_t \left(\frac{f_t^*}{k_t+K_t} \right)^{N_t}$ es un coste variable que considera el tiempo de búsqueda de aparcamiento y el tiempo de salir del aparcamiento. El coste de inversión es lineal y viene dado

$$s_t(k_t) = \hat{S}_t k_t, \tag{5.21}$$

donde $\hat{S}_t$ es una constante.

La función c_t es estrictamente creciente en a_t y decreciente en k_t, el gradiente de la única restricción no es cero y este problema satisface la restricción de cualificación de Slater (ver Bazaraa y otros [12]) y por tanto, la solución debe satisfacer la condición de Karush-Kuhn-Tucker. Obteniendo estas condiciones para las variables k_t y a_t y eliminando el multiplicador de Lagrange de la restricción de igualdad, obtenemos

$$\begin{aligned} s_t'(k_t) + \frac{\eta f_t^* - \mu_{a_t}}{\partial c_t / \partial a_t} \frac{\partial c_t}{\partial k_t} + \mu_{k_t} &= 0, \\ \mu_{a_t} a_t &= 0, \\ \mu_{k_t} k_t &= 0, \\ \mu_{a_t}, \mu_{k_t} &\leq 0, \\ c_t(f_t^*, a_t, k_t) &= y_t, \\ a_t &\geq 0, \\ k_t &\geq 0. \end{aligned} \tag{5.22}$$

donde los flujos en los arcos han sido calculados para un valor fijo de la variable $\mathbf{y}^*$, esto es, $\mathbf{f}^* = \mathbf{f}^*(\mathbf{y})$.

Notar que bajo la hipótesis de existencia de soluciones para cada valor de $\mathbf{x}_t$ y asumiendo que el sistema de inecuaciones (5.22) tiene únicamente una solución, entonces ésta es la solución óptima. Si el sistema se pude resolver explícitamente entonces podemos calcular $\mathbf{x}^*(\mathbf{y})$ y $\Gamma(\mathbf{y})$.

Las condiciones de KKT (5.22) para el caso de este apéndice son

$$\begin{aligned} \hat{S}_t + [\eta f_t^* - \mu_{a_t}](-N_t) B_t \frac{{f_t^*}^{N_t}}{(k_t + K_t)^{N_t+1}} + \mu_{k_t} &= 0, \\ \mu_{a_t} a_t &= 0, \\ \mu_{k_t} k_t &= 0, \\ \mu_{a_t},\ \mu_{k_t} &\leq 0. \end{aligned} \tag{5.23}$$

Distinguiremos cuatros casos para poder resolver el sistema de desigualdades (5.23).

◇ Caso I: $k_t > 0$ y $a_t > 0$. En este caso el sistema de ecuaciones (5.23) se formula

$$\begin{aligned} \hat{S}_t - \eta N_t B_t \left(\frac{f_t^*}{k_t + K_t} \right)^{N_t+1} &= 0, \\ \mu_{a_t} = \mu_{k_t} &= 0, \end{aligned}$$

eliminando k_t, obtenemos

$$k_t = \sqrt[N_t+1]{\frac{\eta N_t B_t}{\hat{S}_t}} f_t^* - K_t > 0, \tag{5.24}$$

sustituyendo (5.24) en la expresión BPR y eliminando a_t, obtenemos

$$a_t = y_t - A_t - B_t \left(\frac{\hat{S}_t}{\eta N_t B_t} \right)^{\frac{N_t}{N_t+1}} > 0. \tag{5.25}$$

◇ Caso II: $k_t = 0$ y $a_t > 0$. Empleando la expresión BPR (5.20), obtenemos

$$a_t = y_t - A_t - B_t \left(\frac{f_t^*}{K_t} \right)^{N_t} > 0. \tag{5.26}$$

En este caso las condiciones de KTT son

$$\begin{aligned} \hat{S}_t + [\eta f_t^* - \mu_{a_t}](-N_t) B_t \frac{{f_t^*}^{N_t}}{K_t^{N_t+1}} + \mu_{k_t} &= 0, \\ \mu_{a_t} &= 0, \\ \mu_{k_t} &\leq 0, \end{aligned} \tag{5.27}$$

eliminando de (5.27) μ_{k_t} obtenemos

$$\mu_{k_t} = \eta N_t B_t \left(\frac{f_t^*}{K_t} \right)^{N_t+1} - \hat{S}_t \leq 0. \tag{5.28}$$

◇ Caso III: $k_t > 0$ y $a_t = 0$. Empleando la fórmula BPR (5.20), obtenemos

$$y_t = A_t + B_t \left(\frac{f_t^*}{k_t + K_t} \right)^{N_t},$$

y obtenemos

$$k_t = \sqrt[N_t]{\frac{B_t}{y_t - A_t}} f_t^* - K_t > 0.$$

En este caso la condición de KKT (5.22) llega a ser

$$\begin{aligned} \hat{S}_t + [\eta f_t^* - \mu_{a_t}](-N_t) B_t \frac{{f_t^*}^{N_t}}{(K_t + k_t)^{N_t+1}} + \mu_{k_t} &= 0, \\ \mu_{a_t} &\geq 0, \\ \mu_{k_t} &= 0, \end{aligned} \tag{5.29}$$

eliminado de (5.29) μ_{a_t}, obtenemos

$$\mu_{a_t} = \left[\eta - \frac{\hat{S}_t B_t^{1/N_t}}{N_t (y_t - A_t)^{(N_t+1)/N_t}} \right] f_t^* \leq 0. \tag{5.30}$$

◇ Caso IV: $k_t = 0$ y $a_t = 0$. Empleando la fórmula BPR (5.20), obtenemos

$$y_t = A_t + B_t \left(\frac{f_t^*}{K_t} \right)^{N_t},$$

y obtenemos

$$y_t - A_t = B_t \left(\frac{f_t^*}{K_t} \right)^{N_t}. \tag{5.31}$$

En este caso las condiciones de KKT (5.22) llegan a ser

$$\begin{aligned} \hat{S}_t + [\eta f_t^* - \mu_{a_t}](-N_t) B_t \frac{{f_t^*}^{N_t}}{(K_t)^{N_t+1}} + \mu_{k_t} &= 0, \\ \mu_{a_t} &\geq 0, \\ \mu_{k_t} &\geq 0. \end{aligned} \tag{5.32}$$

y sustituyeno (5.31) en (5.32) obtenemos

$$\hat{S}_t + [\eta f_t^* - \mu_{a_t}](-N_t) \frac{y_t - A_t}{K_t} + \mu_{k_t} = 0. \tag{5.33}$$

En este caso los multiplicadores (si existen) no son únicos. La condición para que la ecuación (5.34) tenga solución con el multiplicador $\mu_{k_t} \leq 0$ es que la siguiente desigualdad se cumpla

$$\hat{S}_t + [\eta f_t^* - \mu_{a_t}](-N_t) \frac{y_t - A_t}{K_t} \geq 0, \tag{5.34}$$

y eliminando μ_{a_t}

$$\mu_{a_t} \geq \eta f_t^* - \frac{\hat{S}_t K_t}{N_t (y_t - A_t)}.$$

Por otro lado el multiplicador μ_{a_t} debe ser no positivo, obteniendo la relación

$$0 \geq \mu_{a_t} \geq \eta f_t^* - \frac{\hat{S}_t K_t}{N_t (y_t - A_t)}. \tag{5.35}$$

Si la relación (5.35) se satisface, entonces el punto $k_t = a_t = 0$ tiene un punto de KKT asociado.

Capítulo 6

Diseño de intercambiadores multimodales urbanos

Resumen

En este capítulo se aborda el diseño de intercambiadores multimodales urbanos en un contexto de planificación estratégica.

La red de transporte urbano considerada está formada por una red principal de metro-cercanías y una red secundaria que facilita el acceso a la red principal. Se han tratado los siguientes aspectos del problema de diseño de los intercambiadores:

- ◇ Determinar la localización de los intercambiadores en la red principal.
- ◇ Diseño de la red secundaria.
- ◇ Dimensiones y tarifas de los aparcamientos disuasorios de los intercambiadores.

El problema se ha formulado mediante un modelo de programación matemática binivel. En el nivel superior se decide el diseño de la red de transporte y en el nivel inferior los usuarios eligen su patrón de viaje sobre la red diseñada, produciendo la partición de la demanda total por modos, por intercambiadores y por tipos de aparcamiento. El nivel inferior está definido por un modelo de equilibrio con modos combinados del que se ha presentado dos formulaciones: una del tipo *punto fijo* y la otra mediante un problema de optimización. En el apéndice de este capítulo se muestra la equivalencia entre ambas formulaciones. Para la formulación de tipo punto fijo, se desarrolla un algoritmo de Gauss-Seidel para su resolución.

El problema de optimización binivel es mixto (con variables enteras y continuas) y no lineal. Se han considerado varios algoritmos heurísticos para resolverlo. Los dos primeros se basan en técnicas "greedy", el tercero emplea una técnica de intercambio y el último es un algoritmo de recocido simulado para problemas discretos. Se han presentado una comparativa de estos métodos sobre un conjunto de redes de transporte generadas aleatoriamente y se ha ilustrado esta metodología de diseño sobre una red de prueba.

Palabras clave: Diseño de intercambiadores multimodales urbanos. Programación matemática binivel. Localización. Algoritmos golosos. Algoritmos de intercambio. Algoritmos de recocido simulado.

6.1 Introducción

En muchas de las grandes ciudades europeas, se están desarrollando políticas que incentivan el transporte público mediante el desarrollo de facilidades atractivas de transferencia entre redes de transporte. Es por eso, que la Unión Europea está potenciando el estudio en estos campos, dentro de sus programas marcos de investigación. En particular, el grupo Euritrans (Hansen y otros [117]) presentó en el IV Programa Marco y dentro de la tarea 5.3: "Transiciones en transporte multimodal" una *metodología macro* para estudiar el diseño de intercambiadores. Estas áreas de investigación son también preferentes en España, donde se le está dando soporte financiero mediante entidades como el Ministerio de Educación y Cultura o la Comunidad de Madrid.

En el momento actual, la movilidad continúa creciendo tanto en el número de viajes como en las distancias que se recorren. Esto provoca un incremento del tiempo de viaje y un aumento de la congestión. Para reducir el uso de los vehículos privados, se están diseñando sistemas de transporte público veloces y altamente interconectados mediante los denominados intercambiadores multimodales urbanos, donde es posible cambiar de modo de transporte. Esta red principal de transporte público (formada, por ejemplo, de la red de cercanías en unión con el metro) se alimenta mediante otras formas de transporte, como por ejemplo: en autobús, en coche privado, en bicicleta, en taxi, andando, etc.

Este tipo de viajes, que emplean más de un modo de transporte, se denominan combinados y tienen tres partes: la primera componente está definida por el viaje del nodo origen al de transferencia y este trayecto puede ser realizado en coche, autobús, andando, bicicleta, etc.; la segunda componenete es la transferencia entre redes de transporte y ésta puede incluir el aparcamiento, el viaje andando a la parada de transporte público, el periodo de espera en la parada, etc; la última componente es el viaje en transporte público y este trayecto va desde la parada de transporte público hasta el nodo destino.

En el proceso de desarrollo y evaluación de facilidades de intercambio (como es el diseño de intercambiadores multimodales urbanos) es necesario definir herramientas de planificación que auxilien la toma de decisiones. Se pueden considerar dos puntos de vista: uno *macroscópico* y otro *microscópico*.

La metodología microscópica analiza los intercambiadores considerando que el flujo de pasajeros en ellos es independiente del resto de la red de transporte y de las políticas de gestión de tráfico. Este valor se calcula mediante encuestas. El objetivo de estos estudios es definir unos estándares de calidad en la operación de transferencia entre redes, concentrándose fundamentalmente en aspectos operacionales como son: las conexiones, la seguridad, las facilidades para la espera, la información dinámica, la reducción de distancias andando, los métodos de venta de billetes, la descripción de rutas, el diseño de paneles informativos y horarios, etc.

La metodología macroscópica analiza las relaciones entre el diseño del intercambiador y la asignación del tráfico multimodal. En una metodología macro, se analizan aspectos como las capacidades de transferencia, tiempos de transferencia, capacidades y tarifas de los aparcamientos en un contexto de rutas multimodales, caracterización de la demanda, costes generalizados en la red de transporte y otras características macroscópicas del sistema de transporte.

La metodología macroscópica puede ser diferenciada en función del horizonte temporal empleado en la planificación. A largo plazo, o planificación estratégica, se decide el número de intercambiadores, su localización, así como otras características topológicas de la red multimodal. Los estudios realizados consideran simplificadamente la red de tráfico, por ejemplo, emplean representaciones de la red mediante grafos en forma de estrella, árboles, etc.

En la planificación a medio plazo, o planificación táctica, la topología de la red se considera fija. Los estudios deciden la distribución de recursos teniendo en cuenta el problema de asignación de la demanda a la red. En el capítulo 1 se ha ilustrado como el modelo TAP-M sirve para evaluar la demanda en los intercambiadores en función de ciertas características macroscópicas del sistema de transporte tales como: la capacidad de los intercambiadores, las distancias medias andando y el nivel de congestión en la red de tráfico y de transporte público.

La planificación a corto plazo, o planificación operacional, no se considera en la metodología macroscópica del diseño de intercambiadores.

Oppenheim [187] describe el proceso de diseño de redes como un problema de programación matemática binivel, en el que el nivel superior diseña la oferta de servicios de transporte y el nivel inferior define la demanda de estos servicios. El problema de diseño de intercambiadores, tanto en un contexto de planificación estratégica como táctica, es un ejemplo de problema de diseño de redes.

Los problemas de diseños de redes se clasifican en continuos (CNDP) o en discretos (DNDP). Esta taxonomía obedece a la naturaleza continua o discreta de las variables de diseño. El CNDP se concentra en problemas de parametrización de la red mientras que el DNDP en el diseño de su topología. El problema presentado en este capítulo es un problema mixto. Por un lado recoge la tarifación y capacidad de los aparcamientos, que es un problema continuo, y por otro lado aborda la localización y tipo de alimentación de los intercambiadores, que es un problema discreto.

Leblanc [147] realizó la primera aproximación al problema DNDP. Poorzahedy y Turnquist [202] presentaron un algoritmo heurístico para resolver el problema DNDP que es no lineal y entero. Boyce y Janson [31], Chen y Alfa [46] presentaron variaciones a este modelo DNDP así como nuevas propuestas para su resolución.

El problema de distribuir una flota de autobuses en una red de transporte público es otro ejemplo de DNDP. Han y Wilson [116] y Leblanc [148] formularon este problema mediante un problema binivel.

El CNDP elige las capacidades de la red de modo que se minimice el tiempo total de transporte en la misma. Abdulaal y LeBlanc [4] formularon este CNDP como un problema de programación matemática binivel donde el nivel inferior está definido por un problema de asignación en equilibrio y lo resolvieron mediante el algoritmo de Hook-Jeeves. Yang y Bell [243] realizaron una exhaustiva revisión del problema de diseño de redes de tráfico y de los algoritmos empleados en su resolución.

El diseño de intercambiadores no se ha estudiado en la literatura bajo la óptica de un problema de diseño de redes. Las aproximaciones a este problema se han dado bajo la perspectiva de planificación de aparcamientos.

Uno de los primeros artículos en esta dirección se debe a Florian y Los [84] que evalúan la demanda de aparcamientos disuasorios en función de los costes de transporte para acceder a ellos. Más recientemente, Carrese y otros [39] abordan el problema de localización de aparcamientos considerando dos niveles de decisión: por un lado, el planificador determina la localización de los aparcamientos y por otro los usuarios, representados por un modelo de asignación, eligen su patrón de viaje. El planificador genera un conjunto de políticas para satisfacer la demanda de aparcamientos y, mediante el modelo de asignación, evalúa la reacción de los usuarios a cada una de ellas. Esta evaluación se realiza para diferentes escenarios futuros y el planificador elige *la mejor* de las políticas. Coppola [56] considera una metodología similar a la de [39] pero empleando un modelo diferente de asignación para poder recoger la elasticidad en la demanda de aparcamiento.

Con relación a la localización de aparcamientos, dos referencias recientes son: Nickel y otros [182], que presentan un modelo basado en diseño (de un solo nivel) de redes, y Mesa y Ortega [171], que consideran la localización de estaciones en redes de metro teniendo en cuenta los viajes de tipo *park'n ride*. Estos autores establecen el área de captación de estas estaciones comparando el tiempo de viaje en vehículo privado con el coste de viaje en modo combinado.

Los problemas de programación matemática binivel son muy difíciles de resolver debido a su inherente no convexidad y no diferenciabilidad. Para estos problemas es adecuado emplear los *métodos de búsqueda probabilística*, tales como el recocido simulado (SAA), cuando se buscan óptimos globales.

Anandalingam y otros [7] usan el SAA para resolver la programación lineal binivel. Friesz y otros [89] aplican el SAA al problema continuo de diseño de redes. En este capítulo, se emplea el SAA, un algoritmo de intercambio y los algoritmos golosos para un problema discreto de diseño de redes.

La metodología desarrollada en este capítulo difiere de la presentada en el capítulo 1 en que las políticas son generadas automáticamente por el modelo (binivel), mientras que en el capítulo 1 son obtenidas por el planificador de la red, siendo posteriormente evaluadas por el modelo de asignación TAP-M. Ambas metodologías tienen en cuenta el flujo en los intercambiadores en función de su diseño. En el capítulo 5, se ha planteado un modelo binivel para abordar la tarifación y capacidad de los aparcamientos disuasorios de estos intercambiadores, situando este problema en un contexto de

planificación táctica. El modelo desarrollado en este capítulo incorpora aspectos de la planificación estratégica como son la localización y el diseño de la red de alimentación de los intercambiadores, para lo cual se perderá realismo en la representación de la congestión en la red.

Este capítulo se organiza del siguiente modo: en primer lugar se introduce el problema de diseño de redes, después se desarrolla un modelo de equilibrio con modos combinados que definirá el nivel inferior del problema de diseño binivel. El nivel superior está definido por el problema de diseño de los intercambiadores. En la siguiente sección desarrollamos cuatro algoritmos heurísticos para la resolución del modelo binivel. Posteriormente realizaremos una comparativa entre los métodos empleados y finalizaremos el capítulo con una ilustración del uso del modelo sobre una red de pruebas. En el apéndice demostramos la equivalencia entre las condiciones de equilibrio y la formulación mediante la programación matemática.

6.2 El problema de diseño de intercambiadores multimodales urbanos

Asumiremos que se gestiona una red de transporte público jerárquica, formada por una red principal y otra secundaria (ver figura 6.1). La red principal de transporte está formada por líneas que permiten realizar viajes urbanos de larga distancia. Por otro lado, la red secundaria de transporte público permite viajes de acceso a la red principal. La red principal puede ser, por ejemplo, una red de cercanías en unión con la red de metro, mientras que la red secundaria puede estar definida por una red local de autobuses.

Abordaremos el problema de localizar nuevas estaciones en la red principal. Las estaciones con facilidades especiales, como la disponibilidad de aparcamientos, se denominan intercambiadores multimodales urbanos. Si se instala una nueva estación, entonces varias líneas de transporte público podrían efectuar parada en ella. Supondremos que se dispone de un procedimiento para definir qué líneas de transporte público deben parar en la nueva estación y para determinar su frecuencia, permitiéndonos calcular los nuevos tiempos de viajes sobre las nuevas infraestructuras.

Hemos considerado los tres siguientes aspectos del problema:

- ◇ La localización de los intercambiadores en la red principal de transporte público.
- ◇ El diseño de la red de acceso a los intercambiadores. Concretando, no se aborda la definición de la red de servicios decidiendo rutas, frecuencias, etc. sino la evaluación de un diseño específico de la red de acceso en el contexto general del sistema de transporte.
- ◇ El diseño de facilidades de aparcamiento en estos intercambiadores, en concreto, hemos considerado sus tarifas y sus capacidades.

El problema general de diseño de intercambiadores urbanos puede ser dividido en dos subproblemas:

I. *Un subproblema de asignación de la demanda,* que describe como los usuarios eligen el modo de transporte, los intercambiadores y las rutas en el sistema de transporte. En estos modelos con modos combinados hay que fijar qué decisiones son representadas por la demanda de servicios de transporte y cuáles por la red de transporte (oferta de servicios). En esta fase mediante un equilibrio entre oferta y demanda se produce la partición modal y su enrutamiento por los distintos intercambiadores. Este subproblema permite evaluar el comportamiento de los usuarios en cada diseño concreto de red.

II. *Un subproblema de diseño de redes,* donde se elige cuantos intercambiadores y donde se localizarán, la dimensión de sus aparcamientos así como sus tarifas y el tipo de red secundaria para alimentar a estos intercambiadores.

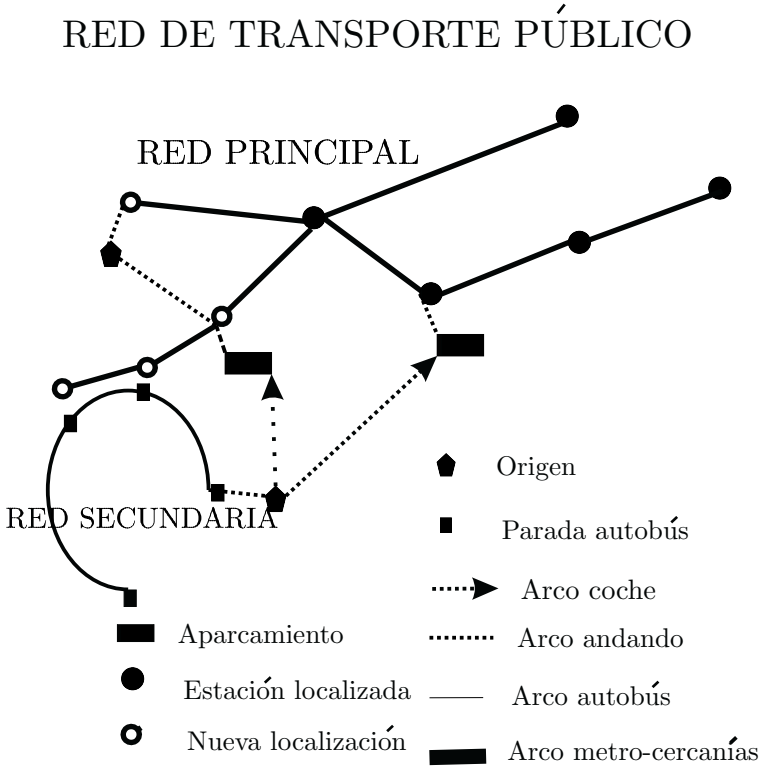


Figura 6.1: Red de transporte público jerárquica

La figura 6.2 recoge la interrelación entre ambos subproblemas. En la fase de asignación de la demanda los usuarios evalúan el diseño del sistema de transporte y eligen el modo de transporte, los intercambiadores y el tipo de aparcamiento, en función de los costes generalizados de tranporte en la red. El planificador del sistema, usualmente una autoridad, evalúa el coste/beneficio de dicho diseño, que dependerá del nivel de servicio de estos intercambiadores y generará un nuevo plan de actuación en el sistema. Las decisiones que realiza afectan a la topología de la red (localización de intercambiadores y tipo de alimentación) que son decisiones propias de un nivel de decisión estratégico y por otro lado, toma decisiones sobre la capacidad y tarifación de los aparcamientos, que son decisiones propias de un nivel táctico.

El problema de asignación de la demanda, se formula mediante un modelo de equilibrio con modos combinados entre oferta y demanda de servicios de transporte. El modelo de demanda es un modelo logit anidado y el modelo de oferta (red de transporte) ha sido simplificado considerablemente.

Hemos formulado el problema de diseño de intercambiadores multimodales urbanos mediante un modelo binivel no lineal mixto. En la sección 6.3 se describirá el nivel inferior y en la sección 6.4 el modelo completo.

6.3 Modelo de equilibrio con modos combinados

A continuación describimos la modelización de la demanda y de la oferta del modelo de equilibrio empleado.

6.3.1 Modelización de la demanda

Supondremos que existe una demanda potencial de viajes origen-destino (O-D) para nuestro periodo de planificación, representada por la matriz O-D $\{\bar{g}_\omega\}_{\omega \in W}$, donde $\omega = (i, j)$ es un par de demanda

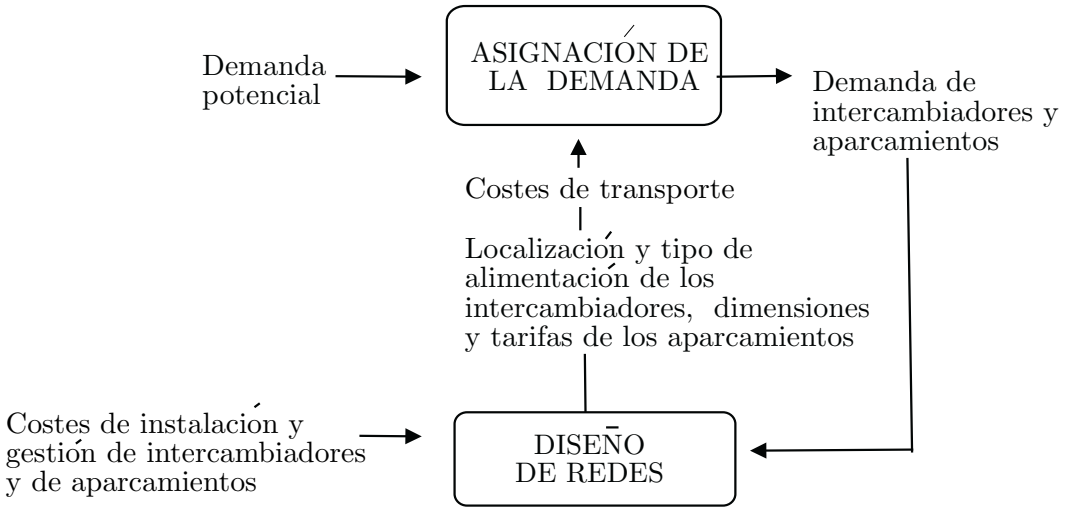


Figura 6.2: El problema de diseño de intercambiadores multimodales urbanos

entre el origen i y el destino j y W es el conjunto de pares O-D. Hemos empleado un modelo logit anidado (ver el libro, por ejemplo, de Ben-Akiva y Lerman [14]) para desagregar esta matriz de demanda por modo de transporte, por intercambiador y por tipo de aparcamiento.

Hemos considerado la siguiente jerarquía en las decisiones que efectúan los usuarios:

1. Elección del modo de transporte. La primera decisión a la hora de realizar un viaje es el modo de transporte. Supondremos que los usuarios eligen una de las siguientes alternativas.
   (a) Park'n ride. La primera componente del viaje se realiza en vehículo privado hasta el intercambiador, se aparca el vehículo, y se completa el viaje empleando la red principal de transporte público.
   (b) Transporte público con acceso mediante la red secundaria. En esta alternativa, el acceso al intercambiador se realiza empleando la red secundaria, que consideraremos que está formada por líneas de autobuses locales.
   (c) Transporte público con acceso andando o en bicicleta. Los viajes se realizan en transporte público y el acceso a la red principal se realiza caminando o en bicicleta.
   (d) Otros. Esta alternativa agrupa todos los modos de transporte que no emplean los intercambiadores, como el vehículo privado, la motocicleta, directamente en autobús (sin emplear la red principal), etc.

   Denotaremos cada una de estas cuatro alternativas por $(s) \in \{a, b, c, d\}$.

   Una función logit G^k produce las proporciones de viajes en cada alternativa de acuerdo a la siguiente fórmula

$$G^k_\omega(\mathbf{U}^*_\omega) = \frac{\exp\{-(\alpha^k + \beta_1 U^{k*}_\omega)\}}{\sum_{k' \in \{a,b,c,d\}} \exp\{-(\alpha^{k'} + \beta_1 U^{k'*}_\omega)\}}, \quad k \in \{a, b, c, d\}, \quad \omega \in W, \tag{6.1}$$

   donde U^{k*}_ω es la percepción del coste generalizado de transporte en el par O-D ω y para el modo k, que corresponde al uso óptimo de la red, $\{\mathbf{U}^*_\omega\}$ es el vector de costes generalizados para todos los modos presentes y α^k, β_1 son parámetros. El coste generalizado de la alternativas (c) y (d) se calcula como la utilidad compuesta de sus subalternativas mediante el denominado "log-suma".

2. Elección de intercambiador . Para determinar la proporción de viajes tipo *park'n ride* para el par O-D ω a través del intercambiador t introducimos la siguiente función logit

$$G^a_{\omega,t}(\mathbf{U}^{a*}_\omega) = \frac{\exp\{-(\alpha_t + \beta_2 U^{a*}_{\omega,t})\}}{\sum_{t' \in I_\omega} \exp -\{(\alpha_{t'} + \beta_2 U^{a*}_{\omega,t'})\}}, \quad t \in I_\omega, \quad \omega \in W. \tag{6.2}$$

La utilidad de la alternativa (a), $\mathbf{U}_{\omega}^{a*}$, se calcula como el "log-suma" de los costes de transporte a través de cada intercambiador, $(U_{\omega,t}^{a*},\ t \in I_\omega)$, esto es

$$U_{\omega}^{a*} = \frac{-1}{\beta_2} \log \left(\sum_{t' \in I_\omega} \exp -\{(\alpha_{t'} + \beta_2 U_{\omega,t'}^{a*})\} \right), \tag{6.3}$$

donde I_ω es el conjunto de intercambiadores disponibles para la demanda $\omega \in W$. El parámetro α_t representa el atractivo del nodo de transferencia t, debido a factores no incluidos en los costes de transporte $\mathbf{U}_{\omega,t}^{a*}$ percibidos por los usuarios, tales como: seguridad, confort, sistemas de información y venta de billetes, etc. y β_2 pondera la importancia de los costes de transporte respecto a estos factores en el proceso de decisión. Supondremos que los usuarios de las alternativas (b) y (c) eligen el intercambiador que minimiza su tiempo total del viaje.

La elección del intercambiador constituye el segundo nivel del modelo logit anidado.

3. Elección del tipo de aparcamiento. Una conclusión del trabajo de Hunt y Teply [131] es que es apropiado emplear un anidamiento en la estructura jerárquica de las decisiones para describir el aparcamiento dentro del intercambiador (en el garaje) o aparcar en la calle. Denotaremos respectivamente por 1 y 2 cada una de estas formas de aparcamiento. Una nueva función logit dará las proporciones de uso de cada tipo de aparcamiento.

$$G_{\omega,t}^{a_s}(\mathbf{U}_{\omega,t}^{a*}) = \frac{\exp\{-\left(\alpha^s{}_t + \beta_3 U_{\omega,t}^{a_s *}\right)\}}{\sum_{s' \in \{1,2\}} \exp\{-(\alpha_t^{s'} + \beta_3 U_{\omega,t}^{a_{s'} *})\}}, \quad s \in \{1,2\},\ t \in I_\omega,\ \omega \in W \tag{6.4}$$

La partición modal de la demanda potencial está dada por la expresión

$$g_{\omega}^{k} = G_{\omega}^{k}(\mathbf{U}_{\omega}^{*})\bar{g}_{\omega} \quad k \in \{a,b,c,d\}.$$

El número de usuarios de tipo *park'n ride* a través del intercambiador t está dada por

$$g_{\omega,t}^{a} = G_{\omega,t}^{a}(\mathbf{U}_{\omega}^{a*}) g_{\omega}^{a},$$

y el número de viajeros en modo *park'n ride* con aparcamiento del tipo s en el intercambiador t es

$$g_{\omega,t}^{a_s} = G_{\omega,t}^{a_s}(\mathbf{U}_{\omega,t}^{a*}) g_{\omega,t}^{a}.$$

Las anteriores expresiones se pueden escribir de forma abreviada por

$$\mathbf{g} = \Phi(\mathbf{U}^*, \mathbf{g}) \tag{6.5}$$

La distribución de la demanda viene dada por la función Φ que depende de los costes generalizados de transporte y de la propia distribución de la demanda.

6.3.2 Modelización de la oferta

La oferta de transporte está representada por una red multimodal, la cual está definida por una red de tráfico, la red de transporte público (principal y secundaria) y los nodos de transferencia entre redes, que se denominan intercambiadores multimodales urbanos.

Nuestras variables de diseño no afectan significativamente a los costes de transporte en la red de tráfico y en la de transporte público, esto es debido a que la cuota de mercado de los viajes combinados es todavía pequeña y cualquier incentivo de estos viajes, por importante que este sea, no pueden descongestionar la red de tráfico. Por este motivo, asumiremos que los costes de transporte en las alternativas $\{b,c,d\}$ son constantes y por tanto independientes de la demanda. Supondremos que los valores $U_{\omega}^{b*}, U_{\omega}^{c*}$, y U_{ω}^{d*} son conocidos para cualquier topología de la red de transporte. Por ejemplo, los costes de transporte se pueden obtener resolviendo un problema de asignación de tráfico para una determinada matriz de demanda O-D. Igualmente se pueden calcular los costes de transporte en la red principal de transporte público.

El coste de transporte, U_ω^{a*}, depende de la demanda y los costes de transferencia de la capacidad de los aparcamientos y de sus tarifas. Otros factores como los tiempos en aparcar o andando hasta la parada, se tienen en cuenta a través de los parámetros de la función de coste de transferencia.

Hemos definido dos costes de transferencia, uno para el aparcamiento dentro del intercambiador y otro para el aparcamiento en la calle. Definimos para cada intercambiador t,

$$c_t^s(f) = v_t^s + B_t^s \left(\frac{f}{u_t^s}\right)^{n_s}, \quad s \in \{1,2\} \tag{6.6}$$

donde c_t^s es el coste de aparcamiento para un nivel de servicio del aparcamiento f, n_s es un parámetro positivo, u_t^1 es la capacidad del aparcamiento t y u_t^2 es el número de aparcamientos en las calles del entorno del intercambiador t. El parámetro u_t^2 se considera fijo y u_t^1 es una variable de diseño. El parámetro v_t^1 representa el coste del aparcamiento t cuando está vacío y viene definido mediante la fórmula $v_t^1 = \tilde{v}_t^1 + \mu d_t$ donde $\tilde{v}_t^1$ es la tarifa del aparcamiento y d_t es la distancia del aparcamiento hasta la parada de transporte público, el parámetro μ transforma los costes medidos en unidades de tiempo a unidades monetarias. Los parámetros v_t^2, d_t y B_t^s deberán ser calibrados con las bases de datos existentes.

El número de usuarios de los dos tipos de aparcamientos se calculan por la expresión

$$g_t^{a_s} = \sum_{\omega \in W_t} g_{\omega,t}^{a_s}, \quad t \in I,\ s \in \{1,\ 2\}, \tag{6.7}$$

donde I es el conjunto de intercambiadores, $W_t = \{\omega \in W \,/\, t \in I_\omega\}$ y el flujo vehicular en el aparcamiento t se calcula

$$g_t^s = \frac{g_t^{a_s}}{\tau}, \tag{6.8}$$

donde τ es la tasa de ocupación vehicular; g_t^s representa el número de vehículos en el aparcamiento del intercambiador t cuando $s = 1$ y el número de vehículos aparcados en la calle cuando $s = 2$.

Las restricciones para garantizar que la capacidad del aparcamiento no sea sobrepasada son $g_t^s \leq u_t^s$ para todo t y $s \in \{1,2\}$, no se han considerado explícitamente, pero sí son tenidas en cuenta mediante los costes de aparcamiento. Cuando la demanda de los aparcamientos se aproxima a su capacidad, o incluso la sobrepasa, los costes generalizados empiezan a crecer, forzando a los usuarios a elegir un aparcamiento o modo de transporte alternativo.

El coste generalizado de transporte para el modo combinado a_s, para el par O-D ω y a través del intercambiador t se calcula por

$$U_{\omega,t}^{a_s*}(g_t^s) = \frac{c_t^s(g_t^s)}{\tau} + \bar{U}_{\omega,t}^{a_s*} \tag{6.9}$$

donde $\bar{U}_{\omega,t}^{a_s*}$ es el coste de transporte en la red multimodal de la primera y tercera componente del viaje combinado.

6.3.3 Modelo del nivel inferior

La expresión (6.9) pone de manifiesto que los costes generalizados de la alternativa *park'n ride* dependen de la demanda, de la capacidad y de las tarifas de los aparcamientos. Hemos supuesto que los costes de transporte $\mathbf{U}^{a*}, \mathbf{U}^{b*}, \mathbf{U}^{c*}$ no dependen de la demanda $\mathbf{g}$ pero sí de las variables de diseño: localización y tipo de alimentación de los intercambiadores.

La figura 6.3 muestra cinco posibles localizaciones de los intercambiadores con relación a una línea de la red de transporte público principal. También está representada la red secundaria que alimenta a esta línea principal. El objetivo de esta figura es ilustrar las relaciones entre las variables de diseño y los costes de transporte.

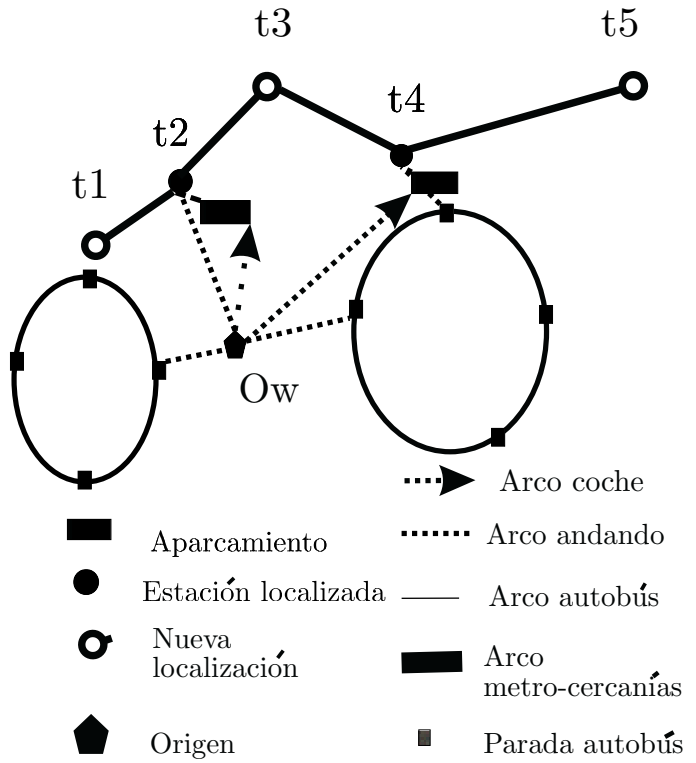


Figura 6.3: Costes generalizados frente a las variables de diseño

Los puntos negros representan que un intercambiador está localizado. El pentágono corresponde al centroide origen de una demanda. Las líneas punteadas son los modos de transporte para acceder a la red primaria o secundaria (coche, bicicleta o andando). Las dos elipses representan las líneas secundarias que alimentan a los intercambiadores. En este ejemplo están representadas dos líneas de autobuses que son atractivas para un origen O_ω.

Ahora explicaremos la dependencia de los costes generalizados para viajar del origen O_ω a un destino D_ω en función de las variables de diseño.

El coste U_ω^{a*} es la percepción del viaje $\omega = (O_\omega, D_\omega)$ en modo *park'n ride*. Un usuario de esta alternativa puede realizar su intercambio modal en uno de los dos nodos t_2 o t_4 que son las estaciones (intercambiadores) que han sido localizadas. Si sólo se hubiese localizado un intercambiador en el nodo t_5, entonces el coste de la primera componente del viaje combinado sería mayor pero el coste de la tercera componente sería menor. Esto pone en evidencia como influye la variable localización de los intercambiadores ($\mathbf{y}$) en el coste de esta alternativa. Por otro lado, el coste de transferencia depende de las capacidades y tarifas de los aparcamientos, estas variables se denotan por $\mathbf{u}, \mathbf{v}$, por tanto, existe la siguiente relación funcional para el coste de transporte $\mathbf{U}_\omega^{a*}(\mathbf{y}, \mathbf{u}, \mathbf{v})$.

El coste de transporte $\mathbf{U}_\omega^{b*}$, de la alternativa de transporte público con acceso a través de la red secundaria, es la suma del coste de acceso a la red principal, más el coste de transporte en la red principal. La primera componente depende del tipo de alimentación decidida por el planificador, esta variable la denominaremos $\mathbf{z}$. Esta dependencia es una función de las frecuencias del servicio, del recorrido, etc. y, desde luego, de la decisión de abrir una estación donde puedan tener parada las líneas principales de transporte público. En la figura 6.3 se muestra la decisión de abrir t_4 y esta estación es empleada por los usuarios del modo de transporte (b). El hecho de que el acceso a la red principal se haga por una u otra estación también condiciona el coste de viaje en la red principal. En resumen, el coste de la primera componente del viaje depende de las variables $\mathbf{z}$ e $\mathbf{y}$; y el coste de la segunda componente depende sólo de $\mathbf{z}$, por tanto, el coste total es una función $U_\omega^{b*}(\mathbf{y}, \mathbf{z})$.

El coste de transporte $\mathbf{U}_\omega^{c*}$, de la alternativa con acceso andando o en bicicleta a la estación, depende

de la localización de los intercambiadores. En la figura, se muestra que, para esas decisiones de localización, los usuarios sólo tienen acceso por el intercambiador t_2, debido a que el t_4 está demasiado lejos para ser accesible a pie. Si se hubiera localizado otro intercambiador, se producirían (si los usuarios cambiasen de intercambiador) otros costes de acceso y de viaje en la red, por tanto, este coste es una función del tipo $U_\omega^{b*}(\mathbf{y})$. Finalmente, el coste de la alternativa $\mathbf{U}_\omega^{d*}$ es el coste (medio) de todas las alternativas que no emplean los intercambiadores y, por tanto, no dependen de las variables de decisión aquí consideradas.

Si denotamos por $\mathbf{x}$ todas las variables de diseño, esto es, $\mathbf{x} = (\mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v})$, podemos expresar los costes generalizados de transporte en función de las variables de demanda y de diseño, es decir, $\mathbf{U}^*(\mathbf{g}, \mathbf{x})$. La expresión explícita de $\mathbf{U}^*(\mathbf{g}, \mathbf{x})$ está dada en las expresiones (6.6)-(6.9) que junto con la expresión (6.5) nos lleva a la primera formulación tipo punto fijo de las condiciones de equilibrio

$$\mathbf{g} = \Phi(\mathbf{U}^*(\mathbf{g}, \mathbf{x}), \mathbf{g}) = \Gamma(\mathbf{g}, \mathbf{x}) \tag{6.10}$$

Si suponemos que los parámetros del modelo logit satisfacen $\beta_j > 0$ y $\beta_1 < \beta_2 < \beta_3$ las condiciones de equilibrio (ver el apéndice I de este capítulo) se pueden derivar de la solución del siguiente problema de optimización que definirá el nivel inferior.

$$\begin{array}{l} \text{minimizar}_{\,\mathbf{g}} T(\mathbf{g}, \mathbf{x}) = \sum_{s\in\{1,2\}} \sum_{t\in I} \int_0^{g_t^s} c_t^s(s, \mathbf{x}) ds + \sum_{\omega\in W} \left[\bar{U}_{\omega,t}^{a_s *} g_{\omega,t}^{a_s} + \right. \\ \qquad \left. + \sum_{k\in\{b,c,d\}} U_\omega^{k*} g_\omega^k \right] + G(\mathbf{g}) \\ \text{sujeto a} \end{array} \tag{LLP(x)}$$

$$\sum_{k\in\{a,b,c,d\}} g_\omega^k = \bar{g}_\omega, \quad \omega \in W \tag{6.11}$$

$$\sum_{t\in I_\omega(x)} g_{\omega,t}^a = g_\omega^a, \quad \omega \in W \tag{6.12}$$

$$\sum_{\omega\in W_t} g_{\omega,t}^{a_s} = g_t^{a_s} \quad t \in I,\ s \in \{1,2\} \tag{6.13}$$

$$g_{\omega,t}^{a_1} + g_{\omega,t}^{a_2} = g_{\omega,t}^a \quad \omega \in W_t,\ t \in I \tag{6.14}$$

donde la función $G(\mathbf{g})$ está definida por

$$\begin{aligned} G(\mathbf{g}) &= (1/\beta_1) \sum_{k\in\{a,b,c\}} \sum_{\omega\in W} g_\omega^k (\ln g_\omega^k - 1 + \alpha^k) - (1/\beta_2) \sum_{\omega\in W} g_\omega^a (\ln g_\omega^a - 1) \\ &+ (1/\beta_2) \sum_{\omega\in W} \sum_{t\in I_\omega} g_{\omega,t}^a (\ln g_{\omega,t}^a - 1 + \alpha_t) - (1/\beta_3) \sum_{\omega\in W} \sum_{t\in I_\omega} g_{\omega,t}^a (\ln g_{\omega,t}^a - 1) \\ &+ (1/\beta_3) \sum_{\omega\in W} \sum_{t\in I_\omega} \sum_{s\in\{1,2\}} g_{\omega,t}^{a_s} (\ln g_{\omega,s}^{a_s} - 1 + \alpha_t^s) \end{aligned}$$

donde $G(\mathbf{g})$ es el término de la función objetivo que corresponde a la desagregación de la demanda mediante el modelo logit anidado.

Denotamos de forma abreviada el problema anterior por

$$\begin{array}{ll} \text{minimizar} & T(\mathbf{g}, \mathbf{x}), \\ \text{sujeto a} & \mathbf{g} \in \Omega(\mathbf{x}) \end{array} \tag{LLP(x)}$$

donde $\Omega(\mathbf{x})$ denota el conjunto factible definido por las restricciones (6.11)-(6.12).

6.4 Un modelo de programación binivel para el diseño de intercambiadores

El problema de diseño tiene en cuenta decisiones estratégicas y tácticas. En el nivel estratégico, se consideran la localización de los intercambiadores, $\mathbf{y}$, y el tipo de diseño para la red secundaria, $\mathbf{z}$. En un nivel táctico, se consideran la capacidad de los aparcamientos, $\mathbf{u}$, y sus tarifas, $\mathbf{v}$. Denotamos por $\mathbf{x}$ este conjunto de variables de diseño que están asociadas con el nivel superior, esto es $\mathbf{x} := (\mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v})$.

La función objetivo empleada está definida como la diferencia entre el coste y el beneficio en el sistema de transporte. Para poder definir esta función, todos los factores, tanto económicos como sociales, se deben expresar en unidades monetarias, debiendo ser cuidadosos en la calibración de esta transformación.

Consideramos un único decisor, quien controla $t = 1, \ldots, m$ sitios potenciales para localizar los intercambiadores. Suponemos que el decisor dispone de un criterio para evaluar el beneficio, tanto económico como social, del nivel de servicio en la red de transporte y asumimos que este beneficio $B(\mathbf{g}, \mathbf{v})$ depende de la desagregación de la demanda y de las tarifas de los aparcamientos $\mathbf{v}$.

El coste en la red de transporte tiene tres componentes: un coste de localización $L(\mathbf{y})$, que es el coste por abrir los intercambiadores; un coste de gestión e instalación de los aparcamientos, $P(\mathbf{u})$, que dependerá de la capacidad instalada de aparcamiento; y la tercera, es el coste del diseño de la red secundaria, $R(\mathbf{y}, \mathbf{z})$, que es una función de las variables estratégicas.

Las variables de localización son binarias y cada valor tiene asociado una de las dos posibilidades, la de abrir o no, un determinado intercambiador. Las variables $\mathbf{z}$ son enteras y cada uno de sus posibles valores tiene asociado un tipo determinado de diseño. Las variables tácticas son continuas. La formulación matemática del problema de diseño es la siguiente

$$\begin{aligned} &\text{minimizar}_{\mathbf{x}} \Psi(\mathbf{x}, \mathbf{g}) = L(\mathbf{y}) + R(\mathbf{y}, \mathbf{z}) + P(\mathbf{u}) - B(\mathbf{g}, \mathbf{v}), \\ &\text{sujeto a} \quad \mathbf{y} \in Y \subset \{0,1\}^m \\ &\qquad\qquad \mathbf{z} \in Z \subset \mathcal{Z}^m \\ &\qquad\qquad (\mathbf{y}, \mathbf{z}) \in S \subset \{0,1\}^m \times \mathcal{Z}^m \\ &\qquad\qquad \mathbf{u} \in U \subset \mathcal{R}^m \\ &\qquad\qquad \mathbf{v} \in V \subset \mathcal{R}^m \end{aligned} \qquad [\text{ULP}(\mathbf{g})]$$

En esta formulación, los conjuntos Z e Y representan otras restricciones de inversión, como restricciones presupuestarias, etc. El conjunto S tiene en cuenta las interacciones entre los problemas de localización de los intercambiadores y de alimentación de los mismos. Los conjuntos U y V representan restricciones en los recursos destinados a los aparcamientos.

Se pueden integrar los problemas de diseño y de equilibrio en redes mediante la programación matemática binivel. En el nivel superior el decisor localiza y diseña el tipo de alimentación de los intercambiadores, así como las facilidades de aparcamientos y en el nivel inferior los usuarios eligen el modo de transporte, el intercambiador y el tipo de aparcamiento en función de la red diseñada.

El modelo binivel se formula del siguiente modo

$$\begin{aligned} &\text{minimizar}_{\mathbf{x}} \Psi(\mathbf{x}, \mathbf{g}) = L(\mathbf{y}) + R(\mathbf{y}, \mathbf{z}) + P(\mathbf{u}) - B(\mathbf{g}, \mathbf{v}), \\ &\text{sujeto a} \quad \mathbf{x} \in X \\ &\qquad\qquad \mathbf{g} = \arg \ \min_{\mathbf{q} \in \Omega(\mathbf{x})} T(\mathbf{q}; \mathbf{x}) \end{aligned} \qquad [\text{BLP}]$$

donde X es la región factible de ULP($\mathbf{g}$).

6.5 Algoritmos heurísticos para el problema de diseño de intercambiadores

Las variables del nivel estratégico $(\mathbf{y}, \mathbf{z})$ están asociadas al problema de expansión de la red de transporte público en áreas suburbanas y un aspecto secundario del mismo es el diseño de aparcamientos disuasorios en estas zonas. El BLM resuelve ambos problemas simultáneamente, pero se podían haber planteado separadamente. Un ejemplo, lo constituye el capítulo 5, donde se aborda exclusivamente el problema de aparcamientos disuasorios. En esta sección, consideramos que las variables tácticas son fijas y se desea encontrar el valor de las estratégicas $(\mathbf{y}, \mathbf{z})$. Esta situación conduce a un nuevo problema binivel en las variables estratégicas $(\mathbf{y}, \mathbf{z})$

$$\begin{array}{ll} \text{minimizar} & \Psi(\mathbf{y}, \mathbf{z}, \mathbf{g}), \\ \text{sujeto a} & \mathbf{y} \in Y \subset \{0,1\}^m \\ & \mathbf{z} \in Z \subset \mathcal{Z}^m \\ & (\mathbf{y}, \mathbf{z}) \in S \subset \{0,1\}^m \times \mathcal{Z}^m \\ & \mathbf{g} = \arg\min_{\mathbf{q} \in \Omega(\mathbf{y},\mathbf{z})} T(\mathbf{q}; \mathbf{y}, \mathbf{z}) \end{array}$$

La relación implícita entre la variable demanda $\mathbf{g}$ y las variables de diseño $(\mathbf{y}, \mathbf{z})$ define una función $\mathbf{g} = \Theta(\mathbf{y}, \mathbf{z})$, ya que el problema LLP($\mathbf{x}$) es estrictamente convexo sobre un conjunto compacto y por tanto tiene solución única para cada valor del par $(\mathbf{y}, \mathbf{z})$. El modelo resultante se puede expresar por

$$\begin{array}{ll} \text{minimizar} & \bar{\Psi}(\mathbf{y}, \mathbf{z}) = \Psi\left(\mathbf{y}, \mathbf{z}, \Theta(\mathbf{y}, \mathbf{z},)\right), \\ \text{sujeto a} & \mathbf{y} \in Y \subset \{0,1\}^m \\ & \mathbf{z} \in Z \subset \mathcal{Z}^m \\ & (\mathbf{y}, \mathbf{z}) \in S \subset \{0,1\}^m \times \mathcal{Z}^m \end{array} \qquad \text{[BLP']}$$

6.5.1 Algoritmos para el LLP

La mayor dificultad de la anterior formulación es que la función $\Theta(\mathbf{y}, \mathbf{z})$ no se conoce explícitamente, pero sí está definida implícitamente por el modelo de equilibrio desarrollado en la sección 6.3. Hay dos caminos alternativos para resolverlo: el primero pasa por resolver el problema de optimización LLP($\mathbf{x}$), por ejemplo mediante la clase CG/SD, y el segundo es resolver la formulación del tipo punto fijo (6.10). El método más natural de resolver este sistema de ecuaciones es mediante un esquema de iteración funcional

$$\mathbf{g}^{i+1} = \Gamma(\mathbf{g}^i, \mathbf{x})$$

Si la sucesión generada $\{\mathbf{g}^i\}$ converge a un punto $\hat{\mathbf{g}}$, entonces se cumple que este punto límite es una solución del sistema por la continuidad de Γ. Una mejora de este esquema se obtiene observando la estructura del sistema de ecuaciones (6.10), que es:

$$\begin{array}{c} \Gamma_\omega(\mathbf{g}_\omega, \mathbf{g}_t, \mathbf{x}) = \mathbf{g}_\omega, \quad \forall \omega \ \text{ demanda}, \\ \displaystyle\sum_{\omega \in W_t} g^{a_s}_{\omega,t} = \gamma g^s_t, \quad \forall t \ \text{ intercambiador}, \end{array} \qquad (6.15)$$

donde Γ_ω está definida por las relaciones (6.1)-(6.4).

Si conociésemos el nivel de servicio de los aparcamientos, esto es, las variables g^s_t, podríamos descomponer este sistema en $|W|$ sistemas de menor dimensión, uno por cada demanda $\omega \in W$. Además, cada uno de estos sistemas también puede ser resuelto mediante un esquema de iteración funcional del tipo $\Gamma_\omega(\mathbf{g}^i_\omega, \mathbf{g}_t, \mathbf{x}) = \mathbf{g}^{i+1}_\omega$. Lo anterior conduce a un algoritmo de tipo Gauss-Seidel para resolver el sistema (6.15). Este método fija todas las variables a su valor actual excepto $\mathbf{g}_\omega$ y $\mathbf{g}_t$, entonces el correspondiente sistema se resuelve mediante el método de iteración funcional. Si los pares ω son elegidos cíclicamente y si el anterior método es convergente, entonces el punto límite es la solución del sistema buscado. El algoritmo de Gauss-Seidel está descrito en la tabla 6.1.

Tabla 6.1: Algoritmo de Gauss-Seidel para el LLP($\mathbf{x}$)

0. *(Inicialización):* Elegir dos números enteros n_W y n que son respectivamente el número de iteraciones funcionales realizadas para cada par y el número de iteraciones principales. Tomar un valor inicial de $\mathbf{g}^0$ y hacer $i = 0$.
1. Para todo $\omega \in \{1, \ldots, |W|\}$ hacer

   1.1 *(Inicialización del par ω):* Tomar $\mathbf{q}_\omega^0 = \mathbf{g}_\omega^i$ y $\mathbf{q}_t^0 = \mathbf{g}_t^i$

   1.2 *(Iteración funcional):* Para $j \in \{0, \ldots, n_W - 1\}$ hacer

$$\begin{aligned} \mathbf{q}_\omega^{j+1} &= \Gamma_\omega(\mathbf{q}_\omega^j, \mathbf{q}_t^j, \mathbf{x}) \\ (q_t^s)^{j+1} &= \frac{1}{\gamma}\left\{(q_{\omega,t}^{a_s})^j + \sum_{\omega' \in W_t, \omega' \neq \omega} (g_{\omega',t}^{a_s})^i\right\} \\ j &= j+1 \end{aligned}$$

   1.3. Actualizar las variables de demanda del par ω por

$$\begin{aligned} \mathbf{g}_\omega^{i+1} &= \mathbf{q}_\omega^{n_W} \\ (g_t^s)^i &= (q_t^s)^{n_W} \end{aligned}$$

2. Tomar $i = i + 1$. Si $i = n$ parar, en caso contrario volver al paso 1.

6.5.2 Algoritmos golosos para el BLM'

La idea de los algoritmos golosos hacia adelante (FGA) es simple. Dado un conjunto de localizaciones definido por la variable $\mathbf{y}$, el algoritmo explora todas las posibilidades de abrir k nuevas facilidades, siendo la próxima localización aquella que produce un mayor decrecimiento de la función objetivo $\bar{\Psi}(\mathbf{y}, \mathbf{z})$. Una vez que una facilidad es localizada se mantiene a lo largo de todo el proceso. Para formular el algoritmo FGA, definimos el *k-entorno hacia adelante*:

$$\mathcal{N}_k^+(\mathbf{y}) = \left\{\mathbf{y}' \in Y \,/\, \sum_{j=1}^{m} |y_j' - y_j| \leq k \quad \text{e} \quad \mathbf{y} \leq \mathbf{y}'\right\} \text{ para } \mathbf{y} \in Y$$

El conjunto $\mathcal{N}_k^+(\mathbf{y})$ da todas las localizaciones posibles de k nuevas estaciones respecto a las ya localizadas $\mathbf{y}$. Además, debemos fijar el tipo de alimentación de estos nuevos intercambiadores, que lo hacemos mediante el denominado *k-entorno hacia adelante extendido*:

$$\mathcal{S}_k^+(\mathbf{y}) = \left\{(\mathbf{y}', \mathbf{z}') \in \mathcal{N}_k^+(\mathbf{y}) \times Z \,/\, (\mathbf{y}', \mathbf{z}') \in S\right\} \text{ para } \mathbf{y} \in Y$$

El algoritmo completo del FGA está recogido en la tabla 6.2.

Tabla 6.2: Algoritmo FGA

0. *(Inicialización):* Encontrar una solución inicial $(\mathbf{y}^0, \mathbf{z}^0)$. Tomar $i = 1$.
1. Sea $(y^i, \mathbf{z}^i) = \arg \min \left\{\bar{\Psi}(\mathbf{y}, \mathbf{z}) \,/\, (\mathbf{y}, \mathbf{z}) \in \mathcal{S}_k^+(\mathbf{y}^{i-1})\right\}$.
2. Si $\bar{\Psi}(\mathbf{y}^i, \mathbf{z}^i) \geq \bar{\Psi}(\mathbf{y}^{i-1}, \mathbf{z}^{i-1})$, entonces parar. $(\mathbf{y}^{i-1}, \mathbf{z}^{i-1})$ es una solución FGA.
3. Si $\mathcal{S}_k^+(\mathbf{y}^i) = \{\emptyset\}$ entonces $(\mathbf{y}^i, \mathbf{z}^i)$ es una solución del FGA, en caso contrario tomar $i = i + 1$ y volver al paso 1.

Este tipo de algoritmo heurístico tiene una versión complementaria. Suponemos que, al principio, se ha instalado un intercambiador en todas las localizaciones posibles, entonces, el algoritmo va cerrando los intercambiadores en los que sus niveles de servicio sean deficientes. En cada iteración evalúa el beneficio de eliminar cada intercambiador y cierra el peor. Este algoritmo los hemos denominado algoritmo goloso hacia atrás (BGA). El algoritmo es obtenido cambiando el conjunto $\mathcal{S}_k^+(\mathbf{y})$ en FGA por

$$\mathcal{S}_k^-(\mathbf{y}) = \left\{(\mathbf{y}', \mathbf{z}') \in \mathcal{N}_k^-(\mathbf{y}) \times Z \,/\, (\mathbf{y}', \mathbf{z}') \in S\right\} \text{ para } \mathbf{y} \in Y$$

donde

$$\mathcal{N}_k^-(\mathbf{y}) = \left\{ \mathbf{y}' \in Y \,/\, \sum_{j=1}^{m} \left|y_j' - y_j\right| \leq k \text{ y } \mathbf{y} \geq \mathbf{y}' \right\} \text{ para } \mathbf{y} \in Y.$$

6.5.3 Un algoritmo de intercambio para el BLM'

Los algoritmos BGA y FGA tienen dos desventajas fundamentales. La primera es que las facilidades seleccionadas no se pueden cambiar a lo largo del proceso y la segunda es que el coste computacional de explorar cada entorno $\mathcal{S}_k^{\pm}(\mathbf{y})$ es muy alto, debido a que analizan todos los elementos del k-entorno.

El primer problema puede ser solucionado considerando una nueva definición de k-entorno

$$\mathcal{S}_k(\mathbf{y}) = \mathcal{S}_k^+(\mathbf{y}) \cup \mathcal{S}_k^-(\mathbf{y}) \text{ para } \mathbf{y} \in Y,$$

que permita abrir o cerrar una estación en cada iteración.

Para aliviar el coste computacional en la exploración de los $k-$entornos, hemos relajado la exploración de los entornos y ésta se interrumpe una vez que se haya encontrado una solución que reduzca el actual valor de la función objetivo.

El algoritmo de intercambio (IA) está recogido en la tabla 6.3

Tabla 6.3: Algoritmo IA

0. *(Inicialización):* Sea $(\mathbf{y}^0, \mathbf{z}^0)$ una solución inicial. Tomar $i = 1$.
1. Dado el punto factible $(\mathbf{y}^{i-1}, \mathbf{z}^{i-1})$, encontrar (si existe) un punto $(\mathbf{y}', \mathbf{z}') \in \mathcal{S}_k(\mathbf{y}^{i-1})$ con $\bar{\Psi}(\mathbf{y}', \mathbf{z}') < \bar{\Psi}(\mathbf{y}^{i-1}, \mathbf{z}^{i-1})$ entonces tomar $(\mathbf{y}^i, \mathbf{z}^i) = (\mathbf{y}', \mathbf{z}')$, $i = i+1$ y repetir la iteración. En caso contrario, que no exista el anterior punto, parar, $(\mathbf{y}^{i-1}, \mathbf{z}^{i-1})$ es una solución del algoritmo IA.

6.5.4 Un algoritmo de recocido simulado para el BLM'

Los algoritmos FGA, BGA, e IA paran cuando han encontrado un óptimo local (óptimo en el entorno considerado). En ocasiones se ejecutan estos algoritmos con diferentes puntos iniciales, para obtener diferentes óptimos locales con el fin de que uno de ellos sea también óptimo global. El algoritmo de recocido simulado (SAA) ofrece una solución alternativa al poblema de mínimos locales. Dada una solución $(\mathbf{y}, \mathbf{z})$ el método selecciona aleatoriamente una solución $(\mathbf{y}', \mathbf{z}')$ del entorno $\mathcal{S}_k(\mathbf{y})$. Si esta solución mejora el coste de la solución actual, se acepta y, en caso contrario, se aceptará con probabilidad

$$\exp - \left(\frac{\bar{\Psi}(\mathbf{y}', \mathbf{z}') - \bar{\Psi}(\mathbf{y}, \mathbf{z})}{K\mathcal{T}} \right) \tag{6.16}$$

y se rechazará con la probabilidad complementaria. La constante $\mathcal{T}$ se denominada *temperatura* del proceso y determina, según sus valores, movimientos pequeños o grandes de las variables de

optimización. La constante K depende de la escala en que se han medido los costes del sistema y tiene la misión de estandarizar estos valores.

La eficiencia del SAA depende de la estructura del entorno y de un conjunto de parámetros, pero no existen reglas para la obtención de estos valores. A continuación, listamos los valores que han producidos los mejores resultados computacionales en los experimentos numéricos realizados.

(a) La temperatura del sistema va disminuyendo cada vez que el sistema alcanza el equilibrio para dicha temperatura. Este decrecimiento de la temperatura se efectúa por $\mathcal{T} = \alpha\mathcal{T}$ donde α es un parámetro con $0 < \alpha < 1$. En la experiencia computacional se ha empleado el valor de $\alpha = 0.95$.

Cuando la temperatura del proceso alcanza una temperatura final $\mathcal{T}_f$, la búsqueda se detiene. Hemos tomado como temperatura inicial del proceso $\mathcal{T}_0 = 1.0$ y como temperatura final $\mathcal{T}_f = 0.1$.

(b) El equilibrio del sistema viene caracterizado por los parámetros:

NAC máximo número de configuraciones aceptadas.

NRC máximo número de configuraciones consecutivas rechazadas.

Estos parámetros se suelen definir en función del tamaño del problema. Hemos empleado los valores $NAC = [0.85m]$ y $NRC = [2m]$ donde m es la dimensión del vector $\mathbf{y}$ (sitios potenciales donde localizar un intercambiador) y $[\cdot]$ es la parte entera de un número.

(c) El SAA converge, con probabilidad uno, a un óptimo global del problema bajo las hipótesis de que cada elemento del entorno $\mathcal{S}_k$ se elige con igual probabilidad. Esta forma de seleccionar las soluciones tiene gran interés bajo el punto de vista teórico, pero computacionalmente no es eficiente. En los experimentos hemos seleccionado sistemáticamente las componentes del vector $\mathbf{y}$ del siguiente modo. En una primera fase, se van seleccionando consecutivamente las componentes de $\mathbf{y}$ cuyo valor es 0, una vez analizada la última de estas componentes, se repite el proceso, pero para las componentes cuyo valor sea 1.

Si una componente se fija al valor 1, es decir, abrimos ese intercambiador, entonces se debe decidir el tipo de alimentación. Esta elección se realiza aleatoriamente de acuerdo a una cierta función de probabilidad.

(d) Hemos empleado el procedimiento del capítulo 5 para seleccionar el valor de K, concretamente hemos elegido

$$K = \frac{-0.05\left|c^0\right|}{\log(0.1)} = 0.0217\left|c^0\right|$$

donde c^0 es el valor de la solución inicial.

La adaptación del SAA al BLM' está descrita en la tabla 6.4.

6.6 Experimentos computacionales

En esta sección comparamos la eficiencia computacional de los cuatro algoritmos heurísticos desarrollados. Se han generado aleatoriamente los problemas de prueba para este experimento. La figura 6.4 ilustra el procedimiento de obtención de las redes de prueba. En primer lugar se generan tres círculos de diferentes radios e igual centro. En el círculo A se distribuyen aleatoriamente (uniformemente) los centroides destino. Sobre la corona circular definida por los círculos A y B se emplazan aleatoriamente (uniformemente) las posibles localizaciones de los intercambiadores. En la corona circular definida por los círculos B y C se sitúan los centroides origenes de forma aleatoria (uniformemente). El conjunto de pares de demanda O-D se generan, par a par, obteniendo su origen y su destino como se ha descrito anteriormente. Supongamos que se ha generado el par $\omega = (O, D)$ (ver la figura 6.4), entonces calculamos las k distancias menores de ir del origen O al destino D mediante intercambiadores.

Tabla 6.4: Algoritmo SAA

0. *(Inicialización):* Sea $(\mathbf{y}^0, \mathbf{z}^0)$ una solución inicial y sea c^0 su coste. Definimos los parámetros $\mathcal{T}_0$, $\mathcal{T}_f$, y α de acuerdo a la regla (a) y decidimos el valor de K mediante (d). Seleccionamos los parámetros NAC y NRC como en (b). Definimos las variables asociadas a la mejor solución encontrada y las inicializamos mediante $\mathbf{y}^{op} = \mathbf{y}^0$ y $\mathbf{z}^{op} = \mathbf{z}^0$. Ponemos todos los contadores a cero: cNAC=0, cNRC=0 e $i = 0$.
1. Dada la solución $(\mathbf{y}^i, \mathbf{z}^i)$ seleccionamos $(\mathbf{y}', \mathbf{z}') \in \mathcal{S}_k(\mathbf{y}^i)$ empleando el criterio (c). Sea c' el coste de la solución $(\mathbf{y}', \mathbf{z}')$.
2. Si $c' < c^i$ entonces $(\mathbf{y}^{i+1}, \mathbf{z}^{i+1}) = (\mathbf{y}', \mathbf{z}')$ y $c^{i+1} = c'$. Si $c' < c^{op}$ entonces $(\mathbf{y}^{op}, \mathbf{z}^{op}) = (\mathbf{y}', \mathbf{z}')$ y $c^{op} = c'$.
3. En caso contrario, $(\mathbf{y}^{i+1}, \mathbf{z}^{i+1}) = (\mathbf{y}', \mathbf{z}')$ con probabilidad $p = \exp -[(c' - c^i)/K\mathcal{T}]$, y $(\mathbf{y}^{i+1}, \mathbf{z}^{i+1}) = (\mathbf{y}^i, \mathbf{z}^i)$ con probabilidad $1 - p$.
4. Si la solución $(\mathbf{y}', \mathbf{z}')$ ha sido aceptada entonces $cNAC = cNAC + 1$ y $cNRC = 0$. En caso contrario $cNRC = cNRC + 1$. Si $cNAC = NAC$ o $cNRC = NRC$ entonces disminuir la temperatura $\mathcal{T} = \alpha\mathcal{T}$, tomar $cNAC = 0$ y tomar $cNRC = 0$.
5. Si $\mathcal{T} < \mathcal{T}_f$ parar. En caso contrario tomar $i = i + 1$ y volver al paso 1.

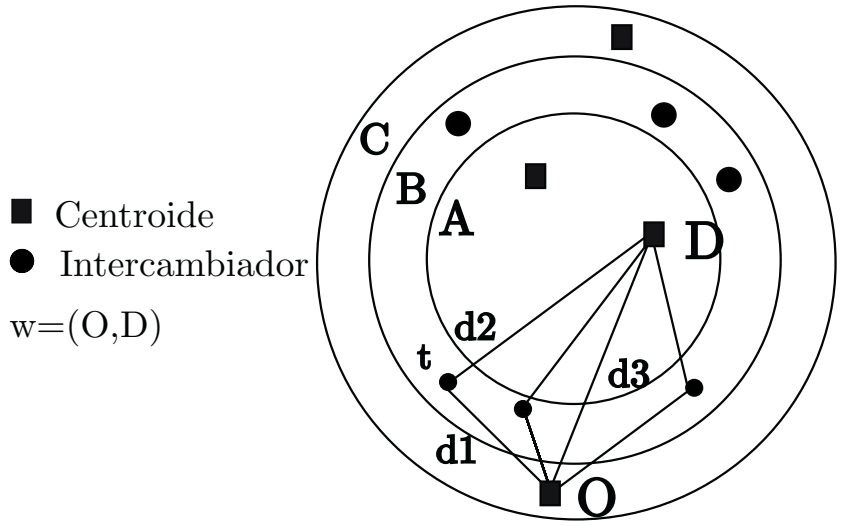


Figura 6.4: Ilustración de la generación de las redes de prueba

Este conjunto de intercambiadores será T_ω. Después se calculará el coste de transporte para cada alternativa, esto es $C^a_\omega, C^b_\omega, C^c_\omega$ y C^d_ω. Hemos supuesto que estos costes son proporcionales a su distancia euclídea (d_i), estando determinada cada constante de proporcionalidad por la velocidad media del modo de transporte. Por ejemplo, el viaje combinado a través del intercambiador t (ver la figura 6.4) será la suma del tiempo empleado en recorrer la distancia d_1 más el tiempo empleado en la distancia d_2. El tiempo empleado en la primera parte dependerá de si se ha recorrido en coche, andando, o en la línea de autobuses locales, mientras que el tiempo empleado en la segunda componente del viaje, d_2, será idéntico para todos los viajes combinados, ya que todos emplearán la misma línea principal de transporte público. El coste de transporte en la alternativa (d) será proporcional a la distancia d_3.

Hemos considerado tres diseños diferentes de la red secundaria para alimentar los intercambiadores que originan tres velocidades de acceso y tres costes diferentes de diseño para los autobuses locales. Estos costes son la suma de la instalación de las líneas y de gestión.

Para obtener los parámetros logit, hemos supuesto que dos localizaciones que generaran los mismos costes de transporte, para un par de demanda determinado, producen el mismo número de usuarios

(de esta demanda) en cada uno de estos intercambiadores. Esto implica que todos los parámetros logit asociados a las localizaciones deben ser iguales, hemos elegido el valor $\alpha_t = 0$.

Por otro lado, hemos asumido que la elección del tipo de aparcamiento, ésto es, aparcar en la calle o en el intercambiador, sólo depende de los costes generalizados y, por tanto, los parámetros satisfacen $\alpha_t^1 = \alpha_t^2$ para todo $t \in I$, siendo este valor 0.

Hemos considerado una partición modal de referencia en el conjunto de la red y, sobre ésta, hemos ajustado los parámetros α^k. Después, teniendo en cuenta los costes de transporte, hemos ajustado mediante mínimos cuadrados ponderados los parámetros β_1 y β_2, de modo que las predicciones efectuadas por el modelo logit se ajusten lo más posible a la partición modal de referencia.

Hemos considerado una medida del beneficio del sistema de transporte, $B(\mathbf{g})$, que depende exclusivamente de la variable $\mathbf{g}$ y su expresión funcional es

$$B(\mathbf{g}) = \sum_{t=1}^{m} B_t(\mathbf{g}) + \mu \left(\bar{U}^{d*} - \sum_{\omega \in W} g_\omega^d U_\omega^{d*} \right), \tag{6.17}$$

donde para cada localización t se define B_t por

$$B_t(\mathbf{g}) = \varsigma \sum_{\omega \in W_t} \left(g_{\omega,t}^a + g_\omega^b \delta_{\omega,t}^b + g_\omega^c \delta_{\omega,t}^c \right),$$

con $\delta_{\omega,t}^k = 1$ si el viaje ω en el modo k emplea el intercambiador t y $\delta_{\omega,t}^k = 0$ en otro caso.

Este beneficio tiene dos componentes, por un lado está el beneficio económico obtenido por las tarifas de transporte público pagadas por los usuarios de esta red y, por otro lado, está el beneficio social que produce la intervención en el sistema de transporte público (reducción de emisiones, reducción de consumo de energía, etc.) debido a la disminución de la cuota de mercado de los vehículos privados (d). La primera componente está determinada por el precio del billete ς y por el número de usuarios en la red de transporte público. El segundo factor se recoge transformando la reducción del número de horas usadas en vehículos privados en beneficio social, para ello, se compara el número total de horas empleadas por los vehículos privados, $\bar{U}^{d*}$, con las empleadas tras la intervención en el sistema de transporte y el parámetro μ transforma esta diferencia a unidades monetarias.

Hemos considerado que el coste de instalación y gestión de los intercambiadores tiene la expresión

$$L(\mathbf{y}) = \sum_{t=1}^{m} f_t y_t$$

Hemos tomado el mismo parámetro f_t para todas las posibles localizaciones.

La formulación matemática del diseño de la red secundaria emplea variables enteras z_t, que toman un conjunto finito de valores, cada uno asociado a un diseño. Hemos supuesto que estas variables sólo pueden tomar los valores $1, 2, 3$, y que la función $C_R(\cdot)$ está definida para cada diseño considerado $1, 2, 3$, proporcionando los costes de cada uno de ellos, por tanto, el coste de la red secundaria está dado por

$$R(\mathbf{y}, \mathbf{z}) = \sum_{t=1}^{m} C_R(z_t) y_t$$

El coste computacional de los algoritmos heurísticos en cada iteración depende crucialmente del parámetro k (que define el tamaño del entorno). Hemos tomado en todas las pruebas computacionales el valor de $k = 1$.

Los códigos han sido ejecutados sobre un ordenador PC de 384 megabytes de RAM y 400 MHz y los programas fuentes se han codificado en FORTRAN (Visual Workbench).

El tamaño de los problemas de prueba se muestran en la tabla 6.5. La columna cuarta muestra el tiempo medio de CPU empleado en la resolución de un modelo LLP. Este valor depende del número

Tabla 6.5: Tamaño de los problemas de prueba

Red de pruebas	$\|W\|^a$	m^b	CPU per LLP^c
NET1	300	25	0.29
NET2	300	50	0.26
NET3	300	100	0.24
NET4	500	100	0.38
NET5	1000	50	0.47
NET6	3000	25	3.12

[a] Número de pares de demanda.
[b] Número posible de localizaciones de intercambiadores.
[c]Tiempo medio CPU empleado en la resolución de un LLP.

Tabla 6.6: Resultados computacionales

Red de pruebas	FGA	BGA	IA	SAA
NET1	-473240.10[a]	-488721.5	-483845.2	-481622.3
	660[b]	882	136	4153
NET2	-466172.6	-485436.0	-469551.8	-472346.2
	1602	3624	288	10013
NET3	-635517.9	-630940.6	-608787.5	-627120.1
	4692	15648	1094	34438
NET4	-512012.4	-506577.9	-490027.4	-494691.6
	5430	15300	1547	46533
NET5	-412825.4	-437841.5	-395456.9	-404498.1
	2241	3564	513	33805
NET6	-466086.2	-471755.7	-469353.4	-474846.6
	741	696	104	9587

[a] Valor de la función objetivo.
[b] Número de evaluaciones de la función objetivo.

de pares O-D considerados y del número de iteraciones realizadas por el algoritmo de Gauss-Seidel. Hemos observado que en los problemas de prueba empleando $n_W = 3$ y $n = 4$ la sucesión $\{\mathbf{g}^i\}$ es convergente y el error relativo $\frac{\|\mathbf{g}^{i+1}-\mathbf{g}^i\|}{\|\mathbf{g}^i\|}$ es aproximadamente de 1.%. Esto indica que este algoritmo es un buen procedimiento para resolver los problemas LLP. La principal desventaja del algoritmo es que no es convergente cuando la capacidad de los aparcamientos ($\mathbf{u}$) es muy pequeña respecto a la demanda potencial. En este caso, el algoritmo genera una sucesión oscilante (no divergente) y para esta situación se hace inevitable recurrir a la formulación mediante programación matemática ya que tiene asociada algoritmos convergentes.

La tabla 6.6 muestra los resultados experimentales. Los mejores resultados se obtienen con los algoritmos FGA e IA. El coste computacional del IA es significativamente menor que el resto. Como conclusión, se recomienda el uso del IA. Para problemas donde se requiera un gran esfuerzo computacional también se puede emplear el SAA y para problemas de menor tamaño, los algoritmos FGA y BFA son competitivos. La figura 6.5 ilustra esta afirmación observando que al principio el IA y SAA obtienen mejores soluciones pero al final esta tendencia se invierte.

6.7 Ejemplo de localización de intercambiadores

En esta sección ilustramos el uso del modelo para localizar nuevas estaciones en una red de metro. Hemos considerado la red de metro de la figura 6.6, que está compuesta por dos líneas de metro y 6

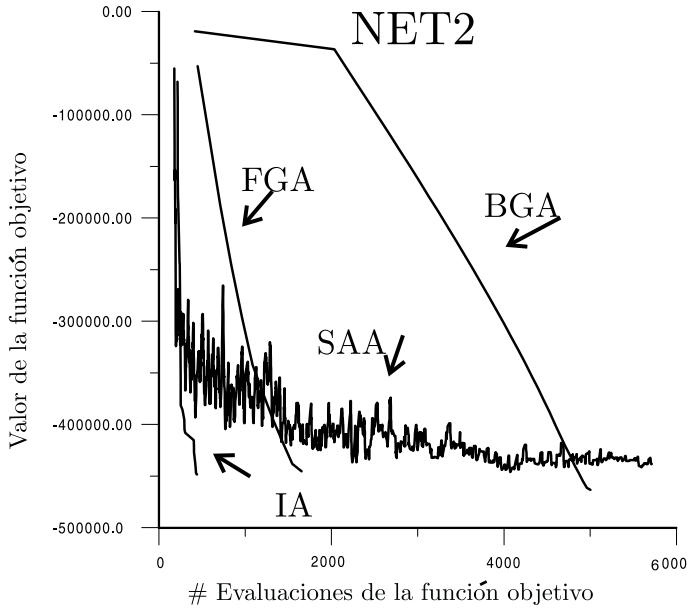
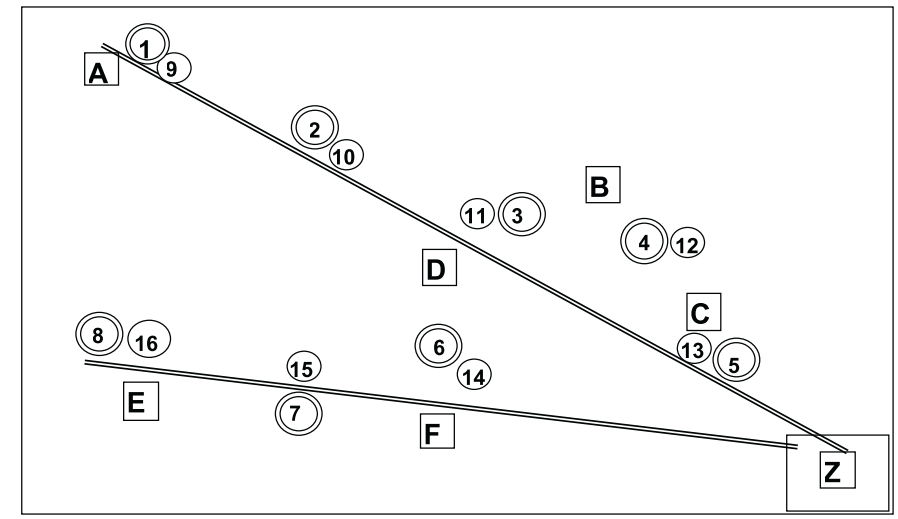


Figura 6.5: Progreso de los algoritmos



Numero de Demandas						
Parametros función BPR:	v^1_t	B^1_t	n_1	v^2_t	B^2_t	n_2
Intercambiador (1-8)	0,09	0,09	2	0,1	0,1	3
Parada (9-16)	0,9	0,9	20	0,1	0,1	3

Parametros función coste inter/parada
Intercambiador (1-8)
Parada (9-16)

Figura 6.6: Red de prueba para la localización de paradas de metro

pares de demanda entre 6 orígenes y un centroide destino. Se supone que en la red se puede instalar hasta 16 nuevas paradas. En las ocho primeras localizaciones, se instalarían paradas con aparcamientos disuasorios (lo que denominaremos intercambiadores) y en el resto simplemente se instala una parada de la línea de metro.

A continuación, mostraremos varios resultados al variar ciertos parámetros y condiciones del modelo. Los resultados de la primera prueba se muestran en la figura 6.7. En esta prueba se decide la localización de intercambiadores y paradas. La solución obtenida sitúa en los nodos 4 y 6 dos intercambiadores y en los nodos 9 y 16 dos paradas. También se muestra la desagregación de los

Prueba Número 1 **Solución: Intercambiadores 4,6,9,16**

DEMANDA O / D	Valor	PARK & RIDE Inter 4 Parking	Inter 4 Calle	Inter 6 Parking	Inter 6 Calle	Parada 9 Calle	Parada 16 Calle	Total Park & R	METRO	COCHE
A - Z	2000	0	0	63	14	0	0	77	681	1242
B - Z	3000	86	21	0	0	0	0	107	1161	1732
C - Z	3500	78	19	0	0	0	0	97	1055	2348
D - Z	3000	35	9	52	12	0	0	108	1025	1867
E - Z	1000	0	0	29	7	0	0	36	320	644
F - Z	2000	0	0	55	12	0	0	67	730	1203
Totales	**14500**	**199**	**49**	**199**	**45**	**0**	**0**	**492**	**4972**	**9036**

Capacidad	Parking	Calle
Inter 1- 8	250	50
Inter 9 - 16	0	0

NFIJ=0
Inter/paradas preinstalados
NMAX=16
Número máximo Inter/paradas
ETA=20
Incentivo por descongestión del tráfico
^{a}=2.5
^{c}=1.5
^{d}=1.0

Intercambiador
Parada
E **Centroide**

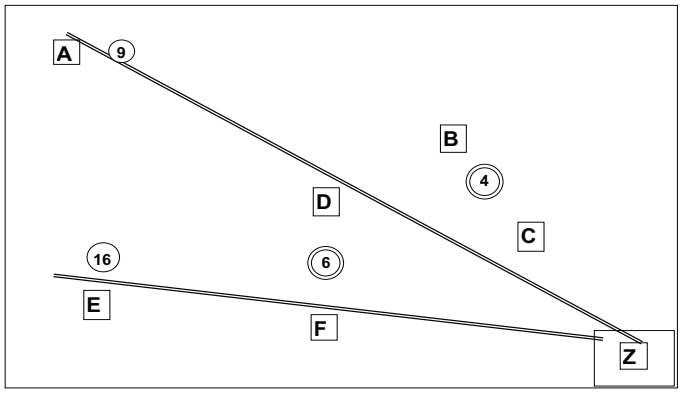


Figura 6.7: Resultados de la prueba 1 para la localización de paradas de metro

14,500 usuarios potenciales por pares de demanda, por modo de viaje y por tipo de aparcamiento. Como en las paradas 16 y 9 no existe posibilidad de aparcar, ya que no es un intercambiador y no existe capacidad en la calle, no existen viajes *park'n ride* a través de estas paradas.

El enfoque del ejemplo anterior diseña las paradas de dos nuevas líneas de transporte, pero en la práctica pudiera ocurrir que lo que se desea es ampliar una línea ya existente. El siguiente ejemplo recoge esta situación. Se supone que ya existe un intercambiador instalado en el nodo 1 y se desea instalar solamente dos nuevas paradas/intercambiadores. Los resultados se muestran en la figura 6.8. La solución coincide con el caso anterior, excepto que se ha eliminado las paradas 9 y 16 para poder satisfacer la restricción del número máximo de estaciones a instalar. Los algoritmos FGA y BGA son los que se pueden adaptar más fácilmente a la nueva situación de limitar el máximo número de paradas.

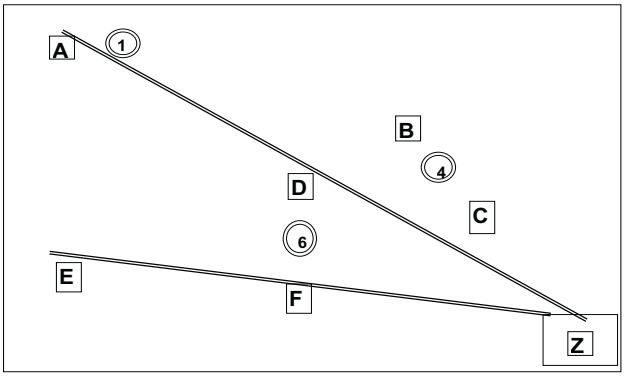
El modelo permite recoger la capacidad de los aparcamientos disuasorios de los intercambiadores. El siguiente experimento es una repetición del primero, exceptuando que la capacidad de los aparcamientos de los intercambiadores se ha triplicado, esto es, tienen una capacidad de 750 plazas. La solución obtenida se muestra en la figura 6.9. El hecho de que la capacidad sea tan elevada ha hecho reemplazar el intercambiador 4 por una estación sin aparcamiento. La parada 16 desaparece, esto puede ser debido a la naturaleza heurística de los algoritmos empleados, pero parece más probable que ocurra por el efecto del aparcamiento en el nodo 6. En el primer caso, la instalación de la parada 16 da servicio a 320 personas de la demanda (E, Z), en este caso, la instalación del intercambiador en el nodo 6 da servicio a 87 en modo *park'n ride*, reduciendo la demanda potencial de la parada 16 y haciendo que no se instale esta parada por falta de una demanda suficiente.

Prueba Número 2 **Solución: Intercambiadores 1,4,6**

DEMANDA O - D	Valor	PARK & RIDE Inter 1 Parking	Inter 1 Calle	Inter 4 Parking	Inter 4 Calle	Inter 6 Parking	Inter 6 Calle	Total Park & R	METRO	COCHE
A - Z	2000	105	35	0	0	11	3	154	654	1192
B - Z	3000	0	0	87	21	0	0	108	1161	1731
C - Z	3500	0	0	80	19	0	0	99	1055	2346
D - Z	3000	0	0	27	7	69	16	119	1021	1860
E - Z	1000	0	0	0	0	47	12	59	0	941
F - Z	2000	0	0	0	0	62	15	77	726	1197
Totales	**14500**	**105**	**35**	**194**	**47**	**189**	**46**	**616**	**4617**	**9267**

Capacidad	Parking	Calle
Inter 1- 8	250	50
Inter 9 - 16	0	0

NFIJ=1
Inter/paradas preinstalados
NMAX=3
Número máximo Inter/paradas
ETA=20
Incentivo por descongestión del tráfico
^{a}=2.5
^{c}=1.5
^{d}=1.0



Intercambiador
Parada
E Centroide

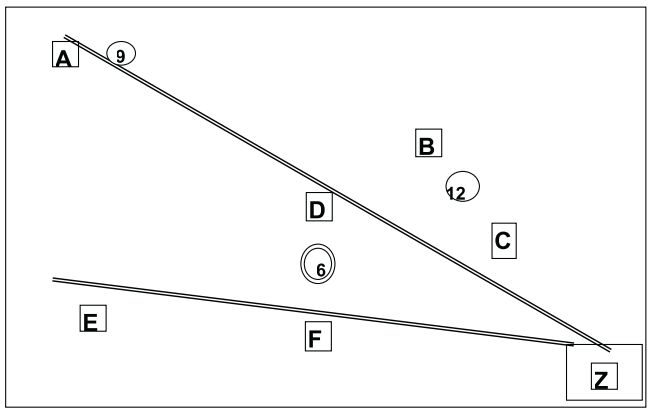
Figura 6.8: Resultados de la prueba 2 para la localización de paradas de metro

Prueba Número 3 **Solución: Intercambiadores 6,9,12**

DEMANDA O - D	Valor	PARK & RIDE Inter 6 Parking	Inter 6 Calle	Parada 9 Calle	Parada 12 Calle	Total Park & R	METRO	COCHE
A - Z	2000	118	10	0	0	128	663	1209
B - Z	3000	0	0	0	0	0	1204	1796
C - Z	3500	0	0	0	0	0	1085	2415
D - Z	3000	141	16	0	0	157	1007	1836
E - Z	1000	78	9	0	0	87	0	913
F - Z	2000	104	9	0	0	113	713	1174
Totales	**14500**	**441**	**44**	**0**	**0**	**485**	**4672**	**9343**

Capacidad	Parking	Calle
Inter 1- 8	750	50
Inter 9 - 16	0	0

NFIJ=0
Inter/paradas preinstalados
NMAX=0
Número máximo Inter/paradas
ETA=20
Incentivo por descongestión del tráfico
α^{a}=2.5
α^{c}=1.5
α^{d}=1.0



Intercambiador
Parada
E Centroide

Figura 6.9: Resultados de la prueba 3 para la localización de paradas de metro

Apéndice I: formulación de las condiciones de equilibrio mediante programación matemática

En este apéndice justificamos que una solución del problema LLP es óptima si y sólo si satisface las condiciones de equilibrio (6.5). Para ello aplicaremos las condiciones de optimalidad de KKT al problema LLP. Este problema es convexo y linealmente restringido, por tanto las condiciones de KKT son necesarias y suficientes.

Las condiciones KKT son necesarias y suficientes si demostramos que la función objetivo de LLP($\mathbf{x}$) es convexa. Solamente el término $G(\mathbf{g})$ de la función objetivo, asociado a la desagregación de la demanda, puede ser no convexo. Veamos que cada uno de sus sumandos es convexo y por tanto $G(\mathbf{g})$ también lo será.

El término que involucra a las variables $g^{a_s}_{\omega,t}$ es suma de funciones de entropía multiplicadas por la constante $1/\beta_3$. Si asumimos que $\beta_3 > 0$ este término será convexo.

Consideremos el siguiente término para un par ω

$$(1/\beta_2)g^a_{\omega,t}(\ln g^a_{\omega,t} - 1 + \alpha_t) - (1/\beta_3)g^a_{\omega,t}(\ln g^a_{\omega,t} - 1), \tag{6.18}$$

que puede reescribirse por

$$[(1/\beta_2) - (1/\beta_3)]g^a_{\omega,t}(\ln g^a_{\omega,t} - 1) + (1/\beta_2)g^a_{\omega,t}\alpha_t. \tag{6.19}$$

Como la función entropía $g^a_{\omega,t} \ln g^a_{\omega,t}$ es convexa, el término $g^a_{\omega,t}\alpha_t$ es lineal y suponiendo que

$$\beta_2 > 0,\ \beta_3 > 0 \text{ y } \beta_3 > \beta_2 \tag{6.20}$$

entonces obtenemos que todos estos términos son covexos y por tanto su suma

$$(1/\beta_2) \sum_{\omega\in W} \sum_{t\in I_\omega} g^a_{\omega,t}(\ln g^a_{\omega,t} - 1 + \alpha_t) - (1/\beta_3) \sum_{\omega\in W} \sum_{t\in I_\omega} g^a_{\omega,t}(\ln g^a_{\omega,t} - 1)$$

también será una función convexa.

Empleando un argumento similar se prueba que si se cumple

$$\beta_1 > 0,\ \beta_2 > 0 \text{ y } \beta_2 > \beta_1 \tag{6.21}$$

entonces la función

$$(1/\beta_1) \sum_{k\in\{a,b,c\}} \sum_{\omega\in W} g^k_\omega(\ln g^k_\omega - 1 + \alpha^k) - (1/\beta_2) \sum_{\omega\in W} g^a_\omega(\ln g^a_\omega - 1)$$

es convexa.

Tras estas consideracionesm asumiremos que se cumplen las condiciones (6.20) y (6.21) y por tanto la función objetivo de LLP($\mathbf{x}$) es convexa.

La función Lagrangiana del problema LLP es

$$\mathcal{L} = T + \sum_{\omega\in W} \lambda_\omega \left(\sum_{k\in\{a,b,c,d\}} g^k_\omega - \bar{g}_\omega \right) + \sum_{\omega\in W} \lambda^a_\omega \left(\sum_{t\in I_\omega} g^a_{\omega,t} - g^a_\omega \right) +$$
$$\sum_{t\in I, s\in\{1,2\}} \lambda^s_t \left(\sum_{\omega\in W_t} g^{a_s}_{\omega,t} - g^{a_s}_t \right) + \sum_{t\in I} \sum_{\omega\in W_t} \lambda^a_{\omega,t} \left(g^{a_1}_{\omega,t} + g^{a_2}_{\omega,t} - g^a_{\omega,t} \right)$$

Primeramente calcularemos las derivadas parciales de la función Lagrangiana respecto a las variables primales

$$\left\{ g^a_\omega,\ g^b_\omega,\ g^c_\omega,\ g^d_\omega,\ g^a_{\omega,t},\ g^{a_s}_{\omega,t},\ g^{a_s}_t \right\}$$

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial g_\omega^k} &= (1/\beta_1)\left(\log g_\omega^k + \alpha^k\right) \\
&+ U_\omega^{k*} + \lambda_\omega, \quad k \in \{b,\ c,\ d\},\ \omega \in W \qquad (6.22)\\
\frac{\partial \mathcal{L}}{\partial g_\omega^a} &= (1/\beta_1)\left(\log g_\omega^a + \alpha^a\right) - (1/\beta_2)\log g_\omega^a \\
&+ \lambda_\omega - \lambda_\omega^a,\ \omega \in W \qquad (6.23)\\
\frac{\partial \mathcal{L}}{\partial g_{\omega,t}^a} &= (1/\beta_2)\left(\log g_{\omega,t}^a + \alpha_t\right) - (1/\beta_3)\log g_{\omega,t}^a \\
&- \lambda_{\omega,t}^a + \lambda_\omega^a,\ t \in I, \omega \in W_t \qquad (6.24)\\
\frac{\partial \mathcal{L}}{\partial g_{\omega,t}^{a_s}} &= (1/\beta_3)\left(\log g_{\omega,t}^{a_s} + \alpha_t^s\right) + \bar{U}_{\omega,t}^{a_s *} \\
&+ \lambda_t^s + \lambda_{\omega,t}^a,\ t \in I, \omega \in W_t, s \in \{1,\ 2\} \qquad (6.25)\\
\frac{\partial \mathcal{L}}{\partial g_t^{a_s}} &= \frac{c_t^s(g_t^s)}{\tau} - \lambda_t^s \quad t \in I, s \in \{1,\ 2\} \qquad (6.26)
\end{aligned}$$

Empleando la condición de estacionariedad del Lagrangiano, obtenemos que $\frac{\partial \mathcal{L}}{\partial g} = \mathbf{0}$. Usando esta condición, de (6.26) obtenemos

$$\lambda_t^s = \frac{c_t^s(g_t^s)}{\tau} \qquad (6.27)$$

Empleando (6.27) para sustituir λ_t^s en (6.25) y usando (6.9), obtenemos que

$$\log g_{\omega,t}^{a_s} = -\beta_3 U_{\omega,t}^{a_s *} - \alpha_t^s - \beta_3 \lambda_{\omega,t}^a$$

y obtenemos

$$g_{\omega,t}^{a_s} = \exp\left(-\{\alpha_t^s + \beta_3 U_{\omega,t}^{a_s *}\}\right)\exp(-\beta_3 \lambda_{\omega,t}^a) \qquad (6.28)$$

donde $\alpha_t^s + \beta_3 U_{\omega,t}^{a_s *}$ es el coste de transporte de la alternativa *park'n ride* para el par ω, empleando el intercambiador t y utilizando un aparcamiento tipo s. Empleando la relación (6.28) es fácil ver que las proporciones de viajes en el par ω, que emplean el intercambiador t y el tipo de aparcamientos s se obtienen por la relación

$$\frac{g_{\omega,t}^{a_s}}{\sum_{s' \in \{1,2\}} g_{\omega,t}^{a'_s}} = \frac{\exp\{-\left(\alpha^s{}_t + \beta_3 U_{\omega,t}^{a_s *}\right)\}}{\sum_{s' \in \{1,2\}} \exp\{-(\alpha_t^{s'} + \beta_3 U_{\omega,t}^{a_{s'} *})\}},$$

Utilizando la relación $g_{\omega,t}^{a_1} + g_{\omega,t}^{a_2} = g_{\omega,t}^a$ y (6.28)

$$g_{\omega,t}^a = \sum_{s \in \{1,2\}} \exp\left(-\{\alpha_t^s + \beta_3 U_{\omega,t}^{a_s *}\}\right)\exp(-\beta_3 \lambda_{\omega,t}^a) \qquad (6.29)$$

Tomando logaritmos en ambos lados de (6.29) se consigue

$$\lambda_{\omega,t}^a = \frac{-1}{\beta_3}\log g_{\omega,t}^a - L_{\omega,t}^a \qquad (6.30)$$

donde $L_{\omega,t}^a$ se calcula como el "log-suma" de los costes de transporte en cada subalternativa de $g_{\omega,t}^a$, es decir,

$$L_{\omega,t}^a = \frac{-1}{\beta_3}\log\left(\sum_{s \in \{1,2\}} \exp\left(-\{\alpha_t^s + \beta_3 U_{\omega,t}^{a_s *}\}\right)\right)$$

Sustituyendo (6.30) en (6.24), obtenemos

$$\frac{1}{\beta_2}(\log g^a_{\omega,t} + \alpha_t) + L^a_{\omega,t} + \lambda^a_\omega = 0$$

Despejando $g^a_{\omega,t}$ de la anterior relación

$$g^a_{\omega,t} = \exp\left(-\{\alpha_t + \beta_2 L^a_{\omega,t}\}\right)\exp(-\beta_2\lambda^a_\omega) \tag{6.31}$$

La proporción de viajeros de tipo *park'n ride* para el par ω que emplean el intercambiador t es

$$\frac{g^a_{\omega,t}}{\sum_{t'\in I_\omega} g^a_{\omega,t'}} = \frac{\exp\{-\left(\alpha_t + \beta_2 L^a_{\omega,t}\right)\}}{\sum_{t'\in I_\omega}\exp\{-(\alpha_t + \beta_2 L^a_{\omega,t'})\}},$$

Empleando (6.12) y (6.31), obtenemos

$$g^a_\omega = \sum_{t\in I_\omega}\exp\left(-\{\alpha_t + \beta_2 L^a_{\omega,t}\}\right)\exp(-\beta_2\lambda^a_\omega) \tag{6.32}$$

y despejando λ^a_ω de la anterior relación

$$\lambda^a_\omega = \frac{-1}{\beta_2}\log g^a_\omega - L^a_\omega \tag{6.33}$$

donde L^a_ω viene dada por (6.3). Sustituyendo λ^a_ω en (6.23), obtenemos

$$\frac{1}{\beta_1}\left(\log g^a_\omega + \alpha^a\right) + \lambda_\omega + L^a_\omega = 0$$

Sustituyendo g^a_ω en la relación anterior

$$g^a_\omega = \exp(-\{\alpha^a + \beta_1 L^a_\omega\})\exp(-\beta_1\lambda_\omega) \tag{6.34}$$

Despejando g^k_ω en (6.22), conseguimos

$$g^k_\omega = \exp(-\{\alpha^k + \beta_1 U^{k*}_\omega\})\exp(-\beta_1\lambda_\omega) \tag{6.35}$$

empleando (6.34) y (6.35), obtenemos que la partición modal de la demanda es

$$\frac{g^k_\omega}{\sum_{k'\in\{a,b,c,d\}} g^{k'}_\omega} = \frac{\exp-\left(\alpha^k + \beta_1 U^{k*}_\omega\right)}{\sum_{k'\in\{a,b,c,d\}}\exp-(\alpha^{k'} + \beta_1 U^{k'*}_\omega)},$$

donde $U^{a*}_\omega = L^a_\omega$ que es el "log-suma" de las utilidades $L^a_{\omega,t}$, donde $t \in I_\omega$.

Empleando (6.11) y (6.24), obtenemos el multiplicador λ_ω

$$\lambda_\omega = \frac{-1}{\beta_1}\log\bar{g}_\omega + \frac{1}{\beta_1}\log\left(\sum_{k'\in\{a,b,c,d\}}\exp-(\alpha^{k'} + \beta_1 U^{k'*}_\omega)\right) \tag{6.36}$$

y si sustituimos las expresiones (6.30), (6.33) y (6.36) de los multiplicadores en (6.35), (6.32) y (6.28) obtenemos las condiciones de equilibrio (6.5).

Capítulo 7

Conclusiones, aportaciones y futuras líneas de investigación

Este capítulo recoge las aportaciones y conclusiones que se han obtenido en los capítulos anteriores. También aborda los puntos que requieren de una mayor profundización y que serán tratados como nuevas líneas de investigación en el futuro.

7.1 Conclusiones y aportaciones

Cada capítulo de la tesis se centra en un problema determinado, estando relacionado con el resto de los capítulos, pero abordado de forma autocontenida, lo que los hace independientes en gran medida. Es por este motivo por lo que en esta sección analizamos capítulo a capítulo cuáles han sido las conclusiones y aportaciones de cada uno de ellos.

El capítulo *Introducción y sumario* está destinado a introducir los temas de referencia para este trabajo y no contiene ninguna aportación original.

§1. Modelos de equilibrio con modos combinados

Este capítulo presenta un modelo de asignación en equilibrio para viajes con modos combinados en el que los usuarios eligen explícitamente la ruta, el modo de transporte y el nodo de transferencia. Este modelo, denominado TAP- M, puede ser considerado una extensión del modelo desarrollado en Fernández y otros [73] para costes simétricos (apéndice III del capítulo 1) al caso asimétrico. Esta extensión ha requerido emplear una formulación mediante desigualdades variacionales en el espacio de flujos en los hipercaminos.

Además, en este trabajo se han desarrollado dos algoritmos de generación de columnas / descomposición simplicial desagregada para el modelo en desigualdades variacionales, formulado en el espacio de flujo en los hipercaminos. El primer algoritmo, genera las columnas mediante un subproblema de tipo Frank-Wolfe y el segundo mediante un subproblema tipo Evans. Este tipo de algoritmos es nuevo en el contexto de desigualdades variacionales, habiéndose sido aplicados en el contexto de modelos de optimización. También se ha analizado el caso especial de que el modelo pueda ser reformulado en el espacio de flujo en los arcos, mostrando que el RMP de estos dos algoritmos es equivalente al RSD aplicado al TAP.

Se han realizado pruebas computacionales para el caso simétrico, mostrando que el algoritmo con generación de columnas tipo linealización parcial de Evans tiene mejores propiedades de convergencia.

La primera contribución es la metodología utilizada en la resolución del TAP-M ya que, aunque se ha empleado la descomposición simplicial en un contexto de desigualdades variacionales, no se ha

aplicado a modelos inelásticos o combinados.

La segunda contribución de este capítulo es la aplicación del modelo TAP-M al diseño paramétrico de intercambiadores multimodales urbanos. Es por esto, por lo que se han desarrollado pruebas numéricas orientadas a ilustrar el uso del modelo para diseñar los llamados factores *hard*, como precios, capacidades, distancias, etc. y los llamados factores *soft*, como la seguridad, tiendas en el intercambiador, zonas de espera, etc. mediante la modificación de los parámetros del modelo de demanda logit. Este diseño lo hemos denominado paramétrico, para reflejar cómo el operador del sistema va introduciendo las políticas mediante la variación de la parametrización de la red. En los capítulos 5 y 6 se ha empleado una metodología binivel para que las políticas del operador sean generadas automáticamente por el modelo.

§2. La clase de algoritmos CG/SD en optimización convexa diferenciable: análisis de la convergencia

En este capítulo se desarrolla una clase de algoritmos de generación de columnas /descomposición simplicial (CG/SD) para resolver el problema de programación matemática convexa diferenciable. La clase CG/SD constituye una generalización de los algoritmos de descomposición simplicial no lineal (NSD) de Larsson y otros [154, 141].

La primera diferencia entre las clases NSD y CG/SD radica en el principio de generación de columnas. El NSD obtiene las columnas como solución (truncada) a una aproximación cuadrática del problema original, mientras que en el CG/SD se obtienen mediante la aplicación de varias iteraciones de un algoritmo cerrado y de descenso (Zangwill [248]) a una función de mérito, que puede ser la propia función objetivo. La clase CG/SD reemplaza los subproblemas CGP del NSD por algoritmos cerrados de descenso en el proceso de generación de columnas. La clase NSD es un caso particular de la CG/SD, donde los algoritmos están definidos a través de los subproblemas CGP y solamente se realiza una iteración para obtener la columna.

La segunda diferencia radica en la gran libertad en la definición de la región factible del RMP. La clase CG/SD emplea conjuntos (compactos) convexos, mientras que en los algoritmos de descomposición simplicial son conjuntos poliedrales.

La principal contribución del capítulo es la formulación de la clase CG/SD y el establecimiento de su convergencia asintótica.

La clase CG/SD (como se pondrá en evidencia en el capítulo 3) se puede interpretar como una forma de acelerar la convergencia de un algoritmo de optimización mediante un esquema de descomposición simplicial. En este contexto, se puede plantear qué propiedades del algoritmo de optimización empleado en la fase CGP son heredadas por el algoritmo simplicial.

La segunda contribución significativa del capítulo es el estudio de la transmisión de las propiedades de identificación de la cara óptima (restricciones) y de la convergencia finita. El estudio de estas dos cuestiones lleva a un análisis preliminar de las condiciones de regularidad del problema (cualificación de restricciones) y al estudio de la geometría de las soluciones óptimas que permitan garantizar la transmisión . El otro elemento clave de este proceso es la definición (reglas de generación/eliminación de columnas) y resolución del RMP. Bajo ciertas condiciones, que cubren todos estos aspectos, se ha dado una caracterización del problema de identificación de la cara óptima (restricciones). Se ha demostrado que la hipótesis de *mínimo débilmente puntiagudo* es una condición suficiente para garantizar la convergencia finita de un importante grupo de algoritmos CG/SD.

Se ha encontrado un algoritmo CG/SD y un problema determinado en el que no se produce convergencia finita. Esto pone de manifiesto que esta propiedad, que poseen los algoritmos SD y RSD, no la posee todos los algoritmos de la clase CG/SD.

Una contribución menor de este capítulo es la demostración, siguiendo las ideas de Hearn y otros [123], de que la región factible del RMP sigue siendo un simplex, bajo la hipótesis de que los problemas RMP son resueltos exactamente y empleando las reglas clásicas de introducción/eliminación de columnas.

§3. La clase de algoritmos CG/SD en optimización: estudio computacional

En este capítulo se hace un estudio computacional de la clase de algoritmos CG/SD empleando dos problemas de flujos en redes no lineales. El primero es el problema uniproducto SCNF y el segundo es el problema multiproducto TAP-M (versión simétrica).

Se ha analizado numéricamente los dos siguientes aspectos.

- ◇ Validación de la clase CG/SD como procedimiento para acelerar la convergencia de los algoritmos de direcciones factibles y para mejorar los algoritmos clásicos de descomposición simplicial.

- ◇ Estudio de los parámetros y del papel de la prolongación de las columnas a la frontera relativa en la eficiencia de los algoritmos CG/SD.

La clase CG/SD respecto a los algoritmos RSD y SD resuelve un menor número de problemas RMP y son de menor tamaño, lo que conlleva a una reducción significativa del coste computacional en la fase RMP, sin embargo, el número total de puntos extremos generados por el CG/SD no siempre es menor que el generado por el RSD o SD. Esta situación dependerá del problema en cuestión y de la precisión demandada en su solución y por tanto, el CG/SD no siempre mejora la fase CGP.

La principal mejora de la clase CG/SD respecto a la clase NSD radica en la forma de mejorar la calidad de las columnas generadas. En los métodos NSD un incremento de la calidad de las columnas sólo se puede conseguir mediante una mejor aproximación del problema original o incrementando la precisión en su resolución. Con la clase CG/SD se puede incrementar la calidad de las columnas aplicando mayor número de veces el algoritmo $\mathcal{A}_c$ en la fase CGP. Otra forma de interpretar esta mejora es ver el NSD como un caso particular de CG/SD donde $n_c = 1$ (número de veces que se aplica el algoritmo $\mathcal{A}_c$ para generar la columna).

La principal conclusión del estudio numérico de la fase de prolongación de las columnas a la frontera relativa, es que se pone de manifiesto que si no se efectúa dicha prolongación entonces la velocidad de convergencia del método CG/SD está monitorizada por el algoritmo empleado en la fase CGP. Por contra, si se efectúa la prolongación y $r \geq \dim F^* + 1$ la velocidad en la convergencia del algoritmo CG/SD está monitorizada por el algoritmo empleado en el RMP ($\mathcal{A}_r$). Este resultado, unido a que la clase de algoritmos CG/SD mantiene pequeño el número de variables en el RMP (y por tanto su complejidad computacional), permite la elaboración de algoritmos con tasa de convergencia superlineal (incluso cuadrática) para problemas de grandes dimensiones en un tiempo admisible de cálculo.

La contribución fundamental de este estudio numérico, es que se ha mostrado que la clase CG/SD mejora (en algunos casos significativamente) los algoritmos de descomposición simplicial RSD, SD y NSD. Esto conduce a que la clase CG/SD constituye el *Estado-del-Arte* para los métodos de resolución de problemas de flujos en redes de grandes dimensiones, como son, los modelos de asignación en equilibrio, y quizás, también lo sea para otro tipo de problemas de programación matemática convexa diferenciable.

Otra contribución del capítulo ha sido el estudio de los parámetros n_c, n_r y r y de sus interacciones. Se ha mostrado que, en general, es recomendable tomar $n_c > 1$, $r = \infty$ para columnas de calidad alta y que el parámetro n_r tiene una importancia menor para este tipo de columnas.

Un resultado importante es el método para calcular la prolongación a la frontera relativa para cierta clase de métodos de direcciones factibles, que unido a la redefinición de la clase CG/SD en un contexto de optimización (no necesariamente convexa diferenciable), permite considerar la clase CG/SD como un procedimiento para acelerar la convergencia de los algoritmos de direcciones factibles. Este resultado se muestra numéricamente en los problemas de prueba.

Otras aportaciones son: la elaboración de aproximaciones cuadráticas monótonas diferenciables para ser empleadas por la clase NSD, el desarrollo de una herramienta para actualizar dinámicamente el valor del parámetro n_c (número de veces que se aplica el algoritmo $\mathcal{A}_c$ para generar la columna) así como los indicios sobre la no idoneidad de los problemas RMP no linealmente restringidos.

§4. Calibración de parámetros y estimación de matrices O-D en modelos combinados: un modelo de programación matemática binivel

Este capítulo está dedicado al problema de calibrar los parámetros y estimar (actualizar) la matriz O-D en los modelos combinados de equilibrio. Estos problemas se tratan a través del estudio del modelo combinado TAP-M.

Inicialmente se ha analizado el problema de la calibración del TAP-M y se ha formulado mediante un modelo de programación matemática binivel. Se han caracterizado tres fuentes de sobrespecificación de los parámetros y se ha realizado una pequeña experiencia numérica para la elección de la métrica del nivel superior. Los mejores resultados computacionales se obtienen mediante el método de máxima verosimilitud.

Posteriormente se ha desarrollado una nueva metodología, basada en la programación matemática binivel, para abordar simultáneamente los problemas de calibración y de estimación de matrices O-D, para el modelo TAP-M. El nivel superior de este modelo, denominado CDAM, decide la combinación de los parámetros y de la matriz O-D de modo que el problema de asignación en equilibrio TAP-M reproduzca lo más fielmente posible toda la información que se dispone sobre observaciones de aforos, matrices desactualizadas, resultados de encuesta, etc.

Se ha demostrado la existencia de soluciones del CDAM bajo la hipótesis de que las métricas empleadas en el nivel superior son continuas. Se ha demostrado la existencia de soluciones para el problema CDAM. La demostración requiere una adecuada formulación de las condiciones de equilibrio (teorema 1.2.1) para poder estudiar la dependencia continua entre los flujos en equilibrio y los parámetros del modelo. Este resultado implica que se puede aplicar el CDAM, incluso cuando el conjunto de observaciones de aforos sea incompleto y/o inconsistentes, o si no se dispone de una matriz O-D o partición modal de referencia.

El CDAM generaliza el primer modelo desarrollado para la calibración del TAP-M y permite emplear información sobre flujos en la red multimodal y encuestas en el proceso de calibración.

El modelo propuesto combina la elección de ruta, de modo y de nodo de transferencia en el proceso de estimación de las matrices y calibración. Este modelo posee respecto a una metodología secuencial dos importantes ventajas:

- ◇ La bondad del ajuste entre el comportamiento observado y estimado por el TAP-M, es mayor en el modelo CDAM que en una metodología secuencial, debido a que el modelo elige entre todas las combinaciones de parámetros y matrices. Mientras que la metodología secuencial, fija uno de los valores y estima el otro, por tanto, el espacio factible en la estimación está más restringido. Además, se emplea toda la información, tanto para la estimación de la matriz como para la calibración de los parámetros. Esta afirmación se ilustra numéricamente mediante un pequeño ejemplo de prueba.
- ◇ El CDAM puede ser empleado en más situaciones que la metodología secuencial, debido a que la fase de calibración (en la metodología secuencial) requiere la realización de encuestas para determinar la partición modal y la distribución de los viajes combinados a través de los intercambiadores. El CDAM permite mayor flexibilidad y puede realizar el ajuste de los parámetros y de la matriz, basándose únicamente en los datos disponibles, por ejemplo, en la información sobre los aforos de la red multimodal, por tanto, es un procedimiento más económico.

La contribución más relevante de este capítulo (además de la formulación del citado modelo) es la elaboración de un marco para desarrollar algoritmos heurísticos para el CDAM. Se ha demostrado, bajo condiciones de no degeneración de los flujos en equilibrio de la solución óptima, que si el algoritmo converge en un número finito de iteraciones, la solución obtenida es un mínimo local del CDAM.

Esta metodología puede ser fácilmente extrapolable al problema particular de estimar las matrices O-D en redes de tráfico congestionadas (DAP), obteniendo como casos particulares de ésta los algoritmos propuestos en Yang [241].

La característica especial de esta clase, es ver el modelo de equilibrio como un procedimiento

para generar los caminos óptimos, siendo su flujo asignado por el nivel superior. Tradicionalmente la mayoría de los algoritmos aplicados al DAP fuerzan al nivel superior a que emplee los caminos óptimos del mismo modo como son usados en el modelo de equilibrio.

§5. Capacidad de aparcamientos y tarifación en viajes combinados: un problema de diseño de redes continuo

En este capítulo se ha analizado un nuevo problema de diseño continuo en redes multimodales con modos combinados (CNDP). Este problema aborda el diseño de aparcamientos disuasorios empleados en los viajes combinados. Se han considerado como variables de interés las tarifas de los aparcamientos y sus capacidades, suponiendo que la localización de los aparcamientos ya ha sido decidida.

Este problema ha sido formulado empleando la programación matemática binivel y el modelo se ha denominado NDP-M. En el nivel superior, se fija un plan de aparcamientos, definido por las variables capacidad y tarifa de los aparcamientos disuasorios. Los usuarios eligen ruta, modo de transporte y aparcamiento en el nivel inferior que está definido por el TAP-M.

El NDP-M asume restricciones presupuestarias y su objetivo es disminuir la congestión en una parte de la red de transporte (por ejemplo, en la red de tráfico). Este modelo puede considerarse un híbrido entre un problema puro de diseño de redes (determinar la capacidad de los aparcamientos) y un problema de tarifación de la congestión (tarifación de los aparcamientos).

En la red multimodal, los aparcamientos están representados por arcos y su función de congestión representa el coste generalizado de aparcamiento en función de la capacidad, número de usuarios, tarifa, distancia a la parada, etc. El problema NDP-M busca la parametrización adecuada de estas funciones (dos parámetros por cada aparcamiento).

Se han desarrollado dos formulaciones equivalentes del NDP-M. La primera, formulación estándar (denominada NDP-M($\mathbf{x}$)), usa directamente como variables de diseño las capacidades y tarifas de los aparcamientos y la segunda, formulación no-estándar (denominada NDP-M($\mathbf{y}$)), emplea como variables de diseño los costes de aparcamiento. Hemos evaluado computacionalmente ambas formulaciones y se ha comprobado que la formulación NDP-M($\mathbf{y}$) tiene ventajas computacionales sobre la formulación NDP-M($\mathbf{x}$) en los problemas de prueba empleados. Este resultado motiva la extensión de esta formulación a otros problemas de diseños de redes.

Friesz y otros en [89] aplicaron el algoritmo de simulado recocido (SAA) al NDP. Este trabajo motivó la utilización del SAA para resolver el NDP-M. Las diferencias esenciales entre ambas adaptaciones son:

- ◇ El NDP-M posee un conjunto de restricciones laterales sobre las variables de diseño. Éstas han sido tenidas en cuenta proyectando las soluciones candidatas en la región factible, definida por las restricciones laterales.
- ◇ En el NDP-M aparecen dos dificultades computacionales para actualizar la matriz de paso mediante la descomposición de Cholesky, que son causadas por la no convexidad del problema NDP-M y por las restricciones laterales. Estas patologías originan matrices de varianza-covarianza muestral no definidas positivas y por tanto, no es aplicable la descomposición original de Cholesky. Se han estudiado varias modificaciones para poder tratar estas situaciones y se han tratado estas cuestiones computacionalmente.

Una primera conclusión del estudio numérico realizado es que el SAA es computacionalmente intensivo, lo que hace que su aplicación se reduzca a problemas relevantes con un tamaño mediano o pequeño. Una segunda conclusión del estudio numérico, es que la convergencia del SAA depende significativamente del nivel de precisión empleado en la resolución del modelo TAP-M, lo que hace adecuado la clase CG/SD como método de resolución.

Además de la formulación no-estándar y del estudio del SAA, para resolver nuevos problemas de diseño de redes, la contribución fundamental del capítulo es la aplicación del NDP-M a la gestión de

aparcamientos (tanto disuasorios como localizados en centros urbanos).

Se pueden distinguir dos tipos de metodologías empleadas hasta el momento en los estudios de planificación de aparcamientos. El primer grupo se basa en modelos de localización, que determinan el número de aparcamientos y su situación y/o capacidades, atendiendo a la optimización de los parámetros: distancia andando y accesibilidad. Estos modelos no tienen en cuenta el comportamiento del usuario frente al diseño de las nuevas facilidades. La otra metodología se basa en la posibilidad de evaluar la demanda de aparcamientos (por ejemplo, mediante modelos de demanda desagregada, con modelos de equilibrio, etc.) en función de las variables de diseño. El planificador obtiene varias políticas, que contrasta en varios escenarios futuros y, de esta evaluación, obtiene el plan de intervención en el sistema.

La metodología desarrollada en el capítulo se fundamenta en la programación matemática binivel, presentando importantes avances respecto a la metodología tradicional. Respecto al primer tipo de modelos, los de localización, el NDP-M tiene en cuenta la reacción de los usuarios a las políticas del planificador. Respecto a la segunda metodología, los que evalúan la demanda en función del diseño, las distintas políticas se generan automáticamente por el modelo y el número de estas políticas que se analizan es considerablemente mayor. Esta metodología tiene un gran potencial de uso para el diseño de aparcamientos en centros urbanos, que permiten ensayar políticas de restricción de aparcamientos (temporal y/o espacial) junto a ampliaciones, y/o tarifas de uso de los aparcamientos.

§6. **Metodología para el diseño de intercambiadores multimodales urbanos**

En este capítulo se aborda el diseño de intercambiadores multimodales urbanos en un contexto de planificación estratégica.

Consideramos la gestión de un sistema de transporte público formado por dos redes de transporte. La red principal, definida por la red de cercanías y metro, ofrece viajes urbanos de larga distancia, mientras que la red secundaria, formada por líneas de autobuses locales, alimenta a las líneas principales. Se supone que se han creado nuevas líneas y/o se han ampliado algunas de las líneas existentes en la red principal y se desea localizar sobre ellas nuevas estaciones, es decir, el trazado lineal es conocido, pero no la localización de las estaciones. Se distinguen dos tipos de estaciones: las convencionales y las que disponen de facilidades adicionales, como aparcamientos disuasorios o que están alimentadas por la red secundaria, que se denominan intercambiadores multimodales urbanos. El problema que se plantea en este capítulo es determinar la localización de los intercambiadores/paradas, la capacidad y tarifas de los aparcamientos disuasorios y el diseño adecuado para la red secundaria de alimentación.

Este problema presenta una estructura de juego de Stackelberg, donde el planificador toma sus decisiones, conociendo la reacción de los usuarios a su propuesta de diseño de red de transporte, lo que conduce a una formulación binivel del problema. En el nivel superior se consideran las decisiones de localización de los intercambiadores, tipo de alimentación mediante una red de transporte secundaria y el diseño de facilidades de aparcamiento definidas por sus capacidades y tarifas. En el nivel inferior, LLP, los usuarios eligen el modo de transporte, intercambiador y tipo de aparcamiento para realizar su viaje.

La complejidad del problema, junto al horizonte temporal de la planificación (a largo plazo), ha llevado a un nuevo modelo de equilibrio multimodal con modos combinados entre oferta y demanda que define el LLP. Las diferencias respecto al TAP-M van en dos direcciones. La primera es un nuevo nivel en el modelo (de demanda) logit anidado, con el fin de recoger la elección que hacen los usuarios al aparcar en el intercambiador o fuera de él. La segunda diferencia está en el modelo de red de transporte (oferta de transporte). En este modelo no se tiene en cuenta cómo las variables de diseño afectan a la congestión, es decir, se considera un nivel de congestión independiente de las variables de diseño.

Se han desarrollado dos formulaciones del LLP: una mediante programación matemática y otra mediante una formulación de *tipo punto fijo*. Se ha demostrado la equivalencia entre ambas formulaciones. Para la formulación punto fijo se ha adaptado un algoritmo de tipo Gauss-Seidel para su resolución. Este algoritmo no tiene garantizada la convergencia y ésta es su mayor debilidad, no

obstante, en los ejemplos numéricos desarrollados converge y además de forma muy eficiente.

El diseño de intercambiadores es un problema de diseño de redes mixto, es decir, con variables discretas (localización de intercambiadores y diseño de la red secundaria) y continuas (precio y capacidades). Desde el punto de vista del horizonte temporal de la planificación, las variables discretas se sitúan en un contexto de planificación estratégica y las variables continuas en un contexto táctico.

La complejidad del modelo nos ha hecho centrarnos únicamente en la resolución del problema de diseño de redes discreto, es decir, en el problema de localizar los intercambiadores y elegir el diseño de la red de acceso para ciertos valores fijos de las variables tácticas. Este problema binivel no lineal y entero ha sido resuelto con cuatro algoritmos heurísticos. Dos de ellos basados en los algoritmos golosos, otro en la técnica de intercambio y el cuarto en la técnica del recocido simulado.

Los experimentos numéricos con estos algoritmos demuestran que esta metodología es computacionalmente intensa pero conduce a soluciones para problemas de mediano tamaño. También muestran que ninguno de los algoritmos heurísticos domina al resto.

La contribución principal radica en la aplicación del modelo a un problema importante de la planificación de transporte urbano, como es el de la expansión de la red de transporte público. Esta metodología transciende de la aplicación concreta al diseño de intercambiadores multimodales urbanos y puede ser aplicada a cualquier problema de diseño de redes de transporte. En un primer paso se construye un modelo de comportamiento del usuario, posteriormente un modelo de comportamiento del planificador del sistema de transporte y se integran mediante un modelo binivel que debe ser resuelto, por lo general, aproximadamente.

7.2 Futuras líneas de investigación

La realización de esta tesis doctoral constituye un aprendizaje de los temas relacionados con la investigación realizada y es punto de partida de nuevos proyectos.

La definición de qué líneas de investigación merecen ser continuadas requiere evaluar cuáles de las desarrolladas son significativas. Consideramos que las principales aportaciones de la tesis son:

- ◇ La clase de algoritmos CG/SD.
- ◇ El desarrollo de modelos matemáticos aplicados a problemas de transporte.
- ◇ Resolución de problemas de programación matemática binivel a gran escala.

A continuación analizamos, aportación por aportación, qué aspectos de los no tratados en este trabajo pueden ser estudiados en el futuro.

7.2.1 La clase de algoritmos CG/SD

En este trabajo se muestra que los algoritmos CG/SD constituyen el *Estado-del-Arte* en algunas aplicaciones. Esto es una motivación fundamental para seguir analizando esta clase de algoritmos. Son varias las líneas que consideramos de interés:

1. Aplicación a nuevos problemas estructurados.

   La clase CG/SD es un marco que permite la elaboración de nuevos algoritmos, resultando altamente interesante en problemas con estructuras especiales, como los problemas que se han abordado en la tesis (con estructura de red).

   Una línea de interés sería analizar nuevas situaciones importantes como, por ejemplo, el problema de asignación de tráfico, problemas con estructura de red y con restricciones laterales o incluso explotar la estructura de producto cartesiano de los problemas de flujo en redes multiproducto.

En este último caso, la gran flexibilidad de la definición de la región factible del RMP en el CG/SD nos permite considerar conjuntos de la forma $X^t = \prod_{\omega \in W} \texttt{conv}\,(X^t_\omega)$, donde X^t_ω son los flujos del producto ω retenidos en la interacción t. La elección del conjunto X^t es válida en el marco CG/SD porque es convexo, compacto y conteniendo la última iteración del CGP y RMP. Este caso se puede interpretar como una generalización del DSD de Larsson y Patriksson en [140], originando los algoritmos CG/SD desagregados.

2. Extensión al problema de desigualdades variacionales.

   Una prolongación de la investigación, es la extensión de la clase de algoritmos CG/SD al problema más general de desigualdades variacionales que se formula del siguiente modo: encontrar un $\mathbf{x}^* \in X$ cumpliendo

   $$\mathbf{F}(\mathbf{x}^*)^T(\mathbf{x} - \mathbf{x}^*) \geq 0, \qquad \forall \mathbf{x} \in X. \qquad [\text{VIP}(\mathbf{F}, X)]$$

   donde $\mathbf{F} : X \mapsto \Re^n$ es continua en X. Un primer paso sería el establecimiento de la convergencia asintótica (que es una extensión del teorema 2.7). Tal resultado requeriría de un detallado estudio para la elección adecuada de la función de mérito y de sus propiedades, que reemplazarían a la función f utilizada en el algoritmo. También se debería validar nuevas reglas de eliminación de columnas y analizar las propiedades de descenso del algoritmo. Algunos resultados en esta dirección han sido ya obtenidos en Larsson y otros [138] y Patriksson [198].

   Para poder extender los dos resultados fundamentales de identificación de la cara óptima y convergencia finita dados en los teoremas 2.4.16 y 2.4.20, deberíamos extender las definiciones usadas en el capítulo 2. La función ∇f se generaliza al contexto VIP$(\mathbf{F}, X)$ reemplazándola por la función $\mathbf{F}$. Esta identificación permitiría extensiones de la definición de *mínimos débilmente puntiagudos* (definición 2.4.18) o la posibilidad de transformar la condición $\{\nabla^X f(\hat{y}^t)\} \rightarrow \mathbf{0}$ a $\{P_{T_X(\hat{y}^t)}[-\mathbf{F}(\hat{y}^t)]\} \rightarrow \mathbf{0}$, necesarias para la identificación de la cara óptima.

3. Algoritmos CG/SD anidados.

   Una propiedad importante de la clase CG/SD, es que cualquier algoritmo CG/SD puede ser empleado como método de generación de columnas en la fase CGP, generando los denominados CG/SD anidados. Esta elección es posible debido a que los algoritmos CG/SD son factibles, de descenso y convergentes. Este hecho le confiere una cierta flexibilidad para la definición recursiva de nuevos algoritmos CG/SD. El mejor ejemplo ilustrativo de este hecho, analizado en la tesis para los problemas SNFP, utiliza el algoritmo RSD para generar las columnas, esto es, un esquema RSD dentro de otro esquema superior de RSD. Esta visión origina el nombre de *CG/SD anidado*.

   El nivel de anidamiento de los ejemplos analizados en la tesis es dos, pero podríamos seguir este proceso y utilizar estos RSD anidados como procedimientos de generación de columnas, obteniendo un tercer nivel de anidamiento y así sucesivamente. Cada nuevo nivel de anidamiento produce una mejora significativa de la calidad de las columnas generadas y reduce el número y tamaño de los problemas maestros, pero el coste computacional para la obtención de las columnas se ve incrementado. Estas consideraciones y el hecho de que el coste computacional del RMP para dos niveles de anidamiento es pequeño, hace que, a priori, no sea de interés un tercer nivel de anidamiento.

   Los comentarios anteriores limitan el interés de la exploración de mayores niveles de anidamiento en general, pero un caso donde el procedimiento de anidamiento puede tener ventajas computacionales es utilizarlo como método de aceleración de la convergencia de algoritmos de búsquedas lineales. En estos algoritmos, todos los parámetros que definen los distintos RMP en los diferentes niveles valen 1. Esta línea de investigación es muy concreta y se basa en evaluar numéricamente el número de búsquedas lineales en función del número de niveles de anidamiento para diversos algoritmos de direcciones factibles.

4. CG/SD en optimización.

   En el capítulo 3 se formula la clase CG/SD para algoritmos convergentes a través de puntos factibles. Esta formulación tiene dos importantes ventajas, por un lado, permite la extensión de

la clase CG/SD a problemas no convexos y/o no diferenciables y por otro, la interpretación de la clase CG/SD como un procedimiento de aceleración de algoritmos ya existentes.

En la versión estándar, la definida en el capítulo 2, se emplean las propiedades de convexidad y diferenciabilidad de las funciones que definen el CDP(f, X) para garantizar la convergencia del algoritmo empleado en el CGP. Si éste fuese convergente, no necesariamente para un problema de programación matemática convexa diferenciable, el algoritmo resultante (eso creemos) debe de seguir siendo convergente. El establecimiento de la convergencia asintótica de esta nueva clase de métodos constituye un campo de interés.

Posteriormente se plantearían nuevas condiciones y modificaciones de la clase CG/SD para poder emplear algoritmos que no usan necesariamente puntos factibles para los métodos de generación de columnas, como lo son la programación matemática secuencial, el algoritmo del Lagrangiano aumentado, los métodos de penalizaciones, etc.

7.2.2 El desarrollo de modelos matemáticos aplicados a problemas de transporte

La motivación última sobre la que descansan los algoritmos y métodos desarrollados en esta tesis doctoral, es la de resolver problemas reales que aparecen en la planificación del transporte urbano.

Su validación ha sido realizada sobre redes de prueba. Los experimentos han sido diseñados para evaluar las necesidades computacionales demandadas (tanto de los métodos como de los modelos), la robustez de las soluciones y los datos de entrada. Este análisis previo es una condición necesaria para su aplicabilidad, pero no constituye la validación definitiva de la metodología desarrollada. Es por este motivo, por lo que se requiere de un cambio cualitativo en la evaluación de la metodología y éste se debe fundamentar en la aplicación a problemas reales.

En la actualidad se está desarrollando el proyecto *Métodos para la estimación de la demanda. Redes jerárquicas urbanas multimodales* financiado por la Comisión Interministerial de Ciencia y Tecnología en el III Plan Nacional de Investigación Científica y Desarrollo Tecnológico, como proyecto de I+D en el área del Programa de Transportes.

En el citado proyecto participa el Consorcio de Transportes de la Comunidad de Madrid y algunas líneas de interés pasan por la aplicación a la red de metro y autobuses de la ciudad de Madrid de ciertas adaptaciones de los modelos desarrollados en esta tesis. Los más relevantes son:

1. Diseño de aparcamientos disuasorios.

   La red de metro de Madrid se está expandiendo fuera de la propia ciudad, uniendo poblaciones de la periferia. Un tema de interés de la Comunidad de Madrid es el diseño de aparcamientos disuasorios en las líneas periféricas de metro, con el fin de potenciar los viajes combinados y reducir la movilidad en vehículos privados dentro de la ciudad de Madrid.

   El modo de abordar este problema consistiría en modelar el corredor sudeste, definido por la N-III, la línea 9 de la red de metro de Madrid, las poblaciones de Rivas-Vaciamadrid y Arganda del Rey y las líneas de autobuses entre estas poblaciones y Madrid.

   El estudio se centraría en la obtención de las capacidades y tarifas de los aparcamientos disuasorios situados en las poblaciones de Rivas-Vaciamadrid y Arganda del Rey, teniendo en cuenta la competencia entre alternativas de transporte y su correspondiente partición modal de la demanda, para lo que se considerarían las características de las infraestructuras implicadas. Este problema requeriría una adaptación del modelo NDP-M (desarrollado en el capítulo 5).

2. Estimación de matrices O-D en la red de metro.

   En la actualidad la obtención de esta información es clave para planificar las frecuencias de los servicios y el diseño de nuevas líneas de transporte.

   El estudio de los servicios (a corto plazo) conduce al problema de actualizar la matriz de demanda O-D existente. Esta matriz se obtiene realizando encuestas en los andenes donde se pregunta a los usuarios el destino de su viaje y algunas características socioeconómicas.

Una motivación fundamental del CDAM (desarrollado en el capítulo 4) es que permite actualizar matrices O-D sobre observaciones de flujo en la red multimodal. Una línea de trabajo sería la adaptación de este modelo para la actualización/estimación de la matriz O-D en una red de transporte público, basada en sus sistemas propios de contajes, que ya no son los cordones para la medición de aforos de tráfico. Los dos sistemas disponibles en la red de metro son los pasos por torniquetes y los flujos en las secciones de línea. Esta última información se obtiene con personal situado en las paradas, que cuentan el número de viajeros que suben o bajan en la misma. Pese a todo, esta información tiene un coste económico apreciable, sobre unos treinta millones de pesetas, que hace que se realice cada cuatro años.

La importancia de esta aplicación no es sólo demostrar que el procedimiento descrito permite obtener económicamente una matriz de movilidad, sino compararla con la obtenida con los métodos tradicionales.

Los estudios de expansión de la red de transporte público requieren la estimación de la matriz O-D de los próximos años, para ello, se proyecta las características sociales y económicas de la población y mediante el modelo de las cuatro etapas se obtiene tal estimación. La determinación de esta matriz se realiza a partir de encuestas de movilidad general, realizadas cada década, debido a su alto coste, y mediante la realización de encuestas puntuales.

Este es un objetivo fundamental para el Consorcio Metropolitano pero cuya metodología difiere de la tratada en la tesis.

3. Expansión de la red de metro.

   La programación matemática binivel es adecuada para el diseño de redes, en concreto, permite el análisis de la expansión de la red de metro.

   Adaptaciones y simplificaciones del modelo de diseño BLPP (desarrollado en el capítulo 6) pueden ser aplicadas al problema de localizar las estaciones sobre el trazado, ya decidido, de una nueva línea de metro. Esta es la base para elaborar nuevos *modelos de localización* de estaciones donde se tuviese en cuenta la reacción de los usuarios al diseño de la red y el tiempo de viaje en la misma, en función de las estaciones localizadas. Estas ideas constituyen el incio para el desarrollo de nuevos modelos matemáticos aplicados a la expansión de la red de metro/cercanías.

7.2.3 Resolución de problemas de programación matemática binivel a gran escala

Una taxonomía de los modelos binivel, aplicados al transporte urbano, la constituyen los modelos de diseño de redes y de estimación de datos (parámetros y/o matrices). Esta clasificación obedece tanto a la finalidad de los modelos como a la estructura matemática de los mismos.

En este trabajo, hemos abordado dos problemas de diseño y un problema de estimación de datos basado en el modelo combinado TAP-M. Los algoritmos desarrollados para la resolución de estos modelos binivel son de naturaleza heurística, debido a su gran complejidad y sus grandes dimensiones. Las mejoras en las capacidades de cálculo de los nuevos ordenadores hace, quizás, que ya sea viable la elaboración de algoritmos exactos para estos modelos.

Los modelos binivel que emplean el TAP-M en el nivel inferior, poseen nuevas dificultades derivadas del modelo de demanda. Por este motivo, parece aconsejable trabajar, a la hora de abordar la obtención de algoritmos exactos, con los modelos clásicos de diseño de redes (NDP) y de estimación de matrices O-D (DAM), ya que trabajan con el problema TAP y éste tiene la demanda fija. Esto es, lo que se trata es de abordar más profundamente problemas que son menos difíciles de resolver.

En el capítulo 4 se ha elaborado una metodología para el desarrollo de algoritmos heurísticos para la resolución del CDAM. Esta clase constituye el punto de partida para el desarrollo de algoritmos exactos.

A continuación analizaremos los aspectos más relevantes para poder aplicar esta metodología a cada uno de estos problemas.

1. Resolución del problema de estimación de matrices

   La clase de algoritmos desarrollados para el CDAM es fácilmente exportable al problema DAM: basta con eliminar todo lo referente al modelo de demanda. La propia demostración del teorema 4.4.1 sigue siendo válida en este contexto simplificado y sirve para garantizar que, si la convergencia se obtiene en un número finito de iteraciones, entonces la solución obtenida es un óptimo local del DAM. Un aspecto que se debería analizar, son las condiciones que garanticen que la convergencia sea finita, y por tanto, que converja a un óptimo local.

   El fundamento de estos algoritmos estriba en la observación de que si se perturba la matriz O-D del problema TAP, en un cierto entorno y bajo ciertas condiciones, es suficiente utilizar los caminos en equilibrio de esta matriz (sin perturbar) para caracterizar la nueva situación de equilibrio asociada a la matriz perturbada. Esto significa que, para ciertos problemas, el conjunto de caminos en equilibrio de una matriz O-D caracteriza su entorno y, por tanto, se pueden describir las direcciones factibles de descenso y, como consecuencia los mínimos locales, en función de ellos. Esta es la base para restringir el DAM a este conjunto de caminos, originando un *problema restringido de estimación de matrices O-D.*

   La dificultad de este planteamiento radica en mantener el equilibrio en los caminos cuando se varía la matriz O-D. Los algoritmos heurísticos planteados relajan esta condición y no exigen que el flujo asignado a los actuales caminos deba estar en equilibrio. Por otro lado, si las observaciones de flujo están en equilibrio, el DAM restringido asignaría el flujo en estos caminos con el fin de recuperar la situación en equilibrio, es decir, asignándoles (si puede ser) un flujo en equilibrio. Esta observación es clave para minimizar las consecuencias adversas de la relajación anterior.

   El problema DAM no es convexo ni diferenciable, por lo que el estudio de los óptimos globales es una cuestión abierta. Utilizando la propiedad de convexidad de la función objetivo, quizás, se puedan dar condiciones suficientes que garanticen que el óptimo local obtenido sea un óptimo global del problema.

   El hecho de que, en la actualidad, el problema DAM no está satisfactoriamente resuelto hace de gran interés el estudio numérico del comportamiento de estos algoritmos en comparación con los algoritmos heurísticos que ya han sido empleados. Un buen comportamiento, definido en función del coste computacional o de la robustez del método, justificaría por si solo estos algoritmos.

   Es por este motivo que una línea de investigación prioritaria es la implementación eficiente de estos algoritmos. A priori, los métodos que requieren almacenar los caminos generados, como lo son los aquí expuestos, han sido criticados por el hecho de que el número de caminos en una red crece exponencialmente con el tamaño de la misma. La experiencia computacional desarrollada por Larsson y Patriksson con la descomposición simplicial desagregada (DSD) en [140] muestra, por contra, que el número de caminos en equilibrio para cada par no suele superar tres o cuatro caminos. Esta es una importante motivación para validar nuestra clase, ya que aunque el número total de caminos crezca exponencialmente, no es así para el número de caminos en equilibrio.

   Los algoritmos descritos en cada iteración resuelven dos problemas de optimización: el problema restringido de estimación de la matriz O-D y con esta matriz, un nuevo problema de equilibrio para determinar los caminos óptimos de la misma. El primer problema tiene la misma estructura que el RMP en el DSD y el segundo es un problema TAP. El algoritmo DSD parece adecuado para resolver el TAP debido a sus importantes propiedades de reoptimización, que serían aplicadas cada vez que se cambiase la matriz O-D, y al hecho de trabajar con los caminos en equilibrio de la red.

2. Resolución de problema de diseño de redes

   Esta clase de algoritmos también es aplicable al NDP. El punto de partida es establecer un resultado similar al teorema 4.4.1, pero para ello se requiere que el TAP presente un comportamiento, frente a las perturbaciones de los parámetros de los costes en los arcos, similar al de las perturbaciones de las matrices O-D. A priori, basándose en hipótesis de continuidad y no degeneración, parece que esto sea así.

   Otra ventaja del *problema restringido de estimación de matrices O-D*, que se mantendría en el correspondiente *problema restringido de diseño de redes,* es que los flujos se asignarían en

equilibrio pero aplicando el segundo principio de Wardrop. Este hecho puede conducir a buenos resultados como algoritmos heurístico pero, quizás, la convergencia no se pueda garantizar.

Notación y acrónimos

Subíndices y superíndices

- a modo coche privado
- b modo transporte público
- c modo combinado coche privado-transporte público
- d modo de transporte sin congestión (andando, bicicleta, etc.)
- i Origen
- j Destino
- k modo de transporte
- l Arco o línea de transporte publico
- n Contador de iteraciones de un algoritmo
- ℓ Contador de iteraciones de un algoritmo
- p Camino o hipercamino
- p_ω^ℓ Camino o hipercamino generado en la iteración ℓ que satisface el par de demanda O-D ω
- s Sección de ruta
- – Contador de iteraciones de un algoritmo iterativo
- t Nodo de transferencia
- – Aparcamiento
- – Cuando es un superíndice, indica la iteración $t-$ ésima
- ω Par origen- destino

Escalares

- a_t Tarifa del aparcamiento t
- $\bar{a}$ Número de caminos en la red de tráfico
- $\bar{b}$ Número de hipercaminos en la red de transporte público
- $\bar{c}$ Número de caminos en la red multimodal
- B Presupuesto del sistema de aparcamientos durante el período de planificación
- C Coste de operación del sistema de aparcamientos durante el período de planificación
- C_s Tiempo esperado de viaje en la sección de ruta
- D_j Número de usuarios atraídos por el origen j
- f^* Valor óptimo del CDP(f,X)
- g_ω^k Número de usuarios del par O-D ω que emplean el modo k
- $g_{\omega t}^{as}$ Número de usuarios del par O-D ω que emplean el modo *park'n ride* mediante el intercambiador t y eligen el aparcamiento tipo s

$g^c_{\omega t}$ Número de usuarios del par O-D ω que emplean el modo *park'n ride* mediante el nodo de transferencia t

g^k_ω Número de usuarios del par O-D ω que emplean el modo k

$\bar{g}_\omega$ Número total de usuarios del par O-D ω en todas las alternativas consideradas

$G^{a_s}_{\omega,t}$ Proporción de usuarios del par ω que emplean el aparcamiento tipo s en el intercambiador t respecto al total de usuarios de modo *park'n ride* en el par ω a través de dicho intercambiador t

$G^c_{\omega,t}$ Proporción de usuarios del par ω en modo *park'n ride* y via nodo de transferencia t respecto al número de usuarios de modo *park'n ride* para el par ω

G^k_ω Proporción de usuarios del par O-D ω que emplean el modo k respecto al total de usuarios del par

GAP^t Diferencia entre la cota superior e inferior del problema CDP en la itearción t

I Dinero obtenido por la tarifación de los aparcamientos disuasorios

K Constante del SAA para estandarizar la temperatura del proceso

k_l Capacidad del arco l

k_t Capacidad instalada en el aparcamiento t. Es una variable de diseño en el NDP-M

ℓ_t Prolongación a la frontera relativa en la iteración t de la columna generada

$M^{(n)}$ Número de soluciones aceptadas por le SAA en la $n-$ésima temperatura

n_c Número de iteraciones realizadas con el algoritmo $\mathcal{A}_\text{c}$ para generar la columna en el CGP

n^t_c Número de iteraciones realizadas con el algoritmo $\mathcal{A}_\text{c}$ en la iteración t para generar la columna en el CGP

n_r Número de iteraciones realizadas con el algoritmo $\mathcal{A}_\text{r}$ para aproximar la solución del RMP

n^t_r Número de iteraciones realizadas con el algoritmo $\mathcal{A}_\text{r}$ para aproximar la solución del RMP en la iteración t

NAC Máximo número de configuraciones aceptadas en cada temperatura por el SAA

NRC Máximo número de configuraciones rechazadas consecutivamente en cada temperatura por el SAA

O_i Número de usuarios generados en el origen i

r Parámetro de restricción del RMP en el algoritmo CG/SD

$\tilde{r}$ Parámetro de restricción del RMP en un algoritmo RSD

t_l Tiempo de viaje en el arco l

$T_{\bar{A}_s}$ Tiempo (esperado) de viaje en la sección de ruta s

$\mathcal{T}$ Temperatura en el algoritmo de simulado recocido

$\mathcal{T}_f$ Temperatura final en el algoritmo de simulado recocido

$\mathcal{T}_o$ Temperatura inicial en el algoritmo de simulado recocido

U^{k*}_ω Coste de transporte en el equilibrio para el par O-D ω en el modo k

$U^{c*}_{\omega,t}$ Coste de transporte en el equilibrio para el par O-D ω en el modo *park'n ride* via nodo de transferencia t

v^s_l Número de usuarios en la sección de ruta s que emplean la línea l

V_s Número de usuarios en la sección de ruta s

$W_{\bar{A}_s}$ Tiempo de espera en la sección de ruta s

x^s_l Variable dicotómica que vale 1 si la línea l es atractiva para la sección de ruta s

α Parámetro de recoción del SAA

α^k Coeficientes del modelo logit anidado asociado a la elección de modo

α^c_t Coeficientes del modelo logit anidado asociado a la elección de nodo de transferencia

β Tiempo esperado de viaje en la sección de ruta

β_i Coeficientes del modelo logit anidado

$\delta^{\mathbf{f}}_{lp}$ Elemento de la matriz de incidencia arco-ruta

$\delta^{\bar{\mathbf{g}}}_{p\omega}$ Elemento de la matriz de incidencia demanda-ruta

γ Parámetro del método del gradiente proyectado de Goldstein-Levitin-Polyak

τ Tasa de ocupación vehicular empleada en el modelo BLP

ϕ_l Frecuencia de la línea l

π^s_l Probabilidad de un usuario de la sección s que emplee la línea l

χ_s "Factor de crecimiento" de la matriz de varianza-covarianza en el SAA

τ Tiempo del ciclo de un semáforo

θ_k Coeficientes de homogeneización de costes entre redes de transporte

Espacios

$\Re^n$ Espacio n-dimensional

$\Re^n_+$ Octante no negativo de $\Re^n$

Vectores

$\mathbf{a}$ Capacidad instalada en los aparcamientos disuasorios

$\mathbf{d}_j$ Dirección extrema de un conjunto convexo

$\mathbf{f}$ Vector de flujos en los arcos

$\hat{\mathbf{f}}$ Vector de flujos observados en los arcos

$\mathbf{f}_t(\bar{\mathbf{g}}, \Theta)$ Selección de la función multievaluada $\Phi_t(\bar{\mathbf{g}}, \Theta)$

$\mathbf{g}$ Vector de variables de demanda

$\hat{\mathbf{g}}$ Vector de demandas observadas

$\bar{\mathbf{g}}$ Vector de demandas $(\cdots, \bar{g}_\omega, \cdots)$

$\mathbf{h}$ Vector de flujos en los caminos

$\mathbf{k}$ Vector de capacidades instaladas en los aparcamientos

$\mathbf{p}_i$ Punto extremo de un conjunto convexo

$\mathbf{u}$ Capacidades de los aparcamientos en el modelo de diseño de intercambiadores BLP

$\mathbf{U}^*_\omega$ Vector de costes de transporte en equilibrio para el par ω en todos los modos presentes, $(U^a_\omega, U^b_\omega \cdots)$

$\mathbf{U}^{c*}_\omega$ Vector de costes de transporte en equilibrio para el par ω en el modo *park'n ride* y todos los nodos de transferencia posibles, $(U^c_{\omega t_1}, U^c_{\omega t_2}, \cdots)$

$\mathbf{v}$ Variable de diseño genérica del NDP-M y puede representar la variable de diseño $\mathbf{x}$ de la formulación estándar o la variable $\mathbf{y}$ de la no-estándar

– Variable tarifas de los aparcamientos en el modelo de diseño BLP

$\mathbf{v}_{\mathrm{LB}}$ Cota inferior de la variable de diseño genérica del NDP-M

$\mathbf{v}_{\mathrm{new}}$ Solución candidata en el NDP-M generada por el SAA

$\mathbf{v}_{\mathrm{old}}$ Solución actual en el NDP-M

$\mathbf{x}$ Variable de diseño del NDP-M y representa el vector de tarifas y el de capacidad instalada en los aparcamientos

$\mathbf{x}^t$ Solución del RMP en la iteración t

$\mathbf{y}$ Variable de diseño del NDP-N que representa el coste generalizado de aparcamiento

– Variable de diseño del BLP que representa la localización de los intercambiadores

$\mathbf{y}^t$ Extensión a la frontera relativa de la columna generada en el CGP en la iteración t, esto es $\hat{\mathbf{y}}^t$

$\hat{\mathbf{y}}^t$ Columna generada en el CGP en la iteración t

$\mathbf{y}_{\neq i}$ Vector obtenido al eliminar de $\mathbf{y}$ la componente i- ésima

$\mathbf{z}$ Variable del BLP que representa el diseño de la alimentación de los intercambiadores

β Vector de tasas en los arcos de red

Θ Vector de parámetros del modelo TAP-M

ρ Proporciones de tiempo en verde en los controles semafóricos de las intersecciones

Funciones

$\mathcal{A}_{\mathrm{c}}^k$ Algoritmo empleado en la resolución del CGP

$\mathcal{A}_{\mathrm{r}}^k$ Algoritmo empleado en la resolución del RMP

$\mathbf{c}(\mathbf{f})$ Coste de viaje los arcos de la red en función del flujo en los arcos

$B(\mathbf{g}, \mathbf{v})$ Beneficio de la red de transporte público

$B_t(\mathbf{g})$ Beneficio de la red de transporte público producido por el intercambiador t en función de la demanda $\mathbf{g}$

$c_t^s(g_t^s)$ Coste de aparcamiento en el intercambiador t para el tipo de estacionamiento s en función del número de vehículos g_t^s

$C_p(\mathbf{h})$ Coste de viaje en el camino (hipercamino) p en función del flujo en todos los caminos (hipercaminos)

$\mathbf{C}(\mathbf{h})$ Coste de viaje en los caminos en función de su flujo

$\mathbf{C}_R(z_t)$ Coste del diseño z_t para alimentar el intercambiador t

$\mathbf{F}(\mathbf{x})$ Función genérica de costes en una desigualdad variacional

$F_i(\mathbf{x}, \mathbf{y})$ Métrica en $\Re^n$. Es la distancia entre los puntos $\mathbf{x}$ e $\mathbf{y}$

$G_\omega(U_\omega)$ Demanda en el par ω en función del coste de transporte U_ω

$G_\omega^k(\mathbf{U}_\omega^*)$ Porcentaje de usuarios en la alternativa k para el par ω en función del coste de transporte en equilibrio para las distintas alternativas $\mathbf{U}_\omega^*$

$G_{\omega,t}^c(\mathbf{U}_\omega^{c*})$ Porcentaje de usuarios de la alternativa c que emplean el nodo de transferencia t para el par ω en función del coste de transporte en equilibrio para los distintos nodos de transferencia $\mathbf{U}_\omega^{c*}$

$\mathcal{L}$ Función Lagrangiana

$L(\mathbf{y})$ Costes de localización de los intercambiadores

$p(x)$ Función de disuasión en los modelos de distribución de viajes

$P_S(\mathbf{x})$ Proyección euclídea de $\mathbf{x}$ dentro del conjunto convexo S

$p(\mathbf{u})$ Coste de instalación y gestión de los aparcamientos disuasorios en función de la capacidad instalada $\mathbf{u}$

$R(\mathbf{g})$ Integral de la función inversa de un modelo de demanda logit anidado

$R(\mathbf{g}, \Theta)$ Integral de la función inversa de un modelo de demanda logit anidado en función del vector de parámetros Θ y de la demanda $\mathbf{g}$

$R(\mathbf{y}, \mathbf{z})$ Costes del diseño de la red secundaria

$s(\mathbf{x})$ Restricción presupuestaria

$s_t(k_t)$ Coste de inversión en función de la capacidad de aparcamiento intalada k_t

$S(\mathbf{f}, \Theta)$ Integral de las funciones de coste en los arcos en función del vector de parámetros Θ y del flujo en los arcos $\mathbf{f}$

$T(\mathbf{g}, \mathbf{x})$ Función objetivo del LLP($\mathbf{x}$ parametrizada por $\mathbf{x}$)

$Z(\mathbf{f}, \mathbf{g})$ Función objetivo del TAP-M simétrico en función de los flujos en los arcos $\mathbf{f}$ y de la demanda $\mathbf{g}$

$Z(\mathbf{y})$ Función objetivo del NDP-M en función de la variable de diseño $\mathbf{y}$

$\Lambda(\mathbf{g})$ Función inversa de la demanda de un modelo logit anidado

$\Pi(\mathbf{x}, \mathbf{y})$ Función de mérito empleada en el CGP

$\Phi(\mathbf{U}, \mathbf{g})$ Función que representa la desagregación de la demanda potencial mediante un modelo logit anidado en función de los costes de transporte y de la propia demanda

$\Phi(\mathbf{x})$ Aproximación de la función de costes de una desigualdad variacional

$(\Phi_t(\bar{\mathbf{g}}, \Theta), \Psi_t(\bar{\mathbf{g}}, \Theta))$ Funciones respuesta del modelo TAP-M(t) que determinan el conjunto de flujos y demandas $(\mathbf{f}, \mathbf{g})$ para cada par $(\bar{\mathbf{g}}, \Theta)$

$\Psi(\mathbf{x}, \mathbf{g})$ Función objetivo del BLP

$\bar{\Psi}(\mathbf{x}, \mathbf{g})$ Función objetivo del BLP'

$\Gamma(\mathbf{y})$ Coste de inversión óptimo en el sistema de aparcamientos en función de diseño $\mathbf{y}$. Es un sumando de la función objetivo del NDP-M $Z(\mathbf{y})$

$\Gamma(\mathbf{g}, \mathbf{x})$ Desagregación de la demanda en el modelo con modos combinados LLP($\mathbf{x}$) en función de la propia demanda $\mathbf{g}$

$\Gamma_\omega(\mathbf{g}, \mathbf{x})$ Desagregación de la demanda ω en el modelo con modos combinados LLP($\mathbf{x}$) en función de la propia demanda $\mathbf{g}$

$\theta_i(\mathbf{y})$ Función de pago del jugador i en el juego de Nash o del seguidor i en el juego de Stackelberg

Matrices

$\mathtt{diag}\, A$ Es la matriz diagonal de A

$\mathbf{E}$ Matriz de incidencia nodo-arco

$\bar{\mathbf{g}}$ Matriz de pares de demanda O-D

$\mathbf{Q}$ Matriz de paso en el SAA

$\mathbf{s}^{(n+1)}$ La matriz de varianza-covarianza para la $(n+1)-$ésima temperatura del SAA

$\delta^{\mathbf{f}}$ Matriz de incidencia ruta-arco

$\delta^{a\mathbf{f}}$ Matriz de incidencia ruta-arco en la red de tráfico

$\delta^{\mathbf{g}}$ Matriz de incidencia ruta-desagregación de la demanda

$\delta^{\bar{\mathbf{g}}}$ Matriz de incidencia ruta-demanda total

Conjuntos y operaciones sobre conjuntos

$\mathtt{aff}\,(X)$ Envoltura afín del conjunto A

A Conjunto de arcos de tráfico en una red multimodal

A_s Conjunto de líneas de la sección de ruta s

$\bar{A}_s$ Conjunto de líneas comunes o atractivas de la sección de ruta s

$\mathcal{A}$ Conjunto de arcos de la red de transporte

$\hat{\mathcal{A}}$ Subconjunto de $\mathcal{A}$

B Conjunto de arcos de la red de transporte público de una red multimodal

$\mathrm{Cl}(X)$ Clausura del conjunto X

$\mathtt{dim}\, X$ Dimensión del conjunto X

$\mathcal{D}$ Conjunto de direcciones extremas de X

$E_X(\mathbf{d})$ Cara de X expuesta por el vector $\mathbf{d}$

$\mathcal{F}$ Conjunto de restricciones para los arcos de la red de transporte

F Cara de un conjunto convexo

F^* Cara óptima de CDP(f, X)

$F(\mathbf{x})$ Única cara del conjunto convexo X en la que $\mathbf{x} \in \mathtt{rint}\, F$

$\mathcal{G}$ Red genérica de transporte

$\mathcal{G}^a$ Red de tráfico en la red multimodal

$\mathcal{G}^b$ Red de transporte público

I Conjunto de orígenes

– Conjunto de intercambiadores

$\mathcal{I}(\mathbf{x})$ Conjunto de restricciones activas del problema CDP(f, X) en el punto $\mathbf{x}$

J Conjunto de destinos

$K_X(\mathbf{x})$ Cono $k-$tangente ver la definición 2.4.4 en la página 80

$\mathcal{K}_{\mathrm{c}}$ Colección finita de algoritmos empleados en el CGP

$\mathcal{K}_{\mathrm{r}}$ Colección finita de algoritmos empleados en el RMP

$\mathtt{lin}\, K$ Linealidad del cono K. Ver definición 2.4.5 en la página 80

$\mathcal{L}$ Conjunto de líneas de transporte público

$\mathcal{N}$ Conjunto de nodos en la red de transporte

N^k Conjunto de nodos en la red de transporte $\mathcal{G}^k$

$\mathcal{N}_k^+(\mathbf{y})$ $k-$entorno hacia delante del punto $\mathbf{y}$

$\mathcal{N}_k^-(\mathbf{y})$ $k-$entorno hacia atrás del punto $\mathbf{y}$

$N_X(\mathbf{x})$ Cono normal al conjunto X en el punto $\mathbf{x}$

$N_X(F)$ Cono normal a la cara F

P Conjunto de todos los caminos (hipercaminos) en la red de transporte

$\mathcal{P}$ Conjunto de columnas generadas por un algoritmo CG/SD

– Conjunto factible de controles

$\mathcal{P}_s^t$ Conjunto de columnas generadas por un algoritmo CG/SD y retenidas en la iteración t

$\mathcal{P}_x^t$ Este conjunto es vacío o contiene una columna que es la solución de algún RMP

P_ω^k Conjunto de caminos (o hipercaminos) para el par ω en la red $\mathcal{G}^k$

$P_\omega^{k*}(t)$ Conjunto de caminos óptimos del TAP-M para el par ω en el modo k y para el par $(\bar{\mathbf{g}}^t, \Theta^t)$

P^k Conjunto de todos los caminos (hipercaminos) en la red de transporte $\mathcal{G}^k$

$P_\omega^*(t)$ Conjunto de caminos óptimos del TAP-M para el par ω y para el par $(\bar{\mathbf{g}}^t, \Theta^t)$

$\mathcal{P}_\omega^{\ell-1}$ Conjunto de flujos en los hipercaminos de $\mathcal{Q}_\omega^\ell$

$\mathcal{Q}_\omega^\ell$ Conjunto de hipercaminos para el par O-D ω en la iteración $\ell + 1$

$\mathtt{rfro}\,(X)$ Frontera relativa del conjunto X

$\mathtt{rint}\,(X)$ Interior relativo de X

$\mathcal{S}_k^+(\mathbf{y})$ $k-$entorno hacia delante de $(\mathbf{y}, \mathbf{z})$ para cualquier $\mathbf{z}$

$\mathcal{S}_k^-(\mathbf{y})$ $k-$entorno hacia atrás de $(\mathbf{y}, \mathbf{z})$ para cualquier $\mathbf{z}$

$\mathcal{S}_k(\mathbf{y})$ $= \mathcal{S}_k^+(\mathbf{y}) \cup \mathcal{S}_k^-(\mathbf{y})$

$\mathrm{SOL}(f, X)$ Conjunto de soluciones óptimas de CDP(f, X)

T Conjunto de nodos de transferencia o de aparcamientos

T_ω Conjunto de nodos de transferencia o de aparcamientos empleados por el par O-D ω

$T_X(\mathbf{x})$ Cono tangente a X en el punto $\mathbf{x}$

V Proyección en el $\mathbf{y}-$espacio del conjunto de facitbilidad NDP-M$(\mathbf{x}, \mathbf{y})$

W Conjunto de pares de demanda origen-destino en una red de transporte genérica

W^k Conjunto de pares de demanda origen-destino en la red $\mathcal{G}^k$

W_t Conjunto de pares de demanda origen-destino que emplean el intercambiador t

X Región factible del problema de optimización

X^t Región factible del problema RMP en la iteración t

$\hat{X}$ Región factible del RMP

Y_i Conjunto de estrategia del jugador i en el juego de Nash o del seguidor i en el juego de Stackelberg

Ω Espacio factible de flujos en los hipercaminos en la red multimodal

$\Omega(t)$ Subconjunto de caminos del TAP-M que son óptimos para el par $(\bar{\mathbf{g}}^t, \Theta^t)$

$\tilde{\Omega}$ Región factible del TAP-M para costes simétricos

$\tilde{\Omega}(\bar{\mathbf{g}})$ Región factible del TAP-M para costes simétricos expresada en flujo en los arcos y desagregación de la demanda parametrizada por la matriz de demanda O-D $\bar{\mathbf{g}}$

$\tilde{\Omega}^*(\bar{\mathbf{g}})$ Conjunto de soluciones óptimas del TAP-M para la matriz de demanda O-D $\bar{\mathbf{g}}$, expresadas como flujo en los arcos y desagregación de la demanda

$\Omega_{\mathbf{f}}$ Espacio de flujos en arcos con demanda inelástica

$\Omega_{\mathbf{h}}$ Espacio de flujos en los caminos (hipercaminos) para el TAP-M

– Espacio de flujos en caminos (hipercaminos) con demanda inelástica

$\Omega_{\mathbf{h}}(\bar{\mathbf{g}})$ Espacio de flujos en los caminos para el TAP-M parametrizado por la matriz de demanda O-D $\bar{\mathbf{g}}$

$\Omega^*_{\mathbf{h}}(\bar{\mathbf{g}})$ Espacio de flujos óptimos en los caminos para el TAP-M y para una matriz de demanda O-D $\bar{\mathbf{g}}$

Ω^a Espacio factible de flujos en los caminos en la red de tráfico

$\Omega^a_{\mathbf{f}}$ Espacio de flujos en arcos para el problema de asignación de tráfico con demanda inelástica

Ω^b Espacio factible de flujos en los hipercaminos en la red de transporte público

$\Omega^{\mathbf{g}}_{\mathbf{f}}$ Región factible del TAP-M en el espacio de flujo en los arcos y desagregación de la demanda

– Espacio de flujos en arcos con demanda elástica

$\Omega^{\mathbf{g}}_{\mathbf{h}}$ Espacio de flujos en caminos (hipercaminos) con demanda elástica

$\tilde{\Omega}^\ell$ Región factible del RMP en la iteración ℓ formulada en el espacio de flujo en los arcos y demandas

Φ Conjunto de frecuencias admisibles

Operaciones

$\arg\min_{\mathbf{x}\in X} f(\mathbf{x})$ Conjunto de mínimos del problema $\min_{\mathbf{x}\in X} f(\mathbf{x})$

$q^{-1}(x)$ Función inversa de q

$\nabla f(\mathbf{x})$ Vector gradiente de f en $\mathbf{x}$

$\nabla^X f(\mathbf{x})$ Gradiente de f en $\mathbf{x}$ proyectado en X. Ver definición 2.16 en página 84

$\nabla^2 f(\mathbf{x})$ Matriz Hessiana de f en $\mathbf{x}$

$|\cdot|$ Cardinal de un conjunto

$\|\cdot\|$ Norma (euclídea) de un vector

$[\,\cdot]$ Parte entera de un número

Códigos de optimización

GRIDGEN Generador de redes de Bertsekas [19]

L2QUE Algorimo de caminos mínimos de Gallo y Pallotino [92]

NETGEN Generador de problemas de flujos en redes lineales de Klingman y otros [136]

RSDNET Algorimo RSD para redes uniproducto de Hearn y otros [125]

Problemas, algoritmos y otros acrónimos

$\mathcal{A}_{\mathrm{c}}$	Algoritmo genérico para la resolución del CGP
${\mathcal{A}_{\mathrm{c}}}_r^{n_r^t,n_c^t}$	Algoritmo CG/SD definido por el algoritmo proyectado de Newton para resolver el RMP y por el algoritmo $\mathcal{A}_{\mathrm{c}}$ en la fase CGP. El número de iteraciones realizadas por cada algoritmo en la iteración t es n_r^t y n_c^t respectiavamente. El parámetro de restricción es r.
$\mathcal{A}_{\mathrm{r}}$	Algoritmo genérico para la resolución del RMP
$(\{\mathcal{A}_{\mathrm{r}}\}_r^{n_r^t}, {\mathcal{A}_{\mathrm{c}}}^{n_c^t})$	Algoritmo CG/SD definido por el algoritmo $\mathcal{A}_{\mathrm{r}}$ para resolver el RMP y por el algoritmo $\mathcal{A}_{\mathrm{c}}$ en la fase CGP. El número de iteraciones realizadas por cada algoritmo en la iteración t es n_r^t y n_c^t respectiavamente. El parámetro de restricción es r.
BGA	Algoritmo goloso hacia atrás para el BLP'
BLP	Problema binivel para el diseño de intercambiadores multimodales urbanos
BLP'	Problema BLP donde las variables del nivel táctico han sido fijadas
CDP(f, X)	Problema de optimización convexa diferenciable definido por la función objetivo f y la región factible X
CDP$(\varphi(\cdot, \mathbf{x}), \nabla f, X, \mathbf{x})$	Subroblema de generación de columnas donde la función objetivo es viene definida por ∇f y por $\varphi(\cdot, \mathbf{x})$ en el punto $\mathbf{x}$ definida en la región factible X
CGP	Problema (o fase) de generación de columnas
CG/SD	Algoritmo de generación de columnas/descomposición simplicial
CS	Condición de complementariedad en las condiciones de KKT
DAM	Problema de estimación de matrices origen-destino
DSD	Algoritmo de descomposición simplicial desagregada
E	Algoritmo de Evans
FGA	Algoritmo goloso hacia delante para el BLP'
FW	Algoritmo de descomposición Frank-Wolfe
IA	Algoritmo de intercambio para el BLP'
KKT	Condiciones de optimalidad Karush-Kuhn-Tucker
LLP$(\mathbf{x})$	Problema del nivel inferior en uno de programación matemática binivel. Este problema está parametrizado por la variable $\mathbf{x}$
-	Modelo de equilibrio con modos combinados que define el nivel inferior del BLP
ME	Estimación basada en la maximización de la entropía
ML	Estimación máximo verosímil
MPEC	Problema de optimización con restricciones de equilibrio
MPEC-TAP	Problema de gestión de tráfico
NDP	Problema de diseño de redes continuo
NDP-TEAP	Problema de diseño de frecuencias en transporte público
NL	Modelo logit anidado
NLLS	Estimación mínimo cuadrática
NSD	Algoritmo de descomposición simplicial no lineal
PARTAN	Método de búsqueda unidimensional de las tangentes paralelas
P(f, X)	Problema de optimización definido por una función objetivo f, que es continua, y sobe la región factible X, que es compacta
RMP	Problema maestro restringido
RSD	Algoritmo de descomposición simplicial restringida
RSDCC	Algoritmo de descomposición simplicial restringida aplicado a problemas de optimización con restricciones convexas

SAA Algoritmo de simulado recocido

SD Algoritmo de descomposición simplicial

SNFP Problema de flujos en redes no lineales uniproducto

STP Problema estocástico de transporte

TAP Problema de asignación de tráfico con demanda inelástica

TAP-D Problema de asignación de tráfico y distribución

TAP-E Problema de asignación de tráfico con demanda elástica

TAP-E-VIP($\mathbf{c}, \Omega_{\mathbf{f}}^{\mathbf{g}}$) Formulación varicional en el espacido de los flujos en los arcos del problema de asignación de tráfico con demanda elástica

TAP-E-VIP($\mathbf{C}, \Omega_{\mathbf{h}}^{\mathbf{g}}$) Formulación varicional en en el espacido de los flujos en los caminos del problema de asignación de tráfico con demanda elástica

TAP-M Problema de asignación multimodal con modos combinados

TAP-M(t) Aproximación tipo Evans del TAP-M utilizada para aproximar el CDAM en la iteración t

TAP-MVIP($\mathbf{c} - \Lambda, \Omega_{\mathbf{f}}^{\mathbf{g}}$) Problema de asignación multimodal con modos combinados en el espacio de flujo en los arcos para costes no simétricos

TAP-SE Problema de asignación de tráfico bajo equilibrio del sistema (segundo principio de Wardrop)

TAP-VIP($\mathbf{c}, \Omega_{\mathbf{f}}^{\mathbf{g}}$) Formulación varicional en en el espacido de los flujos en los arcos del problema de asignación de tráfico con demanda inelástica

TAP-VIP($\mathbf{C}, \Omega_{\mathbf{h}}^{\mathbf{g}}$) Formulación varicional en en el espacido de los flujos en los caminos del problema de asignación de tráfico con demanda inelástica

TEAP Problema de asignación en redes de transporte púlico

VIP($\mathbf{F}, X$) Problema de desigualdades variacionales definida por la función de costes $\mathbf{F}$ y la región factible X

WNLLS Estimación mínimo cuadrática ponderada

Bibliografía

[1] H. Z. Aashtiani. *The multi-modal traffic assignment problem*. PhD thesis, Operations Research Center, Massachusetts Institute of Technology, Cambridge, M.A, 1979.

[2] H. Z. Aashtiani and T. L. Magananti. A linearization and decomposition algorithm for computing the urban traffic equilibria. In *Proceeding of the IEEE Large Scale Systems Symposium*, pages 8–19, 1982.

[3] M. Abdulaal and L. LeBlanc. Continuous equilibrium network design models. *Transportation Research*, 13B:19–32, 1979.

[4] M. Abdulaal and L. LeBlanc. Methods for combining modal split and equilibrium assignment models. *Transportation Science*, 13:292–314, 1979.

[5] T. Abrahamsson and L. Lundqvist. Formulation and estimation of combined network equilibrium models with applications to Stockholm. *Transportation Science*, 33:80–100, 1999.

[6] R. E. Allsop. Some possibilities for using traffic control to influence trip distribution and route choice. In D. J. Buckley, editor, *Transportation and traffic theory (Proceedings of the Sixth International Symposium on Transportation and Traffic Theory)*, pages 345–375. Elsevier, New York, 1974.

[7] G. Anandalingam, P. Mathieu, C.L. Pittard, N. Sinha, and A. Vernekar. Nontraditional search technique for solving bi-level linear programming problems. Technical report, University of Pennsylvania, Philadelphia, 1988.

[8] J.-P. Aubin and H. Frankowska. *Set-Valued Analysis*, volume 2 of *Systems & Control: Foundations & Applications*. Birkhäuser, Boston, MA, 1990.

[9] T. Baar and G. J. Olsder. *Dynamic Noncooperative Game Theory*. Academic Press, New York, 1982.

[10] J. F. Bard. Convex two-level optimization. *Mathematical Programming*, 40:15–27, 1988.

[11] J. F. Bard. *Practical Bilevel Optimization. Algorithms and Applications", Series Nonconvex optimization and its applications.* Kluwer Academic Publishers, Dordrecht, 1998.

[12] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, New York, NY, second edition, 1993.

[13] M. J. Beckman, C. B. McGuire, and C. B. Wisten. *Studies in the Economics of Transportation*. Yale University Press, New Haven, 1956.

[14] M. Ben-Akiva and S. Lerman. *Discrete choice analysis: Theory and applications to travel demand*. MIT Press, Cambridge, MA, 1995.

[15] M. E. Ben-Akiva and John L. Bowman. Activity based travel demand model systems. In Patrice Marcotte and Sang Nguyen, editors, *Advanced Transportation Modelling*, pages 27–46. Kluwer Academic Publishers, Massachusetts, 1998.

[16] F. Benitez. Una metodología eficiente para la estimación y ajuste de matrices origen-destino. In A. López and F. Robusté, editors, *Actas del III Congreso de Ingeniería del Transporte*, pages 209–218, Barcelona, 1998. CIMNE.

[17] D. P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, San Diego, CA, 1982.

[18] D. P. Bertsekas. Projected Newton methods optimization problems with simple constraints. *SIAM Journal on Control and Optimization*, 20:221–246, 1982.

[19] D. P. Bertsekas. Minicost grid problem generator. http://web.mit.edu/dimitrib/www/home.html, 1990.

[20] D. P. Bertsekas. *Network Optimization: Continuous and Discrete models*. Athena Scientific, Belmont, Massachusetts, 1998.

[21] D. P. Bertsekas and E. Gafni. Projections methods for variational inequalities with applications to the traffic assignment problem. *Mathematical Programming Study*, 17:139–159, 1982.

[22] D. P. Bertsekas, L. C. Polymenakos, and P. Tseng. $\epsilon-$ relaxation and auction methods for separable convex cost network flow problems. In P. M. Pardalos, W. W. Hager, and D. W. Hearn, editors, *Network Optimization*, number 450 in Lecture Notes in Economics and Mathematical Systems, pages 103–126. Springer-Verlag, Berlin, 1997.

[23] Z. Bi, P. Calami, and A. Conn. An exact penalty function approach for the linear bilevel programming problem. Techical Report 167-0-310789, Department of Systems Design and Engineering, University of Waterloo, Waterloo, 1989.

[24] W. F. Bialas and M. H Karwn. On two-level linear optimization. *IEEE Transaction on Automatic Control*, AC-27:211–214, 1982.

[25] W. F. Bialas and M. H Karwn. Two-level linear programming. *Management Science*, 30:1004–1020, 1984.

[26] M. Bierlaire, T. Lotan, and P. Toint. On the overspecification of multinomial and nested logit models due to alternative specific constants. *Transportation Science*, 31:363–371, 1997.

[27] G. N. Bifulco. A stochastic user equilibrium assignment model for la evaluation of parking policies. *European Journal of Operational Research*, 71:269–287, 1993.

[28] B. Bouzaïene-Ayari, M. Gendreau, and A. Nguyen. Passenger assignment in congested transit networks: A historical perspective. In Patrice Marcotte and Sang Nguyen, editors, *Advanced Transportation Modelling*, pages 47–71. Kluwer Academic Publishers, Massachusetts, 1998.

[29] D. E. Boyce. Urban transportation network-equilibrium and design models: recent achievements and future prospects. *Environment and Planning A.*, 16:1445–1474, 1984.

[30] D. E. Boyce. Integration of supply and demand models in transportation and location: Problem formulations and research questions. *Environ. Plan.*, A18:485–489, 1986.

[31] D. E. Boyce and B.N. Janson. A discrete transportation network design problem with combined trip distribution and assignment. *Transportation Research*, 14B:147–154, 1980.

[32] H. Brézis. Équations et inéquations non linéaires dans les espaces vectoriales en dualité. *Annalees de lÍnstittut Fourier*, 18:115–175, 1968.

[33] J. V. Burke and M. C. Ferris. Characterization of solution sets of convex programs. *Operations Research Letters*, 10:57–60, 1991.

[34] J. V. Burke and M. C. Ferris. Weak sharp minima in mathematical programming. *SIAM Journal on Control and Optimization*, 31:1340–1359, 1993.

[35] J. V. Burke and J. J. Moré. On the identification of active constraints. *SIAM Journal on Numerical Analysis*, 25:1197–1211, 1988.

[36] J. V. Burke and J. J. Moré. Exposing constraints. *SIAM Journal on Optimization*, 4:573–595, 1994.

[37] G. E. Cantarella. A general fixed-point approach to multimode multi-user equilibrium assignment with elastic demand. *Transportation Science*, pages 107–128, 1997.

[38] G. E. Cantarella and A. Sforza. Methods for equilibrium network traffic signal setting. In A. R. Odoni, L. Bianco, and G. Szegö, editors, *Flow Control of Congested Networks*, volume 38 of *NATO ASI Series F: Computer and Systems Science*, pages 69–89. Springer-Verlag, Berlin, 1986.

[39] S. Carrese, S. Gori, and T. Picano. Relationship between parking location and traffic flows in urban areas. In L. Bianco and P. Toth, editors, *Advanced Methods in Transportation Analysis*, pages 183–214. Springer-Verlag, Berlin, 1996.

[40] E. Cascetta and S. Nguyen. A unified framework for estimating or updating origin/destination matrices from traffic counts. *Transportaion Research B*, 22:437–455, 1986.

[41] E. Castillo, A. Conejo, P. Pedregal, R. García, and N. Alguacil. *Building and solving mathematical programming models in engineering and science*. Wiley-Interscience, New York, Aceptado.

[42] CATS. *Chicago Area Transportation Study*. Final Report, Volume 2, 300 W Adams Street, Chicago, IL, 1960.

[43] J. De Cea and E. Fernández. Transit assignment to minimal routes: An efficient new algorithm. *Traffic Engineering and Control*, 30:491–494, 1989.

[44] J. De Cea and E. Fernández. Transit assignment for congested public transport systems: An equilibrium model. *Transportation Science*, 27:133–147, 1993.

[45] A. Charnes and W. W. Cooper. Nonlinear network flows and convex programming over incidence matrices. *Naval Research Logistics Quartely*, 5:231–240, 1958.

[46] M. Chen and A. S. Alfa. A network design algorithm using a stochastic incremental traffic assignment approach. *Transportation Science*, 25:215–224, 1991.

[47] Y. Chen. Bilevel programming problems: analysis, algorithms and applications. Publication 984, Centre de Recherche sur les Transports, Université de Montréal, 1994.

[48] Y. Chen and M. Florian. O-D demand adjustment problem with congestion: Part I. model analysis and optimality conditions. In L. Bianco and P. Toth, editors, *Advanced Methods in Transportation Analysis*, pages 1–22. Springer-Verlag, Berlin, 1996.

[49] C. Chiriqui. Réseaux de transport en commun: les problémes de cheminement et d'accès. Publication 11, Centre de Recherche sur les Transports, Université de Montréal. Montréal, 1974.

[50] C. Chiriqui and P. Robillard. Common bus lines. *Transportation Science*, 9:115–121, 1975.

[51] S. C. Choi, W. S. Desarbo, and P. T. Harker. Product positioning under price competition. *Management Science*, 36:175–199, 1990.

[52] F. H. Clarke. *Optimization and Nonsmooth Analysis*. SIAM, Philadelphia, 1990.

[53] E. Codina and J. Barceló. Adjustment of O-D trip matrices from observed volumes: an algorithmic approach based on conjugate directions. In *Proceedings of 8^{th} EURO Working Group on Transportation*, Roma, Italy, 2000.

[54] M. Coffin and M. F. Saltzman. Statistical analysis of computational tests of algorithms and heuristics. *INFORMS Journal on Computing*, 12:24–44, 2000.

[55] I. Constantin and M. Florian. A method for optimizing the frequencies in transit network: a especial case of nonlinear bilevel programming. Technica report TRISTRAN I, Centre de Recherche sur les Transports, Université de Montréal, 1991.

[56] P. Coppola and G. N. Bifulco. A joint model of mode/parking choice with elastic parking demand. In *Proceedings of the Conference 6^{th} EURO Working Group on Transportation*, Gothenburg, Sweden, 1998.

[57] S. C. Dafermos. *Traffic assignment and resource allocation in transportation networks.* PhD thesis, The Johns Hopkins University, Baltimore, M. A., 1968.

[58] S. C. Dafermos. The traffic assignment problem for multiclass-user transportation networks. *Transportation Science*, 6:73–87, 1972.

[59] S. C. Dafermos. The traffic equilibrium and variational inequalities. *Transportation Science*, 14:42–54, 1980.

[60] S. C. Dafermos. The general multi-modal network equilibrium problem with elastic demand. *Networks*, 11:57–72, 1982.

[61] S. C. Dafermos and F. T. Sparrow. The traffic assignment problem for a general network. *Journal of Research of the National Bureau of Standards*, 73B:91–118, 1969.

[62] C. F. Daganzo and M. Kusnic. Two properties of the nested logit models. *Transportation Science*, 27:395–400, 1993.

[63] A. J. Daly and S. Zachary. Improved multiple choice models. In D.A. Hensher and M. Q. Dalvi, editors, *Determinants of Travel Choice*, pages 335–357. Teakfield, Farnborough, U.K, 1978.

[64] O. Damberg, J.T. Lundgren, and M.Patriksson. An algorithm for the stochastic user equilibrium problem. *Transportation Research B*, 30:115–131, 1996.

[65] R. S. Dembo and U. Tulowitzki. Computing equilibria on large multicommodity networks: An application of truncated quadratic programming algorithms. *Networks*, 18:273–284, 1988.

[66] S. Dempe. An implicit function approach to bilevel programming problems. In A. Migdalas, P. M. Pardalos, and P. V ärbrand, editors, *Multilevel Optimization: Algorithms and Applications*, pages 273–294. Kluwer Academic Publishers, Netherlands, 1998.

[67] E. W. Dijkstra. A note on two problems in connexion with grphs. *Numerische Mathematik*, 1:269–271, 1959.

[68] J. C. Dunn. On the convergence of projected gradient processes to singular critical points. *Journal of Optimization Theory and Applications*, 55:203–216, 1987.

[69] J.P. Dussault and P. Marcotte. Conditions de régularité géometrique pour les inéquations variationnelles. *Recherche opérationelle*, 23:1–16, 1989.

[70] R.W. Eash, B. N. Janson, and D. E. Boyce. Equilibrium trip assignment: advantages and implications for practice. *Transportation Research Record*, 728:1–8, 1979.

[71] S. Erlander. Accesibility, entropy and the distribution and assignment of traffic. *Transportation Research*, 11:149–153, 1977.

[72] S. P. Evans. Derivation and analysis of some models for combining trip distribution and assignment. *Transportation Research*, 10:37–57, 1976.

[73] E. Fernández, J. De Cea, M. Florian, and E. Cabrera. Network equilibrium models with combined modes. *Transportation Science*, 28:182–192, 1994.

[74] E. Fernández and T. L. Friesz. Equilibrium predictions in transportations markets: the state of the art. *Transportation Research*, 17:155–172, 1983.

[75] P. Ferrari. A model of urban transport management. *Transportation Research B*, 33:43–61, 1999.

[76] C. Fisk and S. Nguyen. Solution algorithms for network equilibrium models with asymetric user costs. *Transportation Science*, 16:361–381, 1982.

[77] C. S. Fisk and D. E. Boyce. Alternative variational inequality formulations of the network equilibrium-travel choice problems. *Transportation Science*, 17:454–463, 1983.

[78] M. Florian. A traffic equilibrium model of travel by car and public transit modes. *Transportation Science*, 11:166–179, 1977.

[79] M. Florian. Asymmetrical variable demand multi-mode traffic equilibrium problems: existence and uniqueness of solution and a solution algorithm. Publication 347, Département d'Informatique et de Recherche Opérationelle, Université de Montreéal. Montréal, 1979.

[80] M. Florian. Private communication, 1990.

[81] M. Florian and Y. Chen. A coordinate descent method for the bilevel O-D matrix adjustment problem. Publication, Centre de Recherche sur les Transports, Université de Montréal, 1993.

[82] M. Florian and D. Hearn. Network equilibrium models and algorithms. In M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, editors, *Network Routing*, volume 8, pages 485–550. Informs, North-Holland, 1995.

[83] M. Florian and H. Los. Determining intermediate origin-destination matrices for the analysis of composite mode trips. *Transportation Research B*, 13:91–103, 1978.

[84] M. Florian and M. Los. Determining intermediate origin-destination matrices for the analysis of composite mode trips. *Transportation Research*, 13B:91–103, 1979.

[85] M. Florian and S. Nguyen. A combined trip distribution modal split and trip assignment model. *Transportation Research*, 12:241–246, 1978.

[86] M. Florian, S. Nguyen, and Ferland. On the combined distribution-assignment of traffic. *Transportation Science*, 9:43–53, 1977.

[87] M. Florian and H. Spiess. On binary mode choice/assignment models. *Transportation Science*, 17:32–47, 1983.

[88] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:95–110, 1956.

[89] T. L. Friesz, H-J. Cho, N. Tobin, and G. Anandalingam. A simulated annealing approach to the network design problem with variational inequality constraints. *Transportation Science*, 26:18–26, 1992.

[90] T. L. Friesz and P.T. Harker. Properties of the iterative optimization-equilibrium algorithm. *Civil Engineering Systems*, 2:142–154, 1985.

[91] T. L. Friesz, R.L. Tobin, H.-J. Cho, and N.J. Mehta. Sensibity analysis based heuristic algorithms for mathematical programs with variational inequality constraints. *Mathematical Programming*, 48:265–284, 1990.

[92] G. Gallo and S. Pallotino. Shortest paths algorithms. *Annals of Operations Research*, 13:3–79, 1988.

[93] R. García, M.L. López, and D. Verastegui. Extensions of Dinckelbahc's algorithm for solving nonlinear fractional programming problems. *TOP, Journey of the Spanish Statistical and Operation Research Society*, 7:33–70, 1999.

[94] R. García and A. Marín. Algoritmos para modelos de asignación de tráfico con modos combinados. In *Actas del XXIII Congreso Nacional de Estadística e Investigación Operativa*, Valencia. España, 1997. Departamento de Estadística e Investigación Operativa. Universidad de Valencia.

[95] R. García and A. Marín. Urban multi-modal interchanges (macro vision). In *Proceedings of EURO XV–INFORMS XXXIV Joint International Meeting*, Barcelona, 1997.

[96] R. García and A. Marín. Using restricted simplicial decomposition within partial linearization methods. In *Proceedings of* 16^{th} *International Symposium on Mathematical Programming*, Lausanne, Switzerland, Agosto 24–29, 1997. École Polytechnique Fédérale de Lausanne.

[97] R. García and A. Marín. A bi-level programming approach for estimation of origin-destination matrix and calibration of parameters of a network equilibrium model with combined modes. In *Proceedings of the Conference* 6^{th} *EURO Working Group on Transportation*, Gothenburg, Sweden, 1998.

[98] R. García and A. Marín. Estimación de matrices origen-destino a partir de observaciones de flujo en los arcos y de la demanda. In A.López and F.Robusté, editors, *Actas del III Congreso de Ingeniería del Transporte*, pages 219–226, Barcelona, 1998. CIMNE.

[99] R. García and A. Marín. Modelo de diseño de intercambiadores en una red con modos combinados. In A.López and F.Robusté, editors, *Actas del III Congreso de Ingeniería del Transporte*, pages 227–235, Barcelona, 1998. CIMNE.

[100] R. García and A. Marín. Urban multimodal interchange design methodology. In *Proceedings of the* 11^{th} *Min-EURO Conference on Artificial Intelligence in Transportation Systems and Science, and* 7^{th} *EURO Working Group on Transportation*, volume 98, pages XV1–XV5, Espoo, Finland, 1999. Helsinki University of Techology, Transportation Engineering Publication.

[101] R. García and A. Marín. Parking capacity and pricing in park'n ride trips: a continuous equilibrium network design problem. In *Actas del X Congreso Latino Americano de Investiagción de Operaciones*, México D. F., México, Septiembre 4–8, 2000.

[102] R. García and A. Marín. Urban multimodal interchange design methodology. In M. Pursula and J. Niittymäki, editors, *Mathematical Methods on Optimization in Transportation Systems*, volume 48 of *Applied Optimization*, pages 49–79. Kluwer Academic Publishers, Dordrecht, 2001.

[103] R. García, A. Marín, and M. Patriksson. Restricted simplicial decomposition methods with partial linearization subproblems. Report, Universidad Politécnica de Madrid, 1997.

[104] R. García, A. Marín, and M. Patriksson. A class of column generation/simplicial decomposition algorithms in convex differentiable optimization, I: Convergence analysis. Technical report, Departamento de Matemática Aplicada y Estadística, Universidad Politécnica de Madrid, 2000.

[105] R. García, A. Marín, and M. Patriksson. A class of column generation/simplicial decomposition algorihms. In *Proceedings of ISMP 2000 17th International Symposium on Mathematical Programming*, Atlanta, GA, USA, del 7–11 de Agosto, 2000.

[106] R. García, A. Marín, and M. Patriksson. A class of column generation/simplicial decomposition algorithms in convex differentiable optimization, II: Numerical analysis. Technical report, Departamento de Matemática Aplicada y Estadística, Universidad Politécnica de Madrid, 2000.

[107] R. García, A. Marín, and M. Patriksson. A class of column generation/simplicial decomposition for equilibrium traffic assignment models. In *Proceedins of* 8^{th} *Meeting of the EURO Working Group on Transportation*, Roma, Italia, Septiembre 11-15, 2000.

[108] R. García, A. Marín, and M. Patriksson. Network design applications of the class of column generation / simplicial decomposition algorithms in convex differentiable optimization. *Investigación Operacional*, 22:1–10, 2001.

[109] R. García, A. Marín, and M. Patriksson. Network equilibrium models: A class of column generation/simplicial decomposition algorithms. In *Proceedings of TRITRAN IV*, Azores, Junio, 2001.

[110] N.H. Gartner. Area traffic control and network equilibrium. In M.A. Florian, editor, -, number 118 in Lecture Notes in Economics and Mathematical Systems, pages 274–297. Springer-Verlag, Berlin, 1976.

[111] A. Geoffrion. Elements of large-scale mathematical programming. *Management Science*, 16, 1970.

[112] A. Goldstein. Convex programming in Hilbert space. *Bulletin of the American Mathematical Society*, 70:709–710, 1964.

[113] M. Guignard. Generalized Kuhn–Tucker conditions for mathematical programming problems in a Banach space. *SIAM Journal on Control*, 7:232–241, 1969.

[114] M. D. Hall, D. Van Vliet, and L. G. Willumsen. SATURN: A simulation assignment model for the evaluation of traffic management schemes. *Traffic Engineering and Control*, 21:168–176, 1980.

[115] J. Hammond. *Solving Asymmetric Variational Inequality Problems and Systems of Equations with Generalized Nonlinear Programming Algorithms*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachussetts, USA, 1984.

[116] A.F. Han and H.M. Wilson. The allocation of buses in heavily utilised networks with overlapping routes. *Transportation Research*, 16B:221–232, 1982.

[117] I. Hansen, A. Marín, and A. Rystam. EURITRANS: macro methodology. Euritrans Group Meeting, Rotterdam, 1996.

[118] P. Hansen, B. Jaumard, and G. Savard. New branch-and-bound rules for linear bilevel programming. *SIAM Journal on Scientific and Statistical Computing*, 13:1194–1217, 1992.

[119] P. T. Harker. Accelerating the convergence of the diagonalization and projection algorithms for finite-dimensional variational inequalities. *Mathematical Programming*, 41:29–59, 1988.

[120] P. Hartman and G. Stampacchia. On some non-linear elliptic differential-functional equations. *Acta Mathematica*, 115:271–310, 1966.

[121] D. Hearn, S. Lawphongpanich, J. Ventura, and K. Yang. RSDNET user manual. Research Report 87-17, Department of Industrial and Systems Engineering, University of Florida, Florida, 1987.

[122] D. H. Hearn and M. Ramana. Solving congestion toll pricing models. In Patrice Marcotte and Sang Nguyen, editors, *Advanced Transportation Modelling*, pages 109–124. Kluwer Academic Publishers, Massachusetts, 1998.

[123] D. W. Hearn, S. Lawphongpanich, and J. A. Ventura. Finiteness in restricted simplicial decomposition. *Operations Research Letters*, 4:125–130, 1985.

[124] D. W. Hearn, S. Lawphongpanich, and J. A. Ventura. Optimization algorithms for congested network model. In *Proceedings of the NATO Advanced Research Workshop on flow Control of Congested Networks*, Capri, Italy, 1986, 1987. Springer-Verlag.

[125] D. W. Hearn, S. Lawphongpanich, and J. A. Ventura. Restricted simplicial decomposition: Computation and extensions. *Mathematical Programming Study*, 31:99–118, 1987.

[126] B. F. Hobbs and K. A. Kelly. Using game theory to anlyse electric transmission pricing policies in the United States. *European Journal of Operational Research*, 56:154–171, 1992.

[127] B. von Hohenbalken. Simplicial decomposition in nonlinear programming algorithms. *Mathematical Programming*, 13:49–68, 1977.

[128] C. A. Holloway. An extension of the Frank and Wolfe method of feasible directions. *Mathematical Programming*, 6:14–27, 1974.

[129] A. J. Horowtiz. Test of ad hoc algorithm of elastic-demand equilibrium traffic assigment. *Transportation Research*, 23B:309–313, 1984.

[130] H. Huang and W.K. Lam. Modified Evans' algorihms for solving the combined trip distribution and assignment problem. *Transportation Research*, pages 325–337, 1991.

[131] J. D. Hunt and S. Teply. A nested logit model of parking location choice. *Transportation Research B*, 127:253–265, 1993.

[132] B. Jaumard, G. Savard, and X. Xiong. An exact algorithm for convex bilevel programming. Working paper G-95-33, École des Hautes Études Commerciales, École Polytechnique de Montréal, Québec, 1995.

[133] R. G. Jeroslow. The polynomial hierarchy and a simple model for competitive analysis. *Mathematical Programming*, 32:273–284, 1988.

[134] J.J. Júdice and A. M. Faustino. A sequential LCP method for bilevel linear programming. *Annals of Operations Research*, 34:89–106, 1992.

[135] A. Kennington and R. Helgason. *Algorithms for Network Programming*. John Wiley & Sons, New York,NY, 1980.

[136] D. Klingman, A. Napier, and J. Stutz. NETGEN- A program for generating large scale (un) capacitated assignment, transportation, and minimum cost flow network problems. *Management Science*, 20:814–822, 1974.

[137] W. H. K. Lam and H-J. Huang. A combined trip distribution and assignment model for multiple user classes. *Transportation Research*, 26:275–282, 1992.

[138] T. Larsson, A. Migdalas, and M. Patriksson. A generic column generation scheme. Technical Report LiTH-MAT-R-94-18, Department of Mathematics, Linköping Institute of Technology, Linköping Institute of Technology, Linköping, Sweden, 1994.

[139] T. Larsson, M.Patriksson, and A.-B. Strömberg. Ergodic convergence in subgradient optimization. Technical report, Department of Mathematics, Linköping Institute of Technology, Linköping Institute of Technology, Linköping, Sweden, 1997.

[140] T. Larsson and M. Patriksson. Simplicial decomposition with disaggregated representation for the traffic assignment problem. *Transportation Science*, 26:4–17, 1992.

[141] T. Larsson, M. Patriksson, and C. Rydergren. Applications of simplicial decomposition with nonlinear column generation to nonlinear network flows. In P. M. Pardalos, W. W. Hager, and D. W. Hearn, editors, *Network Optimization*, number 450 in Lecture Notes in Economics and Mathematical Systems, pages 346–373. Springer-Verlag, Berlin, 1997.

[142] T. Larsson, M. Patriksson, and A. B. Strömberg. Ergodic, primal convergence in dual subgradient schemes for convex programming. *Mathematical Programming*, 86:283–312, 1999.

[143] L. S. Lasdon. *Optimization Theory for Large Systems*. Macmillan, New York, NY, 1970.

[144] S. Lawphongpanich and D. W. Hearn. Simplicial decomposition of the asymmetric traffic problem. *Transportation Research*, 18B:123–133, 1984.

[145] L. J. LeBlanc and K. Farhangiam. Efficient algorithms for solving elastic demand traffic assignment problems and mode split-assignment problems. *Transportation Science*, 15:306–317, 1981.

[146] L. J. LeBlanc, R. V. Helgason, and D. E. Boyce. Improved efficiency of the Frank-Wolfe algorithm for convex network programs. *Transportation Science*, 19:445–462, 1985.

[147] L.J. Leblanc. An algorithm for the discrete network design problem. *Transportation Science*, 9:183–199, 1975.

[148] L.J. Leblanc. Transit system network design. *Transportation Research*, 22B:383–390, 1988.

[149] L.J. LeBlanc, E. K. Morlok, and W. P. Pierskalla. An efficient approach to solving the road network equilibrium traffic assignment problem. *Transportation Research B*, 9:309–318, 1975.

[150] E.S. Levitin and B. T. Polyak. Constrained minimization methods. *USSR Computational Mathematics and Mathematical Physics*, 6:1–50, 1966.

[151] J. T. Ludgren and M. Patriksson. An algorithm for the combined distribution and assignment model. Technical report, Department of Mathematics, Linköping Institute of Technology, Linköping Institute of Technology, Linköping, Sweden, 1997.

[152] D. G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, Reading, MA, second edition, 1984.

[153] J. T. Lundgren. Models for the OD-matrix estimation problem. Working Paper LiTH-MAT/OPT-WP-1991-14, Dept. of Mathematics, Institute of Technology, Linköping, Sweden, 1991.

[154] Z. Luo, J. Pang, and D. Ralph. *Mathematical programs with equilibrium constraints*. Cambridge University Press, Cambridge, 1996.

[155] T. Magnanti. Models and algorithms for predicting unban traffic equilibria. In M. Florian, editor, *Transportation Planning Models*, pages 153–518. North-Holland, Amsterdam, 1984.

[156] H. S. Mahmassani and K. C. Mouskos. Vectorization of transportation network equilibrium codes. Publication, Department of Civil Engineering, University of Texas, Austin, TX, 1989.

[157] P. Marcotte. An analysis of heuristics for the continuous network design problem. In *Proceedings of the 8th International Symposium on Transportation and Traffic Theory*, pages 452–468. University of Toronto, 1981.

[158] P. Marcotte. Network optimization with continuous control parameters. *Transportation Science*, 17:181–197, 1983.

[159] P. Marcotte. Network design problem with congestion effects: a case of bilevel programming. *Mathematical Programming*, 34:142–162, 1986.

[160] P. Marcotte and J.-P. Dussault. A sequential linear programming algorithm for solving monotone variational inequalities. *SIAM Journal on Control and Optimization*, 27:1260–1278, 1989.

[161] P. Marcotte and J. Guélat. Adaptation of a modified newton method for solving the asymmetric traffic equilibrium problem. *Transportation Science*, 22:112–124, 1988.

[162] P. Marcotte and G. Marquis. Efficient implementation of heuristic for the continuous network design problems. *Annals of Operation Research*, 34:163–176, 1992.

[163] A. Marín. *Contribuciones a la teoría matemática de la planificación del transporte y sus aplicaciones*. PhD thesis, Escuela Técnica Superior de Ingenieros Aeronaúticos, Universidad Politécnica de Madrid, 1982.

[164] A. Marín. Inaccurate step search: Frank-Wolve algorithm and elastic demand model. In *Actas del V Congreso Panamericano de Ingeniería de Tránsito y Transporte*, Mayaguez, Puerto Rico, Julio 18-22, 1988.

[165] A. Marín. A multimodal combined modelo analyzed via generalized geometric programming. *Transportation Research*, 22B:391–397, 1988.

[166] A. Marín. Generalized benders decomposition applied to transportation models. In *Proceedings of Workshop on Large-Scale Optimization*, Coimbra, 1991.

[167] A. Marín. Restricted simplicial decomposition with side constraints. *Networks*, 26:199–215, 1995.

[168] A. Marín and R. García. Terminales intermodales urbanos. In *Actas del II Congreso Nacional del Transporte*, Madrid, 1996. Escuela Técnica Superior de Ingenieros de Caminos Canales y Puertos. Universidad Politécnica de Madrid. España.

[169] D. McFadden. Conditional logit analysis of qualitative choice behavior. In P. Zarembka, editor, *Frontiers in Econometrics*, pages 105–142. Academic Press, New York, 1974.

[170] Q. Meng, H. Yang, and M.G.H. Bell. An equivalent continously differentiable model and a locally convergent algorithm for the continuous network design problem. *Transportation Research*, 1999.

[171] J.A. Mesa and F.A. Ortega. Park-and-ride station catchment areas in metropolitan transit systems. In *Proceedings of the 11th Min-EURO Conference on Artificial Intelligence in Transportation Systems and Science, and 7th EURO Working Group on Transportation*, volume 98, pages XXIII1–XXIII5, Espoo, Finland, 1999. Helsinki University of Techology, Transportation Engineering Publication.

[172] A. Migdalas. A regularization of the Frank–Wolfe method and unification of certain nonlinear programming methods. *Mathematical Programming*, 65:331–345, 1994.

[173] L. Montero. *A simplicial decomposition approach for solving the variational inequality formulation of the general traffic assignment problem for large scale networks.* PhD thesis, Universidad Politécnica de Cataluña, Barcelona, Spain, 1992.

[174] L. Montero and J. Barceló. A simplicial decomposition algorithm for solving the variational inequality formulation of the general traffic assignment problem. *TOP, Journey of the Spanish Statistical and Operation Research Society*, 4:225–256, 1996.

[175] J. J. Moré. Coercivity conditions in nonlinear complementarity problems. *SIAM Review*, 16:1–16, 1974.

[176] J. M. Mulvey, S.A. Zenios, and D. P. Ahlfeld. Simplicial decomposition for convex generalized networks. *Journal of Information & Optimization Sciences*, 11:359–387, 1990.

[177] J. F. Nash. Non-cooperative games. *Annals of Mathematics*, 54:286–295, 1951.

[178] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization.* John Wiley & Sons, New York, 1988.

[179] S. Nguyen and C. Dupuis. An efficient method for computing traffic equilibria in networks with asymmetric transportation costs. *Transportation Science*, 18:185–202, 1984.

[180] S. Nguyen and L. James. TRAFFIC- an equilibirum traffic assignment program. Publication 17, Centre de Recherche sur les Transports, Université de Montreéal. Montréal, 1975.

[181] S. Nguyen and S. Pallotino. Equilibrium traffic assignment for large scale transit network. *European Journal of Operational Research*, 37:176–186, 1988.

[182] S. Nickel, A. Chöbel, and T. Sonneborn. Hub location problems in urban traffic network. In *Proceedings of the 11th Min-EURO Conference on Artificial Intelligence in Transportation Systems and Science, and 7th EURO Working Group on Transportation*, volume 98, pages XXV1–XXV5, Espoo, Finland, 1999. Helsinki University of Techology, Transportation Engineering Publication.

[183] J. Nocedal and J. Wright. *Numerical Optimization.* Springer-Verlag, NY, 1999.

[184] A. R. Odoni and R. C. Larson. *Urban Operation Research.* Prentice-Hall, Englewood Cliffs, NJ,, 1981.

[185] Roads Bureau of Public. *Traffic Assignment Manual*. U.S. Department of Commerce, Washington, D.C., 1964.

[186] K. Okuguchi. *Expectations and Stability in Oligopoly Models*. Number 138 in Lectures Notes in Economics and Mathematical Systems. Springer-Verlarg, Berlin, 1976.

[187] N. Oppenheim. *Urban Travel Demand Modelling*. Wiley-Interscience, N. Y., 1994.

[188] J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York, NY,, 1970.

[189] J. de D. Ortúzar and L. G. Willumsen. *Modelling Transport*. John Wiley & Sons, Chichester, 1994.

[190] M. Ottomanelli. Effects of data in aggregate travel demand models calibration with traffic counts. In *Proceedings of the 11^{th} Min-EURO Conference on Artificial Intelligence in Transportation Systems and Science, and 7^{th} EURO Working Group on Transportation*, volume 98, pages XX–XXI, Espoo, Finland, 1999. Helsinki University of Techology, Transportation Engineering Publication.

[191] J. S. Pang and D. Chan. Iterative methods for variational and complementary problems. *Mathematical Programming*, 24:284–313, 1982.

[192] J. S. Pang and C. S. Yu. Linearized simplicial decomposition methods for computing traffic equilibria on networks. *Networks*, 14:427–432, 1984.

[193] M. Patriksson. Partial linearization methods in nonlinear programming. *Journal of Optimization Theory and Applications*, 78:227–246, 1993.

[194] M. Patriksson. A unified description of iterative algorithms for traffic equilibria. *European Journal of Operational Research*, 71:154–176, 1993.

[195] M. Patriksson. *A unified framework of descent algorithms forn nonlinear programs and variational inequalities*. PhD thesis, Department of Mathematics, Linköping Institute of Technology, Linköping, Sweden, 1993.

[196] M. Patriksson. *Traffic Assignment Problem. Models and Methods*. VSP, Utrecht, The Netherlands, 1994.

[197] M. Patriksson. Cost approximation: A unified framework of descent algorithms for nonlinear programs. *SIAM Journal on Optimization*, 8:561–582, 1998.

[198] M. Patriksson. *Nonlinear programming and variational inequality problems. A unified approach*. Kluwer Academic Publishers, Dordrecht, 1999.

[199] M. Patriksson and R. T. Rockafellar. A mathematical model and descent algorithm for bilevel traffic management, Submitted. Submitted, 2000.

[200] E. R. Petersen. A primal-dual traffic assignment algorithm. *Management Science*, 22:87–95, 1975.

[201] B.T. Polyak. *Introduction to Optimization*. Optimization Software, New York, 1987.

[202] P. Poorzahedy and M.A. Turnquist. Approximate algorithms for the discrete network design problem. *Transportation Research*, 16B:45–56, 1992.

[203] W. H. Press, S.A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in FORTRAN. The Art of Scientific Computing*. Cambridge University Press, Cambridge, 1992.

[204] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, NJ, 1970.

[205] R. T. Rockafellar. Augmented lagrangian and applications of the proximal point algorithm. *Mathematics of Operations Research*, 1:97–116, 1976.

[206] R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis*, volume 317 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, Berlin, 1998.

[207] W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, Auckland, third edition, 1976.

[208] G. Salinetti and R. B. Wets. On the convergence of sequences of convex sets in finite dimensions. *SIAM Review*, 21:18–33, 1979.

[209] P. Serra and A. Weintraub. Convergence of decomposition algorithms for traffic assignment problem. In P. Hansen, editor, *Studies on Graphs and Discrete Programming*, pages 313–318. North-Holland, Amsterdam, 1981.

[210] Y. Sheffi. *Urban Transportation Networks*. Prentice-Hall, Englewood, Clifs, N. J., 1985.

[211] H. D. Sherali, A. L. Soyster, and F. H. Murphy. Stackelberg-Nash-Cournot equilibria: characterizations and computations. *Operations Research*, 31:253–276, 1983.

[212] K. Shimizu, Y. Ishizuka, and J. Bard. *Nondifferentiable and Two-Level Mathematical Programming*. Kluwer Academic Publishers, Massachusetts, 1997.

[213] M. J. Smith. The existence, uniqueness and stability of traffic equilibria. *Transportation Research B*, 13:295–304, 1979.

[214] M. J. Smith. A descent algorithm for solving monotone variational inequalities and monotone complementary problems. *Journal of Optimization Theory and Applications*, 44:485–496, 1984.

[215] M. J. Smith and T. Van Vuren. Traffic equilibrium with reponsive traffic control. *Transportation Science*, 27:118–132, 1993.

[216] M. J. Smith, Y. Xiang, R. A. Yarrow, and M. Ghali. Bilevel and other modelling approaches to urban traffic management and control. In Patrice Marcotte and Sang Nguyen, editors, *Advanced Transportation Modelling*, pages 283–325. Kluwer Academic Publishers, Massachusetts, 1998.

[217] K. L. Sobel. Travel demand forecasting by using the nested multimodal logit model. *Transportation Research Record*, 7:301–309, 1980.

[218] H. Spiess. On optimal route choice strategies in transit networks. Technical Report Pub. 286, Centre de Recherche sur les Transports, Université de Montréal, 1983.

[219] H. Spiess. A gradient approach for the O-D matrix adjustment problem. Publication, Centre de Recherche sur les Transports, Université de Montréal, 1990.

[220] H. Spiess and M. Florian. Optimal strategies: A new assignment model for transit networks. *Transportation Research B*, 23:83–102, 1989.

[221] G. E. Stackelberg. *The Theory of Market Economy*. Oxford University Press, Oxford, 1952.

[222] G. Stampacchia. Variational inequalities. In *Theory and Applications of Monotone Operators. Proceedings of the NATO Advanced Study*, pages 101–192, Gubbio, Italy, 1969. Edizioni Oderisi.

[223] P. A. Steenbrink. *Optimization of Transportation Networks*. John Wiley & Sons, New York, 1974.

[224] C. Suwansirikul, T. Friesz, and R. L. Tobin. Equilibrium decomposed optimization: A heuristic for the continuous equilibrium network design problem. *Transportation Science*, 4:254–263, 1987.

[225] H. N. Tan, S. S. Gershwin, and M. Athans. Hybrid optimization in urban traffic networks. Techical Report DOT-TSC-RSP-79-7, National Technical Report Service, Springfield, 1979.

[226] R. L. Tobin and T. L. Friesz. Sensitivity analysis for equilibrium network flows. *Transportation Science*, 22:242–250, 1988.

[227] P. Toint and L. Wynter. Asymmetric multiclass traffic assignment: A coherent formulation. In *Proceedings of International Symposium on Transportation and Traffic Theory*, 1995.

[228] D. M Topkis and A. F. Veinott. On the convergence of some feasible direction algorithms for nonlinear programming. *SIAM J. Control*, 5:268–279, 1967.

[229] P. Tseng. Decomposition algorithm for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 70:109–135, 1991.

[230] US DOT. *UTPS Reference Manual US Department of Transportation*. US Department of Transportation, Washington, D.C, 1979.

[231] D. Vanderbilt and S. G. Louie. A Monte Carlo simulated annealing approach to optimization over continuous variables. *J. Comp. Phys.*, 56:259–271, 1984.

[232] J. A. Ventura and D. W. Hearn. Restricted simplicial decomposition for convex constrained problems. *Mathematical Programming*, 59:71–85, 1993.

[233] J. G. Wardrop. Some theoretical aspects of road traffic research. In *Proceedings of the Institute of Civil Engineers Part II*, pages 325–378, 1952.

[234] F.V. Webster. Traffic signal settings. Road Research Technical Paper 39, Department of Transport, HMSO, London, 1958.

[235] A. Weintraub, C. Ortiz, and J. Gonz lez. Accelerating convergence of the Frank-Wolfe algorithm. *Transportation Research*, 19:113–122, 1985.

[236] D. J. White and Anandalingan. A penalty function approach for solving bi-level programs. *Journal of Global Optimization*, 3:397–419, 1993.

[237] H. Williams, H. K. Lam, and H.J. Huang. A combined trip distribution and assignment model for multiple use classes. *Transportation Research*, 1992.

[238] P. Wolfe. Convergence theory in nonlinear programming. In J. Abadie, editor, *Integer and Nonlinear Programming*, pages 1–36. North-Holland, Amsterdam, 1970.

[239] J. H. Wu and M. Florian. A simplicial decomposition method for the transit equilibrium assignment problem. *Annals of Operations Research*, 44:245–260, 1993.

[240] J. H. Wu, M. Florian, and P. Marcotte. Transit equilibrium assignment: A model and solution algorithms. *Transportation Science*, 28:193–203, 1994.

[241] H. Yang. Heuristic algorithms for the bilevel Origin-Destination matrix estimation problem. *Transportation Research*, 29B:231–242, 1995.

[242] H. Yang. Sensitivity analysis for the elastic demand network equilibrium problem with applications. *Transportation Research*, 31B:55–70, 1997.

[243] H. Yang and M. G. H. Bell. Models algorithms for road network design: a review and some new developments. *Transportation Reviews*, 18:257–278, 1998.

[244] H. Yang and W. H. K. Lam. Optima road tolls under conditions of queueing and congestion. *Transportation Research*, 30A:319–332, 1996.

[245] H. Yang, T. Sasaki, and Y. Asakura. Estimation of origin-destination matrices from link traffic counts on congested networks. *Transportation Research*, 26B:417–434, 1992.

[246] H. Yang and S. Yagar. Traffic assignment and traffic control in general freeway-arterial corridor systems. *Transportation Research*, 28B:463–486, 1994.

[247] H. Yang, S. Yagar, Y. Iida, and Y. Asakura. An algorithm for the inflow control problem on urban freeway networks with user-optimal flows. *Transportation Research*, 28B:123–139, 1994.

[248] W.I. Zangwill. *Nonlinear Programming: A Unified Approach*. Prentice-Hall, Englewood Cliffs, NJ, 1969.

[249] Y.F. Zhang. *Parameter Estimation for Combined Models of Urban Travel Choices Consistent with Equilibrium Travel Costs*. PhD thesis, University of Illinois, Chicago, 1994.

[250] D. L. Zhu and P. Marcotte. Transit equilibrium assignment: A model and solution algorithms. *Coupling the auxiliary problem principle with descent methods of pseudoconvex programming*, 83:670–685, 1995.