



**TRABAJO FIN DE GRADO**

---

**Método de descomposición de Benders aplicado al  
problema de diseño de redes con carga fija**

---

Realizado por: María del Reposo Rivera Lineros

Supervisado por: Eduardo Conde Sánchez

FACULTAD DE MATEMÁTICAS  
DPTO DE ESTADÍSTICA E INVESTIGACIÓN OPERATIVA



# Índice general

<b>1. Introducción</b>	<b>5</b>
<b>2. Método de descomposición de Benders</b>	<b>9</b>
2.1. Descripción y esquema del algoritmo . . . . .	10
2.2. Variantes de la descomposición de Benders . . . . .	12
<b>3. Algunos problemas de diseño de redes con carga fija</b>	<b>15</b>
3.1. Problemas de diseño de redes con capacidad ilimitada . . . . .	16
3.1.1. Origen-destino UNDP . . . . .	16
3.1.2. Origen único UNDP . . . . .	18
3.1.3. Origen único, dos tecnologías UNDP . . . . .	20
3.2. Problemas de diseño de redes con capacidad limitada . . . . .	21
3.2.1. Una única tecnología CNDP . . . . .	21
3.2.2. Varias tecnologías CNDP . . . . .	22
3.2.3. CNDP con coste de capacidad escalonado . . . . .	24
<b>4. Una aplicación al rediseño de una red con carga fija y costes de transporte variables</b>	<b>27</b>
4.1. El problema de la red ferroviaria de alta velocidad entre España y Francia	27
4.2. Planteamiento del modelo. Algoritmo de Benders . . . . .	29
4.3. Modelo con distintas tecnologías . . . . .	35
<b>5. Experimento computacional</b>	<b>39</b>
5.1. Planteamiento del problema . . . . .	39
5.2. Resolución computacional con AMPL . . . . .	42
5.2.1. Fichero .dat . . . . .	42
5.2.2. Fichero .mod . . . . .	43
5.2.3. Fichero .run . . . . .	45
5.2.4. Salida . . . . .	49
<b>6. Conclusiones</b>	<b>61</b>

## Abstract

New communication networks are constantly emerging to improve connectivity services and facilitate the interconnection of devices of various types. This involves the development of different technologies, such as device-to-device communications, wireless sensor networks and vehicular communications. Mathematical Optimization techniques have always been at the heart of such design problems to formulate and propose computationally efficient numerical algorithms. Benders Decomposition, which is the focus of this document, is one of the more powerful techniques of Mathematical Programming.

To design efficient algorithms for large scale problems, the exploitation of the mathematical structure of the problem formulation is crucial. In this way, Benders Decomposition is one the most popular schemes, because it exploits the structure of the problem and decentralizes the overall computational burden. This method was proposed by Benders in 1962, with the main objective of tackling problems with complicating variables, which, when temporarily fixed, yield a problem significantly easier to handle.

The Benders Decomposition algorithm has been successfully applied to a wide range of difficult optimization problems. Successful applications are found in many different fields, including planning and scheduling, health care, transportation and telecommunications, energy and resource management and chemical process design.

The main purpose of this document is to describe this method and how it can be implemented in the context of network design. We present a general algorithm based on Benders Decomposition for solving the multicommodity uncapacitated fixed-charge network design problem, which is the type of problem that we will focus on. This problem is defined on a directed graph. The key feature of this model is its use in evaluating the trade-off between infrastructure investment and operational costs. The former is modeled by the fixed cost paid for using a given connection. The latter is modeled by a linear transportation cost paid per unit of commodity routed on such a connection. The goal is to route all commodities from origins to destinations at minimal cost.

In the final part of this document we conduct a numerical experiment in which a transportation network is designed according to a set of costs where some of them are considered unknown when decisions are made. An extension of the classical network design formulation is proposed in order to take into consideration this drawback.

# Capítulo 1

## Introducción

La noción de grafo (o de red) es de las primeras ideas que se adquieren en el estudio de estructuras abstractas. Este hecho no es una casualidad, puesto que nuestro mundo es un mundo entrelazado por redes: las que forman las personas en la sociedad, las de neuronas, la *World Wide Web*, las de transportes urbanos en una ciudad, las de rutas de los aeronáuticos...

El diseño de redes constituye un área muy activa y relevante de la Programación Matemática. Los problemas que comprende consisten, en general, en seleccionar de una red subyacente o potencial una subred que optimice un determinado indicador de eficiencia (función objetivo) y que permita que se cumplan ciertas condiciones (restricciones), como condiciones de flujo por ejemplo.

En la actualidad los problemas de diseño de redes de gran dimensión están muy presentes en múltiples contextos de la industria debido a su capacidad de modelar de manera realista multitud de procesos logísticos. Encontramos diversos ejemplos de aplicaciones del diseño de redes:

- Diseño de redes de transporte en las que la mercancía a transportar puede venir dada de diversas formas, por ejemplo, vehículos (redes de carreteras, ferroviarias o aéreas) o productos comerciales (redes que modelan la distribución de productos de almacenes a comercios).
- Diseños de redes de telecomunicaciones donde el ‘producto’ que fluye es la información, en las que se puede valorar el uso de distintas tecnologías, como es el caso de la fibra óptica o el cobre.
- Planificación laboral: el problema de organizar los turnos de trabajo del equipo de una empresa implica el diseño de un eficiente esquema en el que se alternen periodos laborables y de descanso. Este esquema debe satisfacer los requisitos de

personal en cada turno y debe cumplir con las condiciones correspondientes sobre la secuencia de los periodos de trabajo y descanso. En relación al diseño de redes, Balakrishnan y Wong [15] modelan este problema como un problema de flujo.

Problemas como el del árbol de unión de mínimo coste, el de Steiner, diseño de redes multiartículo, el del viajante, construcción de redes de supervivencia y de restauración, etc. son bien conocidos pero muchas de sus variantes se están investigando actualmente. De hecho, de algunos de ellos se sabe que se pueden resolver en tiempo polinomial y se conoce su estructura poliédrica, pero la mayoría tienen carácter no polinomial y, por tanto, necesitan para su resolución todas las herramientas posibles entre las que se encuentran descomposiciones, relajaciones o métodos heurísticos.

En este trabajo nos centraremos en el problema del diseño de redes con carga fija. La utilidad de este tipo de problema es su capacidad para evaluar la compensación entre la inversión en infraestructura y los costes operativos. El problema se define sobre un grafo dirigido. La inversión en infraestructura se modela a través de los llamados costes fijos. Cada arco tiene un coste fijo asociado, el cual debe ser pagado si el arco forma parte de la solución. Los costes operativos son usualmente costes de transporte y determinan el precio a pagar por el transporte de una unidad de mercancía a través de un arco. El objetivo es establecer una red donde se satisfagan ciertas condiciones de flujo minimizando el gasto en costes fijos y de transporte.

Para la resolución de estos problemas nos centraremos en el método de descomposición de Benders, introducido a principio de los años sesenta como un método de resolución de problemas de programación entera mixta. La idea esencial del método de descomposición de Benders es acelerar el proceso de búsqueda de soluciones identificando alguna estructura en el problema original que nos permita descomponerlo en dos problemas más sencillos de resolver. Una vez establecidos los dos problemas, se procede con el algoritmo de descomposición de Benders: se resuelven de manera iterada hasta que se verifique alguna condición que trataremos más adelante, existiendo una comunicación entre ellos generalmente basada en la información de los problemas duales.

Desde su introducción, la descomposición de Benders ha sido utilizada para resolver numerosos problemas entre los que se encuentran los de planificaciones de aerolíneas [16], transporte de mercancías peligrosas [17] e incluso problemas de servicios sanitarios [18].

Comenzaremos el trabajo describiendo de forma genérica el método de descomposición, exponiendo la teoría sobre la que se basa y proporcionando un esquema básico del algoritmo necesario para su implementación. En el capítulo 3 presentamos varios problemas de diseño de redes con carga fija a los que es aplicable el método de des-

composición de Benders y mencionamos los enfoques que algunos autores les han dado a estos problemas. En el capítulo 4 planteamos y desarrollamos una aplicación de la descomposición de Benders al rediseño de una red: el problema consiste en mejorar una red preexistente a través de la creación de nuevas conexiones o la mejora de la tecnología. Finalmente, presentamos un experimento computacional: resolvemos con AMPL un problema particular del modelo desarrollado en el capítulo anterior.





## Capítulo 2

# Método de descomposición de Benders

A principios de los años sesenta Jacques F. Benders propone en [19] un nuevo método de resolución de problemas de programación matemática entera mixta. La dificultad de este tipo de problema reside en que, dado que algunas variables que lo definen son de naturaleza entera, no se puede usar la propiedad de convexidad de la región factible, lo cual dificulta su resolución. El algoritmo de Benders se basa en la idea de identificar algún tipo de estructura que nos permita separar el problema en dos: el problema maestro y el subproblema. De esta forma obtenemos dos problemas que de forma separada son más sencillos de resolver, desde el punto de vista computacional, que el problema original. Cada vez que se resuelve el subproblema obtenemos un nuevo corte que se incorpora al problema maestro, el cual propone una aproximación a la solución óptima. El problema maestro y el subproblema son resueltos de forma iterativa, hasta que no se puedan generar más cortes o hasta que las cotas inferior y superior del óptimo que se generan durante el proceso discrepen menos que el error establecido. La solución obtenida en la última iteración del problema maestro constituye la solución del problema original.

En particular, el tipo de problema a tratar en este trabajo, el diseño de redes con carga fija, presenta un esquema natural de descomposición: las variables de diseño, es decir, las variables que representan la selección de los arcos y por tanto diseñan la red, forman parte del problema maestro, mientras que las variables que representan el flujo de mercancía son tratadas en el subproblema. De esta forma, en cada iteración el problema maestro genera un posible diseño de la red, para el cual el subproblema identifica el flujo óptimo de mercancía.

## 2.1. Descripción y esquema del algoritmo

Consideramos la formulación general del problema del diseño de redes de carga fija:

$$\text{Minimizar} \quad cx + dy \quad (2.1)$$

$$\text{sujeto a} \quad Ax + By \geq b, \quad (2.2)$$

$$Dy \geq e, \quad (2.3)$$

$$x \geq 0, \quad (2.4)$$

$$y \geq 0, y \in \mathbb{Z}. \quad (2.5)$$

donde los vectores  $x$  e  $y$  denotan las variables continuas y enteras, respectivamente, y los vectores  $c$  y  $d$  son los vectores de los costes asociados. Las matrices  $A$ ,  $B$  y  $D$  y los vectores  $b$  y  $e$  definen las restricciones del problema. Este problema puede ser expresado como:

$$\min_{y \in Y} \{dy + \min_{x \geq 0} \{cx : Ax \geq b - By\}\} \quad (2.6)$$

donde  $Y = \{y : Dy \geq e, y \geq 0, y \in \mathbb{Z}\}$ .

El problema  $\min_{x \geq 0} \{cx : Ax \geq b - By\}$  es lineal. Por tanto, asociando variables duales  $u$  a las restricciones  $Ax \geq b - By$ , podemos escribir el problema dual como:

$$\max_{u \geq 0} \{u(b - By) : uA \leq c\} \quad (2.7)$$

Este es el subproblema de la descomposición de Benders. Haciendo uso de la teoría de la dualidad, podemos reformular (2.6):

$$\min_{y \in Y} \{dy + \max_{u \geq 0} \{u(b - By) : uA \leq c\}\} \quad (2.8)$$

Observamos que la región factible del subproblema es independiente de la elección hecha para las variables  $y$ , usamos la notación

$$F = \{u : u \geq 0, uA \leq c\},$$

para esta región factible. Suponemos que  $F$  es no vacía ya que esto se correspondería, por el teorema fundamental de dualidad lineal (véase Teorema 6.1 de [11]), con un problema primal infactible o no acotado. Denotemos por  $u^p$  (con  $p = 1, \dots, P$ ) a los puntos extremos y por  $r^q$  (con  $q = 1, \dots, Q$ ) a las direcciones extremas de  $F$ .

La solución del subproblema puede ser acotada o no acotada. Por el teorema fundamental de la Programación lineal se tiene que, en el primer caso, la solución se alcanza en algún punto extremo  $u^p$  ( $p = 1, \dots, P$ ) y en el segundo caso, existe una dirección  $r^q$  tal que  $r^q(b - By) > 0$ . Este último caso se corresponde con un problema primal infactible,

por lo que debe ser evitado. Por tanto, debemos eliminar aquellos valores de  $y$  que impliquen una solución no acotada del problema dual. Para ello consideramos las restricciones:

$$r^q(b - By) \leq 0, \quad q = 1, \dots, Q.$$

Con estas restricciones, la solución del subproblema se alcanza en uno de los puntos extremos de  $F$ . Así el problema (2.8) se convierte en:

$$\begin{aligned} \text{Minimizar} \quad & dy + \text{máx}\{u^p(b - By) : p = 1, \dots, P\} \\ \text{sujeto a} \quad & r^q(b - By) \leq 0, \quad q = 1, \dots, Q, \\ & y \in Y. \end{aligned}$$

O, utilizando una variable auxiliar  $z$  continua:

$$\text{Minimizar} \quad dy + z \tag{2.9}$$

$$\text{sujeto a} \quad z \geq u^p(b - By), \quad p = 1, \dots, P, \tag{2.10}$$

$$r^q(b - By) \leq 0, \quad q = 1, \dots, Q, \tag{2.11}$$

$$y \in Y. \tag{2.12}$$

Esta formulación es llamada la *reformulación de Benders*. Su desventaja es que en la práctica, el número de puntos extremos y direcciones extremas suele ser muy grande. Para evitar este problema, Benders propuso considerar en un principio solo la restricción (2.12), lo cual lleva al primer problema maestro (la inicialización del algoritmo):

$$\begin{aligned} \text{Minimizar} \quad & dy \\ \text{sujeto a} \quad & y \in Y. \end{aligned}$$

Una vez resuelto este problema, la solución obtenida  $y$  es utilizada en el subproblema (2.7). Al resolver el subproblema pueden presentarse dos casos:

- Solución no acotada. En este caso, debemos introducir una restricción del tipo (2.11) en el problema maestro.
- Solución acotada (alcanzada en un punto extremo). En este caso, las variables obtenidas en el problema maestro y en el problema primal del subproblema constituyen una solución del problema original y la solución dual del subproblema es utilizada para generar una restricción de la forma (2.10), que incluimos en el problema maestro.

El problema maestro y el subproblema son resueltos de forma iterada hasta que las

cotas superiores e inferiores sean suficientemente cercanas, teniendo en cuenta que:

- Cota superior: corresponde al valor de la función objetivo del problema original.
- Cota inferior: corresponde al valor de la función objetivo del problema maestro.

El esquema general del algoritmo de la descomposición de Benders, sería, por tanto:

1. Obtenemos un vector  $y^0$  de inicialización resolviendo el problema maestro inicial:

$$\begin{aligned} & \text{Minimizar} && dy \\ & \text{sujeto a} && y \in Y. \end{aligned}$$

El valor objetivo obtenido es una cota inferior del problema original.

2. Se resuelve el subproblema (2.7) utilizando el vector obtenido en el paso previo.
3. Si la solución es no acotada debemos añadir al problema maestro una restricción del tipo (2.11): sea  $r^0$  una dirección tal que  $r^0(b - By) > 0$ , entonces consideramos el corte

$$r^0(b - By) \leq 0.$$

En caso contrario, sean  $x^0$  y  $u^0$  las soluciones óptimas del problema primal y dual, respectivamente, del subproblema. El valor  $dy^0 + cx^0$  constituye una cota superior del problema original. Si las cotas superior e inferior se consideran suficientemente próximas, termina el algoritmo. Si no, construir el corte de la forma (2.10):

$$z \geq u^0(b - By).$$

4. Se añade el corte construido al problema maestro y se resuelve, obteniendo una cota inferior del problema original.
5. Volver al segundo paso, resolviendo el subproblema (2.7) con el vector  $y$  obtenido en el problema maestro y cambiando el superíndice 0 por el correspondiente. En cada iteración, el problema maestro será un problema lineal entero de la forma (2.9) - (2.12). Si en alguna iteración el subproblema es infactible, se detiene el algoritmo puesto que no se pueden generar más cortes.

## 2.2. Variantes de la descomposición de Benders

Se han propuesto muchas extensiones para el método de descomposición de Benders. Una de las más importantes es la propuesta por Geoffrion [1], quien sugiere una “descomposición de Benders generalizada”. Geoffrion hace uso de la teoría de la dualidad

no lineal y extiende el método de Benders al caso donde el subproblema es un problema convexo de optimización. Este avance permite la aplicación de la descomposición de Benders a todo un nuevo conjunto de problemas, en particular a aquellos donde el problema original es no convexo pero puede transformarse en convexo fijando un conjunto de variables. Los valores para estas variables son, por tanto, obtenidos en el problema maestro y fijadas en el subproblema, el cual se convierte en un problema convexo, para el que existen diversos métodos de resolución.

Magnanti y Wong [2] estudiaron el efecto de los cortes en la descomposición de Benders, probando que el uso de cortes más fuertes, cuando es posible, puede tener un gran impacto en la convergencia del algoritmo, reduciendo significativamente el número de iteraciones. La idea es simple: muchas veces la solución del subproblema (2.7) no es única. En efecto, en los problemas de diseño de redes esto suele ser cierto dado que los subproblemas son variaciones de la *ruta más corta*, *el problema del transporte* y otros problemas de optimización conocidos por su degeneración. Para determinar cuál es el mejor corte los autores utilizaron la noción de *dominancia*:

**Definición 2.2.1.** Dada la formulación (2.9)-(2.12), se dice que un corte generado por el punto extremo  $u^1$  **domina** a un corte generado por el punto extremo  $u^2$  si

$$u^1(b - By) \geq u^2(b - By)$$

para todo  $y \in Y$  con alguna desigualdad estricta en al menos un punto. Un corte que no es dominado por ningún otro se denomina **Pareto-óptimo**.

**Definición 2.2.2.** Dado un conjunto  $S \subseteq \mathbb{R}^n$ , se define su **cierre afín** como

$$afin(S) = \left\{ \sum_{i=1}^k \lambda_i x_i : x_i \in S, \sum_{i=1}^k \lambda_i = 1 \right\}$$

**Definición 2.2.3.** Sea  $S \subseteq \mathbb{R}^n$  un conjunto cuyo cierre afín es  $afin(S)$ . El **interior relativo** de  $S$  se define como el conjunto de puntos  $x \in S$  tales que existe  $r > 0$  con  $B(x, r) \cap afin(S) \subset S$ .

Sea  $Y^{LP}$  el poliedro obtenido de la relajación lineal de  $Y$  y sea  $ri(Y^{LP})$  el interior relativo de  $Y^{LP}$ . El siguiente problema lleva a un corte Pareto-óptimo para el problema general (2.1)-(2.5):

$$\begin{aligned}
&\text{Maximizar} && u(b - By^o) \\
&\text{sujeto a} && uA \leq c \\
&&& v(\bar{y}) = u(b - B\bar{y}), \\
&&& u \geq 0,
\end{aligned}$$

donde  $y^o \in ri(Y^{LP})$  y  $v(\bar{y})$  es el valor óptimo del subproblema (2.7) cuando las variables  $y$  son fijadas en  $\bar{y}$ . La función objetivo maximiza la fuerza del corte para  $y^o$  mientras las restricciones definen el espacio factible como las soluciones óptimas del subproblema original.

Los autores también analizaron cómo influye el uso de diferentes formulaciones en el rendimiento de un algoritmo de descomposición de Benders. La conclusión es que dos formulaciones distintas para un problema pueden llevar a rendimientos muy diferentes. Formulaciones más estrictas conducen a cortes de Benders más fuertes. Sin embargo, añadir restricciones en la formulación, aunque mejora la calidad de los cortes, puede complicar la solución de los subproblemas lineales, incrementando el coste computacional.

En trabajos posteriores, Magnanti et al. [3] extendieron sus estudios al contexto de los *problemas de diseño de redes sin capacidad*. Mostraron lo general que es la descomposición de Benders probando que algunos cortes conocidos [4] son en realidad cortes de Benders para elecciones particulares de las variables enteras (posibles configuraciones de la red). Además, mostraron que en este caso particular, pueden generarse cortes Pareto-óptimos resolviendo  $k$  problemas de mínimo coste de flujo (uno para cada mercancía).

## Capítulo 3

# Algunos problemas de diseño de redes con carga fija

Los problemas de diseño de redes consisten en la selección de arcos en un grafo para satisfacer, con coste mínimo, algunas condiciones de flujo, comúnmente expresadas en forma de demandas, con ciertos orígenes y ciertos destinos. El *problema de diseño de redes con carga fija* (FNDP, *Fixed Charge Network Design Problem*) tiene la particularidad de que cada arco tiene un coste fijo asociado, el cual debe ser pagado si el arco forma parte de la solución (lo habitual es que sea un coste de construcción o mantenimiento).

En este capítulo presentamos algunos algoritmos de Benders propuestos para el FNDP. Los organizaremos en función de los diferentes problemas, primero caracterizamos los modelos en función de si tienen o no capacidad de flujo limitada y después estudiaremos los problemas más comunes dentro de cada categoría.

Denotaremos por  $G(N, A, K)$  un grafo dirigido donde:

- $N$  es el conjunto de nodos
- $A \subseteq N \times N$  es el conjunto de arcos o aristas
- $K$  es el conjunto de mercancías a transportar (personas, bienes, datos, etc.)

Por otra parte, utilizaremos la siguiente notación:

- Las variables  $x_{ij}^k$  representan el flujo de la mercancía  $k$  a través del arco  $(i, j)$ .
- En el caso sin capacidad:

$$y_{ij} = \begin{cases} 1 & \text{si el arco } (i, j) \text{ es seleccionado} \\ 0 & \text{si el arco } (i, j) \text{ no es seleccionado} \end{cases}$$

En el caso en el que la capacidad es limitada, es decir, cuando hay límite en el flujo que puede circular a través de los arcos, las variables  $y_{ij}$  representan la capacidad del arco  $(i, j)$ .

### 3.1. Problemas de diseño de redes con capacidad ilimitada

Primero consideraremos problemas de diseño de redes sin capacidad limitada o UNDP (*uncapacitated network design problems*). En estos problemas, no hay límite en el flujo que puede circular a través de los arcos seleccionados.

#### 3.1.1. Origen-destino UNDP

Comenzamos con el problema de diseño de redes origen-destino. En este caso, a cada mercancía  $k = 1, \dots, |K|$  se le asocia una demanda  $d_k$ , un nodo origen  $O(k)$  y un nodo destino  $D(k)$ . En algunos casos, los costes fijos representan completamente el coste real de la red, es decir, no hay costes asociados al volumen de la mercancía que fluye por un arco. Sin embargo, a veces para modelar adecuadamente algunos problemas es necesario establecer costes unitarios de transporte.

#### Formulación origen-destino UNDP ( $F_1$ )

$$\text{Minimizar} \quad \sum_{(i,j) \in A} \left( \sum_{k \in K} c_{ij}^k x_{ij}^k + f_{ij} y_{ij} \right) \quad (3.1)$$

$$\text{sujeto a} \quad \sum_{j|(i,j) \in A} x_{ij}^k - \sum_{j|(j,i) \in A} x_{ji}^k = \begin{cases} d_k, & i = O(k), & \forall k \in K, \\ 0, & i \notin \{O(k), D(k)\}, & \forall k \in K, \\ -d_k, & i = D(k), & \forall k \in K, \end{cases} \quad (3.2)$$

$$\begin{aligned} x_{ij}^k &\leq d_k y_{ij}, \quad \forall (i, j) \in A, \forall k \in K, \\ x_{ij}^k &\geq 0, \quad \forall (i, j) \in A, \forall k \in K, \\ y_{ij} &\in \{0, 1\}, \quad \forall (i, j) \in A. \end{aligned} \quad (3.3)$$

donde

- Las variables  $y_{ij}$  determinan la construcción del arco  $(i, j) \in A$ :  $y_{ij} = 1$  si  $(i, j)$  pertenece a la solución final y  $y_{ij} = 0$  en caso contrario.
- Denotamos por  $c_{ij}^k$  al coste unitario de transporte de la mercancía  $k$  a través del arco  $(i, j)$ .



- Denotamos por  $f_{ij}$  al coste fijo del arco  $(i, j)$ , es decir, el coste que supone construir este arco.

Por otra parte,

- La función objetivo (3.1) es la suma de los costes fijos  $f_{ij}$  y de transporte  $c_{ij}^k$ .
- El conjunto de las restricciones (3.2) aseguran que el flujo de la mercancía  $k$  parte del nodo origen  $O(k)$  y llega al nodo destino  $D(k)$ . Se denominan restricciones de balance de flujo.
- Las restricciones (3.3) limitan el flujo de la mercancía a los arcos seleccionados.

Las restricciones (3.3) pueden ser sustituidas mediante restricciones agrupadas de la forma:

$$\sum_{k \in K} x_{ij}^k \leq \left( \sum_{k \in K} d_k \right) y_{ij}, \quad \forall (i, j) \in A. \quad (3.4)$$

Este modelo, menos restrictivo que el anterior, supone una relajación que puede reducir sustancialmente el número de restricciones. Sin embargo, como indicamos anteriormente, los cortes de Benders obtenidos con formulaciones desagrupadas (mayor número de restricciones) suelen ser más eficientes que los obtenidos con restricciones agrupadas [3].

### Aplicación de la descomposición de Benders

El caso básico sin límite de capacidad fue resuelto en uno de los primeros trabajos de Magnanti et al. [3]. En este trabajo, el problema maestro propone una red provisional a través de las variables de diseño y el subproblema determina el flujo óptimo bajo esta red provisional. Los autores usaron un conjunto de casos de hasta 30 nodos y 130 arcos para probar que:

1. El uso de técnicas auxiliares puede mejorar el rendimiento de la descomposición de Benders.
2. Una elección inteligente de cortes, específicamente, los cortes Pareto-óptimos, pueden afectar fuertemente al rendimiento del algoritmo.

El trabajo de Magnanti et al. fue extendido diez años después por Gutierrez et al. [5], donde proponen un modelo robusto capaz de tener en cuenta cierta incertidumbre en los costes de transporte  $c_{ij}^k$  y en las demandas  $d_k$ . Los datos inciertos son descritos a través de un conjunto de escenarios, cada uno con diferentes valores para  $c_{ij}^k$ . Observemos que, puesto que estamos tratando el caso sin capacidad limitada, las modificaciones en  $c_{ij}^k$

pueden reflejar cambios tanto en los propios costes como en las demandas (trabajando con una escala adecuada). Hallamos la solución a través de un algoritmo de Benders *multi-maestro*, donde cada problema maestro individual se asocia con un posible escenario. Cada vez que un problema maestro es resuelto, el subproblema genera un corte para cada uno de los problemas maestros.

### 3.1.2. Origen único UNDP

Un importante caso particular del problema origen-destino es el caso en el que todos los puntos de demanda deben estar conectados a un origen único (*local access network design*).

Esta es la situación, por ejemplo en el diseño de redes de área local (*local area network, LAN*). Cuando la función objetivo está únicamente definida por costes de transporte entonces el diseño de una red de área local puede ser formulado como un problema de transporte con un único origen. Si la función objetivo estuviera definida solo por costes fijos el problema se convierte en un *problema de Steiner* NP-duro o en hallar un árbol de unión de coste mínimo (*minimum spanning tree problem*) si todos los nodos deben ser alcanzados. En este último caso, una importante situación tiene lugar cuando el número de arcos conectados directa o indirectamente con el origen es limitado (*problema del árbol de unión de coste mínimo con capacidad limitada*), aunque clasificamos este problema como con capacidad ilimitada dado que el flujo sobre los arcos no tiene límite.

Presentamos una formulación para el caso general, en el que cada arco tiene asociados un coste fijo y otro variable.

#### Formulación origen único UNDP ( $F_2$ )

$$\begin{array}{ll}
 \text{Minimizar} & \sum_{(i,j) \in A} \left( \sum_{k \in K} c_{ij}^k x_{ij}^k + f_{ij} y_{ij} \right) \\
 \text{sujeto a} & \sum_{j|(i,j) \in A} x_{ij}^k - \sum_{j|(j,i) \in A} x_{ji}^k = \begin{cases} d_k, & i = O, & \forall k \in K, \\ 0, & i \notin \{O, D(k)\}, & \forall k \in K, \\ -d_k, & i = D(k), & \forall k \in K, \end{cases} \\
 & x_{ij}^k \leq d_k y_{ij}, \quad \forall (i,j) \in A, \forall k \in K,
 \end{array}$$

$$\begin{aligned} x_{ij}^k &\geq 0, \quad \forall (i, j) \in A, \forall k \in K, \\ y_{ij} &\in \{0, 1\}, \quad \forall (i, j) \in A. \end{aligned}$$

En  $(F_2)$  consideramos la demanda en cada nodo como una mercancía distinta, con diferentes costes de transporte  $c_{ij}^k$  que dependen de los arcos y de la mercancía. El nodo  $O$  es el nodo origen. El sentido de esta formulación es análogo al explicado para  $(F_1)$ .

Salvo por el hecho de que todas las mercancías parten de un mismo nodo,  $(F_2)$  es idéntica a  $(F_1)$ . La presencia de un nodo origen único lleva a una formulación simplificada en el caso en el que los costes variables sean independientes de las mercancías:

### Formulación origen único, mercancía única UNDP $(F'_2)$

$$\text{Minimizar} \quad \sum_{(i,j) \in A} c_{ij} x_{ij} + f_{ij} y_{ij} \quad (3.5)$$

$$\text{sujeto a} \quad \sum_{j|(i,j) \in A} x_{ij} - \sum_{j|(j,i) \in A} x_{ji} = \begin{cases} D, & i = O, \\ -d_i, & i \neq O, \end{cases} \quad (3.6)$$

$$\begin{aligned} x_{ij} &\leq D y_{ij}, \quad \forall (i, j) \in A, \\ x_{ij} &\geq 0, \quad \forall (i, j) \in A, \\ y_{ij} &\in \{0, 1\}, \quad \forall (i, j) \in A. \end{aligned} \quad (3.7)$$

El parámetro  $d_i$  representa la demanda de cada nodo y  $D$  es la suma de todas las demandas excluyendo el origen. La función objetivo (3.5) de la formulación  $(F'_2)$  sigue minimizando la suma de los costes fijos y de transporte. Sin embargo, en este caso no es necesario diferenciar las mercancías. El único requisito es que del origen salga la cantidad correcta de mercancía y que a cada nodo llegue la demandada, lo cual imponemos en las restricciones de balance de flujo (3.6). Las restricciones (3.7) prohíben el flujo en los arcos que no formen parte del diseño.

### Aplicación de la descomposición de Benders

En la práctica es conveniente hallar una inicialización adecuada. Por ejemplo, en Randazzo y Luna [6], se utiliza una relajación lineal y un algoritmo del *camino más corto* para obtener una solución factible, la cual es usada para generar valores iniciales para las variables duales. En el problema maestro se añaden restricciones redundantes pero que contribuyen a incrementar la probabilidad de generar una solución factible, y

el subproblema es descompuesto en  $k$  problemas de flujo de red fáciles, uno para cada mercancía.

En un trabajo más reciente, Gavish [7] presentó un algoritmo de descomposición de Benders para el problema del *árbol de unión de coste mínimo con capacidad limitada*. El algoritmo es bastante estándar, el problema maestro fija las variables enteras de diseño y tiene lugar una de las siguientes situaciones:

- Los arcos seleccionados constituyen un árbol de unión. Entonces, o bien la solución es óptima (si el número de arcos conectados al origen respeta el límite) o bien los arcos conectados al origen pueden utilizarse para generar un corte y obtener una nueva iteración.
- El grafo no está conectado. En este caso, se puede generar un corte y resolverse el problema maestro de nuevo.

### 3.1.3. Origen único, dos tecnologías UNDP

Una tercera situación común en los problemas de diseño de redes con capacidad ilimitada es el caso en el que hay disponibles dos tecnologías distintas, cada una con sus correspondientes ventajas. Por ejemplo, una tecnología puede tener unos costes fijos mayores pero unos costes de transporte menores que la otra y viceversa (como ocurre con la fibra óptica y el cobre en redes de telecomunicaciones).

La formulación es una extensión de  $(F_2)$  con la duplicación de las variables  $x_{ij}^k$  e  $y_{ijt}$ . Así, en  $(F_3)$ ,  $y_{ijt}$  es la variable binaria asociada a la construcción del arco  $(i, j)$  con la tecnología  $t$ . Análogamente,  $x_{ijt}^k$  es el flujo de la mercancía  $k$  a través de ese arco.

#### Formulación origen único, dos tecnologías UNDP $(F_3)$

$$\text{Minimizar} \quad \sum_{(i,j) \in A} \sum_{t=1}^2 \left( \sum_{k \in K} c_{ijt}^k x_{ijt}^k + f_{ijt} y_{ijt} \right) \quad (3.8)$$

$$\text{sujeto a} \quad \sum_{t=1}^2 \left( \sum_{j|(i,j) \in A} x_{ijt}^k - \sum_{j|(j,i) \in A} x_{jit}^k \right) = \begin{cases} d_k, & i = O, \forall k \in K, \\ 0, & i \notin \{O, D(k)\}, \forall k \in K, \\ -d_k, & i = D(k), \forall k \in K, \end{cases} \quad (3.9)$$

$$x_{ijt}^k \leq d_k y_{ijt}, \quad \forall (i, j) \in A, \forall k \in K, t = 1, 2, \quad (3.10)$$

$$\sum_{t=1}^2 y_{ijt} \leq 1, \quad \forall (i, j) \in A, \quad (3.11)$$

$$x_{ijt}^k \geq 0, \quad \forall (i, j) \in A, \forall k \in K, t = 1, 2,$$

$$y_{ijt} \in \{0, 1\}, \quad \forall (i, j) \in A, t = 1, 2.$$

En  $(F_3)$  el objetivo (3.8) es minimizar la suma de los costes fijos y de transporte para ambas tecnologías. Se deben respetar las cantidades demandadas (3.9) y, para cada tecnología, solo puede haber flujo si ese arco está abierto (3.10). Además, cada arco utiliza una única tecnología (3.11).

Observamos que  $(F_3)$  es la formulación más completa de las cuatro presentadas. De hecho,  $(F_1)$ ,  $(F_2)$  y  $(F_2')$  pueden ser considerados casos particulares de  $(F_3)$ . Además, se puede adaptar fácilmente al caso en el que hay más de dos tecnologías disponibles.

### Aplicación de la descomposición de Benders

Randazzo et al. [8] utilizan la descomposición de Benders para resolver este problema. El algoritmo es muy similar al presentado en Randazzo y Luna [6] pero sin usar relajación lineal. La primera solución factible se obtiene a través de un algoritmo del *camino más corto* considerando solo los costes de transporte. Esta solución inicial es usada para generar cotas superiores e inferiores iniciales. Se añaden restricciones en el problema maestro con el objetivo de garantizar una solución con estructura de árbol. El subproblema vuelve a descomponerse en problemas de flujo de red fáciles, uno para cada mercancía.

## 3.2. Problemas de diseño de redes con capacidad limitada

Consideramos ahora problemas de diseño de redes con capacidad limitada o CNDP (*capacitated network design problems*). Como antes, el flujo en un arco está permitido solo si se paga su coste fijo asociado. En este caso la capacidad de flujo de los arcos es limitada. Analizamos algunos modelos en las siguientes subsecciones.

### 3.2.1. Una única tecnología CNDP

En este caso hay una única tecnología, la cual puede instalarse varias veces en cada arco. Cada unidad instalada tiene un coste y permite una cierta cantidad de flujo en ese arco.

### Formulación instalación única CNDP ( $F_4$ )

$$\begin{aligned}
 & \text{Minimizar} && \sum_{(i,j) \in A} \left( \sum_{k \in K} c_{ij}^k x_{ij}^k + f_{ij} y_{ij} \right) \\
 & \text{sujeto a} && \sum_{j|(i,j) \in A} x_{ij}^k - \sum_{j|(j,i) \in A} x_{ji}^k = \begin{cases} d_k, & i = O(k), & \forall k \in K, \\ 0, & i \notin \{O(k), D(k)\}, & \forall k \in K, \\ -d_k, & i = D(k), & \forall k \in K, \end{cases} \\
 & && \sum_{k \in K} x_{ij}^k \leq C y_{ij}, \quad \forall (i, j) \in A, & (3.12) \\
 & && x_{ij}^k \geq 0, \quad \forall (i, j) \in A, \forall k \in K, & (3.13) \\
 & && y_{ij} \in \mathbb{Z}_+, \quad \forall (i, j) \in A.
 \end{aligned}$$

En ( $F_4$ ) hay dos importantes modificaciones con respecto a ( $F_1$ ). En primer lugar, hay una única restricción del tipo (3.12) para cada arco (en lugar de  $k$  restricciones por arco). Estas restricciones agrupan las variables  $x_{ij}^k$  como en las restricciones (3.4). Además, en el lado derecho de (3.12),  $d_k$  se cambia por  $C$ , la capacidad que obtenemos al adquirir una unidad de la instalación. La segunda modificación es que las restricciones (3.13) indican que las variables  $y_{ij}$  son enteras. Estas dos modificaciones permiten seleccionar un número entero de unidades a instalar en cada arco de forma individual, proporcionando al arco una capacidad total de  $C y_{ij}$ .

Sridhar y Park [9] trabajan con una versión de este problema que no tiene costes de transporte. Además de las restricciones que limitan la capacidad de los arcos, su modelo incluye también restricciones de capacidad para los nodos, de la forma

$$\sum_{j|(i,j) \in A} \sum_{k \in K} x_{ij}^k + \sum_{k|D(k)=i} d_k \leq \kappa_i \quad \forall i \in N,$$

indicando que la capacidad  $\kappa_i$  del nodo  $i$  debe ser mayor o igual que la cantidad de flujo que pasa a través de él, que se origina o que llega a él. La capacidad  $\kappa_i$  es un parámetro a determinar en función del rendimiento de la red deseado.

### 3.2.2. Varias tecnologías CNDP

Este problema es el análogo al presentado en la sección 3.1.3 en el caso de capacidad limitada (pero aquí no suponemos origen único). Además de elegir la tecnología a utilizar en cada arco, debemos también elegir la capacidad a instalar. Esta capacidad debe venir

dada en múltiplos de la capacidad que proporciona instalar una unidad de la tecnología elegida. Así, dadas dos tecnologías con diferentes costes fijos unitarios  $f_{ij1}$  y  $f_{ij2}$  con diferentes capacidades unitarias  $C_1$  y  $C_2$ , la formulación  $(F_5)$  modela el problema de encontrar la red más barata que cumple los requisitos de demanda, cuando cada demanda tiene asociado un destino y un origen que no tiene por qué ser el mismo para todas.

### Formulación dos instalaciones CNDP $(F_5)$

$$\text{Minimizar} \quad \sum_{(i,j) \in A} \sum_{t=1}^2 \left( \sum_{k \in K} c_{ijt}^k x_{ijt}^k + f_{ijt} y_{ijt} \right) \quad (3.14)$$

$$\text{sujeto a} \quad \sum_{t=1}^2 \left( \sum_{j|(i,j) \in A} x_{ijt}^k - \sum_{j|(j,i) \in A} x_{jit}^k \right) = \begin{cases} d_k, & i = O(k), \forall k \in K, \\ 0, & i \notin \{O(k), D(k)\}, \forall k \in K, \\ -d_k, & i = D(k), \forall k \in K, \end{cases} \quad (3.15)$$

$$\sum_{k \in K} x_{ijt}^k \leq C_t y_{ijt}, \quad \forall (i, j) \in A, t = 1, 2, \quad (3.16)$$

$$y_{ijt} \leq U_t z_{ijt}, \quad \forall (i, j) \in A, t = 1, 2, \quad (3.17)$$

$$\sum_{t=1}^2 z_{ijt} \leq 1, \quad \forall (i, j) \in A, \quad (3.18)$$

$$x_{ijt}^k \geq 0, \quad \forall (i, j) \in A, \forall k \in K, t = 1, 2,$$

$$y_{ijt} \in \mathbb{Z}_+, \quad \forall (i, j) \in A, t = 1, 2.$$

$$z_{ijt} \in \{0, 1\}, \quad \forall (i, j) \in A, t = 1, 2.$$

Las variables  $z_{ijt}$  son necesarias para limitar el número de tecnologías en un arco.

$$z_{ijt} = \begin{cases} 1 & \text{si el arco } (i, j) \text{ tiene tecnología } t \text{ instalada} \\ 0 & \text{en caso contrario} \end{cases}$$

El coeficiente  $U_t$  es el máximo número de instalaciones de la tecnología  $t$  posibles en un arco.

La función objetivo (3.14) minimiza la suma de los costes fijos y de transporte para ambas tecnologías. Los requisitos de demanda se cumplen gracias a las restricciones (3.15). Las restricciones (3.16) imponen los límites de capacidad y las restricciones (3.17) y (3.18) aseguran que solo un tipo de tecnología puede ser instalada en cada arco.

Magnanti et al. [10] proponen un caso muy similar (que aplican al diseño de redes de comunicaciones privadas) en el que ambas tecnologías pueden coexistir en un mismo arco, sumando las capacidades que aportan cada una. Así, no son necesarias las variables  $z_{ijt}$ , no es necesario el índice  $t$  en las variables  $x_{ij}^k$  y las restricciones (3.16) se pueden sustituir por:

$$\sum_{k \in K} x_{ij}^k \leq C_1 y_{ij1} + C_2 y_{ij2} \quad \forall (i, j) \in A.$$

Al igual que en la sección anterior, se puede convertir ( $F_5$ ) en un problema de *origen único* a través de modificaciones análogas a las del caso de capacidad ilimitada.

### 3.2.3. CNDP con coste de capacidad escalonado

El último caso y el más general es el problema de capacidad limitada en el que el coste de adquirir capacidad de flujo (el coste de realizar la instalación) para un arco viene dado por una función  $Coste(t)$  escalonada no decreciente que depende de la capacidad:

$$Coste : T \subset \mathbb{N} \longrightarrow \mathbb{R}_+$$

Así, si asociamos cada  $t \in T$  a una cierta capacidad  $C_t$  y suponemos que  $C_{t+1} \geq C_t$  para cada  $t, t+1 \in T$  para que el modelo tenga sentido, el coste de adquirir una capacidad  $C_{t+1}$  es mayor o igual que el coste de adquirir la capacidad  $C_t$ , es decir,  $Coste(t+1) \geq Coste(t)$ .

Utilizamos la formulación ( $F_6$ ) para modelar este caso, en el que debido a las discontinuidades de la función de coste, es necesario crear variables binarias adicionales. En esta formulación,  $C_t$  es la capacidad permitida en un arco si el arco se encuentra en el nivel  $t$  de la función de coste.

#### Formulación del CNDP con coste de capacidad escalonado ( $F_6$ )

$$\text{Minimizar} \quad \sum_{(i,j) \in A} \left( \sum_{k \in K} c_{ij}^k x_{ij}^k + \sum_{t=1}^T f_{ijt} y_{ijt} \right) \quad (3.19)$$

$$\text{sujeto a} \quad \sum_{j|(i,j) \in A} x_{ij}^k - \sum_{j|(j,i) \in A} x_{ji}^k = \begin{cases} d_k, & i = O(k), \forall k \in K, \\ 0, & i \notin \{O(k), D(k)\}, \forall k \in K, \\ -d_k, & i = D(k), \forall k \in K, \end{cases} \quad (3.20)$$



$$\sum_{k \in K} x_{ij}^k \leq \sum_{t=1}^T C_t y_{ijt}, \quad \forall (i, j) \in A, \quad (3.21)$$

$$\sum_{t=1}^T y_{ijt} \leq 1, \quad \forall (i, j) \in A, \quad (3.22)$$

$$x_{ij}^k \geq 0, \quad \forall (i, j) \in A, \forall k \in K,$$

$$y_{ijt} \in \{0, 1\}, \quad \forall (i, j) \in A, t = 1, \dots, T.$$

De nuevo, la función objetivo (3.19) consiste en minimizar los costes fijos y de transporte. Las restricciones (3.20) garantizan que se satisfacen los requisitos de demanda mientras que las restricciones (3.21) limitan el flujo en cada arco de acuerdo con la capacidad adquirida. Las restricciones (3.22) imponen que solo una variable  $y_{ijt}$  puede ser seleccionada, es decir en cada arco solo es posible instalar una capacidad  $C_t$ , la cual supone un coste  $Coste(t)$ .

Es común expresar la formulación  $(F_6)$  equivalentemente como una formulación basada en caminos. En este caso, sea  $\mathbb{P}(k)$  el conjunto de los posibles caminos entre los nodos  $O(k)$  y  $D(k)$ . El coeficiente  $a_{kp(i,j)}$  vale 1 si la mercancía  $k$  fluyendo en el camino  $p$  utiliza el arco  $(i, j)$  y 0 en caso contrario. Por otra parte, sea  $n_{kp}$  el flujo de mercancía  $k$  en el camino  $p$ . La formulación  $(F'_6)$  modela esta situación.

### Formulación del CNDP con coste de capacidad escalonado $(F'_6)$

$$\begin{aligned} \text{Minimizar} \quad & \sum_{(i,j) \in A} \left( \sum_{k \in K} c_{ij}^k x_{ij}^k + \sum_{t=1}^T f_{ijt} y_{ijt} \right) \\ \text{sujeto a} \quad & x_{ij}^k = \sum_p a_{kp(i,j)} n_{kp}, \quad \forall (i, j) \in A, \forall k \in K, \end{aligned} \quad (3.23)$$

$$\sum_p n_{pk} = d_k \quad \forall k \in K, \quad (3.24)$$

$$\sum_{k \in K} x_{ij}^k \leq \sum_{t=1}^T C_t y_{ijt}, \quad \forall (i, j) \in A, \quad (3.25)$$

$$\sum_{t=1}^T y_{ijt} \leq 1, \quad \forall (i, j) \in A, \quad (3.26)$$

$$n_{kp} \geq 0, \quad \forall p \in \mathbb{P}, \forall k \in K,$$

$$x_{ij}^k \geq 0, \quad \forall (i, j) \in A, \forall k \in K,$$

$$y_{ijt} \in \{0, 1\}, \quad \forall (i, j) \in A, t = 1, \dots, T.$$

Aquí,  $x_{ij}^k$  sigue representando el flujo en cada arco. Las restricciones (3.23) definen el flujo de la mercancía  $k$  en el arco  $(i, j)$  como la suma de los flujos de esa mercancía en todos los caminos que utilizan ese arco. Las igualdades (3.24) son las restricciones de demanda y las desigualdades (3.25) son las restricciones de capacidad. Las restricciones (3.26) de nuevo imponen que solo se puede seleccionar una variable  $y_{ijt}$ .

Este tipo de formulaciones, en las que se utilizan caminos, pueden utilizarse para generar versiones reducidas del problema de diseño de redes considerando un número limitado de caminos para cada demanda (que tiene un origen y un destino asociado). Aunque hemos puesto el ejemplo con  $(F_6)$ , todas las formulaciones anteriores pueden escribirse equivalentemente de esta forma.

## Capítulo 4

# Una aplicación al rediseño de una red con carga fija y costes de transporte variables

En este capítulo vamos a suponer la existencia de una red de transporte en funcionamiento que se quiere mejorar utilizando recursos limitados mediante la creación de nuevas conexiones (nuevos arcos) y con la mejora de la tecnología de transporte usada en los arcos que existen en la red original. Esta hipótesis de partida se acomoda a multitud de situaciones reales como la que veremos a continuación.

### 4.1. El problema de la red ferroviaria de alta velocidad entre España y Francia

El modelo planteado en este capítulo nos permite abordar desde el punto de vista de la Programación Matemática un problema real reciente: el conflicto de los enlaces entre España y Francia de los corredores de alta velocidad ferroviaria mediterráneo y atlántico.

El 21 de marzo de 2023 el diario ABC publica una noticia bajo el título de “*Francia desconecta a España de la alta velocidad ferroviaria en Europa*” [12] en la que se describe el problema actual que hay en la red de alta velocidad ferroviaria entre España y Francia: el conflicto radica en la decisión del Gobierno galo de aplazar las construcciones de las conexiones ferroviarias de alta velocidad entre Irún y Burdeos hasta 2042, entre Montpellier y Beziers hasta 2035 y entre Beziers y Perpiñán hasta 2042 (véase la figura 4.1). Este retraso supone un aplazamiento sustancial en la puesta en marcha de los corredores mediterráneo y atlántico, que supuestamente ambos Gobiernos acordaron tener concluidos para 2030. El aplazamiento ha provocado un gran malestar entre varios sectores del empresariado español, que preveía poder hacer uso en 2030 de estos corredores para el

envío de mercancía a París de forma eficiente. Los empresarios españoles lo consideran un problema económico de primer orden, calificado, como la clara demostración de que Francia “no tiene ningún interés” en que España y sus exportaciones se conecten con la red europea de alta velocidad. Por otro lado, la Comisión Europea está ya al tanto de la situación y pide al Gobierno galo que se de prioridad a los tramos transfronterizos.



Figura 4.1: Red ferroviaria de alta velocidad de España y Francia

Desde el punto de vista de la Programación Matemática, podríamos plantear un rediseño de la red en el que el aplazamiento de los tramos en cuestión se modela mediante la asignación de costes variables a dichas conexiones. Estos costes pueden modelar diferentes situaciones, desde la construcción de las conexiones según el plan previsto hasta su aplazamiento, como ha ocurrido, lo que implicaría aumentar sustancialmente estos costes.

La figura 4.1 representa la red ferroviaria que está actualmente en funcionamiento

(trazo resaltado) junto con las conexiones potenciales (trazo más suave).

El objetivo sería analizar cómo emplear los recursos existentes en mejorar la eficiencia de la red, en particular intentando conectar España y Francia. Por ejemplo, realizar la conexión Huesca-Toulouse conectaría España con Francia pero a nivel práctico supondría atravesar los Pirineos, lo que vendría reflejado en un elevado coste fijo.

## 4.2. Planteamiento del modelo. Algoritmo de Benders

Dado que el objetivo es mejorar la red, y no rehacerla, supondremos que no podemos eliminar conexiones de la red preexistente, lo que se traducirá en fijar el valor de un subconjunto de las variables de diseño. Además, a efectos prácticos, en múltiples ocasiones resulta poco rentable destruir conexiones ya hechas, como en el caso que se ha descrito de la línea de tren de alta velocidad. No obstante, a efectos de modelado, podemos replicar conexiones existentes de modo que la elección de dichos arcos suponga una inversión en tecnología para la conexión existente (ver sección 4.3), por ejemplo modificando su anchura o peraltado de manera que se pueda aumentar la velocidad del tren.

Por otro lado, también supondremos que desconocemos los costes exactos de transporte pero sí supondremos conocido un conjunto finito de posibles valores con los que se modelan posibles escenarios.

Usamos la misma notación que en la sección precedente salvo con la diferencia de que aquí trabajaremos con un grafo dirigido:  $G(N, A, K)$  denota un grafo dirigido donde  $N$  es el conjunto de nodos,  $A \subset N \times N$  es el conjunto de arcos y  $K$  es el conjunto de mercancías a transportar, que por simplicidad supondremos  $K = \{1\}$ . Las variables  $x_{ij}$  representan el flujo a través del arco  $(i, j)$  y las variables  $y_{ij}$  son las que diseñan la red. En este contexto, adaptaremos el algoritmo de Benders utilizando ideas análogas a las descritas hasta el momento.

Supondremos que las variables de diseño  $y$  son binarias y pertenecen al conjunto  $Y \subset \{0, 1\}^{|A|}$ . En particular  $Y$  fija algunas de las componentes de  $y$  de acuerdo a la red preexistente. En el ejemplo de la red ferroviaria de alta velocidad fijamos  $y_{ij} = 1$  las componentes tales que  $(i, j)$  se correspondiese con una conexión ya construida.

Fijado un vector  $y$ , consideramos el siguiente problema de flujo en sus versiones primal ( $P$ ) y dual ( $D$ ):

$$\begin{aligned}
(P) \quad & \text{Minimizar} \quad \sum_{(i,j) \in A} c_{ij} x_{ij} \\
& \text{sujeto a} \quad \sum_{j|(i,j) \in A} x_{ij} - \sum_{k|(k,i) \in A} x_{ki} = d_i \quad \forall i \in N \\
& \quad \quad \quad 0 \leq x_{ij} \leq D y_{ij} \quad \forall (i,j) \in A
\end{aligned}$$

$$\begin{aligned}
(D) \quad & \text{Maximizar} \quad \sum_{i \in N} d_i u_i + \sum_{(i,j) \in A} (D y_{ij}) v_{ij} \\
& \text{sujeto a} \quad u_i - u_j + v_{ij} \leq c_{ij} \quad \forall (i,j) \in A \\
& \quad \quad \quad u_i \text{ s.r.} \quad \forall i \in N \\
& \quad \quad \quad v_{ij} \leq 0 \quad \forall (i,j) \in A
\end{aligned}$$

En este problema, el parámetro  $D$  denota una cota superior sobre la suma de las demandas y lo consideraremos entero. Asumimos que el problema primal de flujo ( $P$ ) es siempre factible ya que en nuestro modelo tomamos como hipótesis la existencia de una red inicial que es capaz de atender las demandas y cuya eficiencia se pretende mejorar. Esta es una hipótesis bastante habitual en problemas prácticos sobre redes logísticas. No obstante, si existiesen problemas de desconexión como en el caso real que describíamos en la red ferroviaria de alta velocidad entre España y Francia, se podría solventar imponiendo restricciones de conexión en el conjunto  $Y$  de posibles diseños (con condiciones del tipo (2.3)).

En esta situación resolver el problema dual ( $D$ ) siempre determina un punto extremo óptimo, es decir, solo consideramos cortes de Benders de la forma (2.10).

**Definición 4.2.1.** La **matriz de incidencia** de un grafo dirigido se construye asignando un nodo a cada fila y una columna a cada arco. En la columna bajo el arco  $(i, j)$  se coloca un 1 en la fila correspondiente al nodo  $i$  y un -1 en la del nodo  $j$ .

**Observación 4.2.1.** Obsérvese que un grafo dirigido queda completamente representado por su matriz de incidencia y que en el caso del problema de flujo ( $P$ ), la matriz correspondiente al poliedro que definen las  $i$  primeras restricciones coincide con la matriz de incidencia del grafo  $G(N, A, K)$ .

En los siguientes resultados consideraremos  $m$  y  $n$  enteros positivos.

**Definición 4.2.2.** Una matriz entera  $A \in \mathcal{M}^{m \times n}$  es **totalmente unimodular** (TU) si el determinante de cualquier submatriz cuadrada de  $A$  es igual a 0, 1 ó -1.

**Proposición 4.2.1.** Sea  $A \in \mathcal{M}^{m \times n}$ . Son equivalentes:

1.  $A$  es TU.
2.  $A^T$  es TU.
3.  $[A|I]$  es TU.

**Demostración.** Las operaciones que implican calcular  $A^T$  y  $[A|I]$  preservan el valor de los determinantes.  $\square$

**Proposición 4.2.2 (Condición suficiente).** Una matriz  $A \in \mathcal{M}^{m \times n}$  es TU si

1.  $a_{ij} \in \{+1, -1, 0\}$  para cada  $i, j$ .
2. Cada columna contiene a lo sumo dos elementos distintos de 0.
3. Existe una partición  $(M_1, M_2)$  del conjunto  $M$  de filas tal que cada columna  $j$  con dos elementos no nulos verifica

$$\sum_{i \in M_1} a_{ij} - \sum_{i \in M_2} a_{ij} = 0.$$

**Demostración.** Puede encontrarse en Wolsey L.A. Proposición 3.2 [14].  $\square$

**Teorema 4.2.1 (Teorema de unimodularidad).** Sea  $A \in \mathcal{M}^{m \times n}$  una matriz con entradas enteras y filas linealmente independientes. Entonces las siguientes tres condiciones son equivalentes:

1.  $A$  es unimodular.
2. Toda solución básica factible definida por las restricciones  $Ax = b$ ,  $x \geq 0$ , es entera para cualquier vector entero  $b$ .
3. Toda matriz básica  $B$  de  $A$  tiene una matriz inversa de entradas enteras  $B^{-1}$ .

**Demostración.** Puede encontrarse en Ahuja et al. Teorema 11.11 [13].  $\square$

**Proposición 4.2.3.** La matriz de restricciones  $A \in \mathcal{M}^{(|N|+|A|) \times |A|}$  del problema  $(P)$  es totalmente unimodular.

**Demostración.** La matriz  $A$  es de la forma  $\begin{pmatrix} C \\ I \end{pmatrix}$ , donde  $C$  es la matriz definida por las restricciones asociadas a la conservación del flujo (que coincide con la matriz de incidencia del grafo  $G(N, A, K)$ ) e  $I$  la matriz definida por las restricciones asociadas a los

límites inferiores y superiores de flujo. Por la proposición 4.2.1 es suficiente probar que  $C$  es TU: las condiciones suficientes de la proposición 4.2.2 se satisfacen con  $M_1 = M$  y  $M_2 = \emptyset$ .  $\square$

Estos resultados justifican que si el problema  $(P)$ , que es un problema de flujo, es factible entonces tiene alguna una solución entera. Además permiten el uso del problema dual  $(D)$  incluso cuando se exige la integridad de las variables de flujo.

El problema original que resolvemos, que denotaremos por  $(P^*)$ , tiene como variables de decisión las de diseño, las de flujo y además los costes  $c_{ij}$  suponemos que son también variables. Por ejemplo, podemos modelar que una de las conexiones sufre retraso en su construcción mediante un coste de transporte muy elevado, de esta forma repercutiríamos los costes por demora.

Una vez elegido un diseño factible  $y \in Y$ , el modelo evalúa la red mediante el coste de carga fija al que se añade el menor coste de transporte que sería necesario en el caso en que se diese el peor escenario posible. Por ejemplo, podríamos tomar como costes  $c_{ij}$  los costes promedios para un cierto periodo y elevar sustancialmente dichos costes en el caso de no estar disponible la conexión correspondiente durante parte de dicho periodo (como posiblemente ocurrirá con los retrasos en las fechas acordadas en el problema de la red ferroviaria de alta velocidad).

Formulamos este problema como:

$$\begin{aligned}
 (P^*) \quad & \text{Minimizar} \quad \sum_{(i,j) \in A} f_{ij} y_{ij} + \max_{c \in C} \min \sum_{(i,j) \in A} c_{ij} x_{ij} \\
 & \text{sujeto a} \quad \sum_j x_{ij} - \sum_k x_{ki} = d_i \quad \forall i \in N, \\
 & \quad \quad \quad 0 \leq x_{ij} \leq D y_{ij} \quad \forall (i,j) \in A, \\
 & \quad \quad \quad y \in Y.
 \end{aligned}$$

Esta forma de modelar la función objetivo supone que, dado que los costes de transporte son desconocidos, podemos tomar decisiones ‘conservadoras’ evaluándolas mediante su coste en el peor de los casos, y por tanto en la función objetivo consideramos el mayor coste que se pueda dar. Asumimos que  $C$  se puede modelar con restricciones lineales en  $c_{ij}$ ,  $(i,j) \in A$  y que es acotado. Los costes de carga fijos  $f_{ij}$  para cada arco  $(i,j) \in A$  se suponen conocidos, al igual que la demanda  $d_i$  en cada nodo  $i \in N$ .

También podemos incluir en el modelo restricciones lineales para modelar condiciones técnicas, por ejemplo:



- $\sum_{(i,j) \in A} y_{ij} \leq (\geq) m$ ,  $m \in \mathbb{Z}$ , restricción que representa una cota superior (inferior) sobre el número de nuevas conexiones. Este tipo de restricción podría ayudar a exigir que no se produzca la desconexión de parte de la red.
- $y_{ij} \leq y_{pq}$ , restricción que fuerza a construir la conexión  $(p, q)$  en el caso de que se construya la conexión  $(i, j)$ .

Denotaremos por  $(P^*)'$  a la parte interior del problema:

$$\begin{aligned}
 (P^*)' \quad & \max_{c \in C} \quad \min \sum_{(i,j) \in A} c_{ij} x_{ij} \\
 \text{sujeto a} \quad & \sum_{j|(i,j) \in A} x_{ij} - \sum_{k|(k,i) \in A} x_{ki} = d_i \quad \forall i \in N, \\
 & 0 \leq x_{ij} \leq D y_{ij} \quad \forall (i, j) \in A.
 \end{aligned}$$

Usando el dual  $(D)$  podemos reformular  $(P^*)'$  como sigue:

$$\begin{aligned}
 (SP) \quad & \text{Maximizar} \quad \sum_{i \in N} d_i u_i + \sum_{(i,j) \in A} (D y_{ij}) v_{ij} \\
 \text{sujeto a} \quad & u_i - u_j + v_{ij} \leq c_{ij} \quad \forall (i, j) \in A, \\
 & c \in C, \\
 & u_i \text{ s.r.} \quad \forall i \in N, \\
 & v_{ij} \leq 0, \quad \forall (i, j) \in A.
 \end{aligned}$$

**Teorema 4.2.2 (Dualidad fuerte).** *Si  $(P)$  y  $(D)$  son factibles, entonces existen soluciones óptimas de ambos problemas y los valores objetivos óptimos coinciden.*

**Demostración.**

Puede encontrarse en Bazaraa et al. Capítulo 6 [11]. □

**Proposición 4.2.4.** *Los valores objetivos óptimos de  $(P^*)'$  y  $(SP)$  coinciden.*

**Demostración.**

Denotamos por  $h(c)$ , con  $c \in C$ , al valor objetivo óptimo del problema  $(P)$ :

$$\begin{aligned}
(P) \quad & \text{Minimizar} \quad \sum_{(i,j) \in A} c_{ij} x_{ij} \\
& \text{sujeto a} \quad \sum_{j|(i,j) \in A} x_{ij} - \sum_{k|(k,i) \in A} x_{ki} = d_i \quad \forall i \in N, \\
& \quad \quad \quad 0 \leq x_{ij} \leq D y_{ij} \quad \forall (i,j) \in A.
\end{aligned}$$

Sabemos que  $h(c)$  está bien definido porque en la red original es posible enviar un flujo factible (ya que se trata de mejorar su coste mediante una inversión extra). Por tanto

$$(P^*)' \equiv \max_{c \in C} h(c).$$

Por otro lado, por el teorema anterior, se tiene para cada  $c \in C$  que

$$\begin{aligned}
h(c) = \max \quad & \sum_{i \in N} d_i u_i + \sum_{(i,j) \in A} (D y_{ij}) v_{ij} \\
\text{s.a.} \quad & u_i - u_j + v_{ij} \leq c_{ij} \quad \forall (i,j) \in A, \\
& c \in C, \\
& u \text{ s.r.}, \quad v_{ij} \leq 0, \quad \forall (i,j) \in A.
\end{aligned}$$

Esto es,  $\max_{c \in C} h(c)$  coincide con el valor objetivo óptimo de  $(SP)$ . □

Este resultado justifica que, tras resolver el subproblema  $(SP)$  en el algoritmo de Benders, si a su valor objetivo le sumamos el gasto en costes fijos, obtenemos una cota superior del problema original.

Observamos que el problema  $(P^*)$  tiene objetivo cuadrático no convexo. Sin embargo, al aplicar dualidad fuerte al problema de optimización interior  $(P)$  el subproblema resultante sí es lineal y por tanto se puede resolver fácilmente. El método de descomposición de Benders nos permite, por tanto, resolver un problema inicialmente de objetivo cuadrático no convexo a través de la resolución iterada de problemas lineales.

El conjunto factible de este problema no depende de  $y$ , por lo que se puede razonar como en el segundo capítulo. En esta situación, aplicamos el esquema de Benders como sigue:

1. Se elige un diseño de la red  $y^0 \in Y$  (puede ser la red original cuya eficiencia se quiere mejorar).
2. Se resuelve el subproblema  $(SP)$ .

3. Sea  $(u^0, v^0, c^0)$  la solución óptima obtenida. Construir el corte de la forma (2.10):

$$\sum_{i \in N} d_i u_i^0 + \sum_{(i,j) \in A} Dv_{ij}^0 y_{ij} \leq z$$

y añadirlo al problema maestro. El valor

$$\sum_{(i,j) \in A} f_{ij} y_{ij}^0 + \sum_{i \in N} d_i u_i^0 + \sum_{(i,j) \in A} (Dy_{ij}^0) v_{ij}^0$$

es una cota superior del problema original.

4. Resolver el problema maestro, obteniendo así una cota inferior del problema original. Si las cotas superior e inferior se consideran suficientemente cercanas, se detiene el algoritmo.
5. Volver al segundo paso, resolviendo  $(SP)$  con el vector  $y$  obtenido en el problema maestro y cambiando el superíndice 0 por el correspondiente. En cada iteración, el problema maestro sería un problema lineal entero de la forma

$$\begin{aligned} & \text{Minimizar} \quad \sum_{(i,j) \in A} f_{ij} y_{ij} + z \\ & \text{sujeto a} \quad \sum_{i \in N} d_i u_i^t + \sum_{(i,j) \in A} Dv_{ij}^t y_{ij} \leq z, \quad t = 0, 1, \dots \\ & \quad \quad \quad y \in Y. \end{aligned}$$

**Teorema 4.2.3.** *El algoritmo termina en un número finito de pasos.*

**Demostración.**

Como el conjunto de puntos extremos de  $(SP)$  es finito, el número de cortes introducidos en el problema maestro es finito y cuando se han introducido todos, el problema maestro es equivalente al problema original, que por hipótesis tiene solución.  $\square$

### 4.3. Modelo con distintas tecnologías

En este modelo podemos considerar otro tipo de mejora además de la creación de nuevas conexiones: la selección de la tecnología a utilizar en cada arco. Podemos suponer que ahora disponemos de dos tecnologías, que en la red preexistente se utiliza la misma en todos los arcos y que podemos instalar cualquier tecnología en estas conexiones ya fijas. Además, supondremos también que en cada conexión solo se puede utilizar una tecnología. Consideramos los conjuntos:

- $Y_1 \subset A$ , el conjunto de los arcos de la red preexistente.
- $Y_2 \subset A$ , el conjunto de los arcos que no existen en la red a mejorar.

En estas condiciones,  $Y_1 \cup Y_2 = A$ .

Las variables de diseño en este problema serán  $y_{ijt}$ ,  $\forall (i, j) \in A$ ,  $t = 1, 2$ . A las conexiones del conjunto  $Y_1$  hay que asignarles una tecnología y las conexiones del conjunto  $Y_2$  podrán o no, estar en la nueva red. Esto puede modelar diferentes tipos de trazado ferroviario de acuerdo a la velocidad máxima a la que puede circular el tren en el ejemplo del corredor de alta velocidad del principio del capítulo.

Con esto, podemos formular este problema como:

$$\begin{aligned}
(\bar{P}^*) \quad & \text{Minimizar} \quad \sum_{(i,j) \in A} \sum_{t=1}^2 f_{ijt} y_{ijt} + \max_{c \in \bar{C}} \min \sum_{(i,j) \in A} \sum_{t=1}^2 c_{ijt} x_{ijt} \\
& \text{sujeto a} \quad \sum_{t=1}^2 \left( \sum_{j|(i,j) \in A} x_{ijt} - \sum_{k|(k,i) \in A} x_{kit} \right) = d_i \quad \forall i \in N, \\
& 0 \leq x_{ijt} \leq D y_{ijt} \quad \forall (i, j) \in A; t = 1, 2, \\
& \sum_{t=1}^2 y_{ijt} = 1 \quad \forall (i, j) \in Y_1, \\
& \sum_{t=1}^2 y_{ijt} \leq 1 \quad \forall (i, j) \in Y_2.
\end{aligned}$$

En este caso, los vectores de diseño  $\bar{y}$  pertenecen a  $\bar{Y} \subset \{0, 1\}^{2|A|}$ , donde  $\bar{Y}$  se modela por simplicidad a través de las restricciones lineales

$$\begin{aligned}
\sum_{t=1}^2 \bar{y}_{ijt} &= 1 \quad \forall (i, j) \in Y_1, \\
\sum_{t=1}^2 \bar{y}_{ijt} &\leq 1 \quad \forall (i, j) \in Y_2,
\end{aligned}$$

aunque se podrían añadir otras restricciones como se ha comentado en la sección anterior. Los costes  $c_{ijt}$  pertenecen al conjunto  $\bar{C}$ , que suponemos que se puede modelar con restricciones lineales en  $c_{ijt}$ . Así, el esquema de la descomposición de Benders correspondiente sería:

1. Se elige un rediseño de la red  $\bar{y}^0 \in \bar{Y}$ .
2. Se resuelve el subproblema  $(\overline{SP})$ :

$$(\overline{SP}) \quad \text{Maximizar} \quad \sum_{i \in N} d_i u_i + \sum_{t=1}^2 \sum_{(i,j) \in A} (Dy_{ijt}) v_{ijt}$$

$$\text{sujeto a} \quad u_i - u_j + v_{ijt} \leq c_{ijt} \quad \forall (i, j) \in A, \quad t = 1, 2;$$

$$c \in \bar{C},$$

$$u_i \text{ s.r. } \forall i \in N,$$

$$v_{ijt} \leq 0, \quad \forall (i, j) \in A, \quad t = 1, 2.$$

3. Sea  $(u^0, v^0, c^0)$  la solución óptima del dual obtenida. Construir el corte de la forma (2.10):

$$\sum_{i \in N} d_i u_i^0 + \sum_{t=1}^2 \sum_{(i,j) \in A} Dv_{ijt}^0 y_{ijt} \leq z$$

y añadirlo al problema maestro. El valor

$$\sum_{t=1}^2 \sum_{(i,j) \in A} f_{ijt} y_{ijt}^0 + \sum_{i \in N} d_i u_i^0 + \sum_{t=1}^2 \sum_{(i,j) \in A} (Dy_{ijt}^0) v_{ijt}^0$$

es una cota superior del problema original.

4. Resolver el problema maestro, obteniendo así una cota inferior del problema original. Si las cotas superior e inferior se consideran suficientemente cercanas, se detiene el algoritmo.
5. Volver al segundo paso, resolviendo  $(SP)$  con el vector  $y$  obtenido en el problema maestro y cambiando el superíndice 0 por el correspondiente.



# Capítulo 5

## Experimento computacional

Trabajaremos con el modelo presentado en el capítulo anterior: el objetivo del problema será mejorar una red preexistente a través de la creación de nuevas conexiones optimizando los costes de transporte y los de carga fija. Para ello seguiremos el esquema del algoritmo de Benders expuesto en el capítulo anterior. Lo resolveremos mediante AMPL utilizando el solver Gurobi.

### 5.1. Planteamiento del problema

Consideramos el siguiente grafo dirigido:

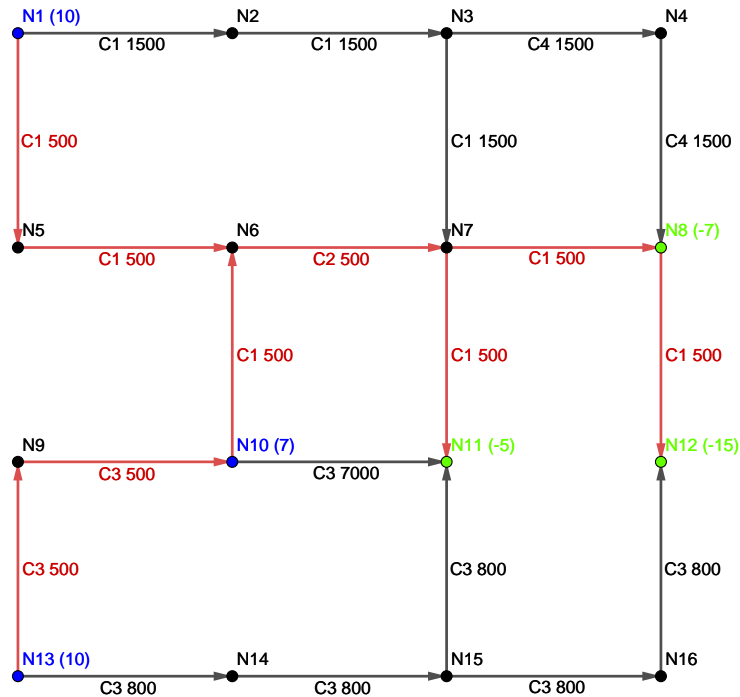


Figura 5.1: Grafo  $G(N, A, K)$  del problema

Los elementos del grafo  $G(N, A, K)$  son:

- 16 Nodos:  $N = \{N1, N2, \dots, N16\}$ .
- 20 Arcos:  $A = \{(N1, N2), (N2, N3), (N3, N4), (N1, N5), (N3, N7), (N4, N8), (N5, N6), (N6, N7), (N7, N8), (N10, N6), (N7, N11), (N8, N12), (N9, N10), (N10, N11), (N13, N9), (N15, N11), (N16, N12), (N13, N14), (N14, N15), (N15, N16)\}$ .
- Supondremos por simplicidad una única mercancía:  $K = \{1\}$ .

La red preexistente viene definida por 9 de los 20 arcos del grafo y está indicada en rojo en la representación gráfica:  $\{(N1, N5), (N5, N6), (N6, N7), (N7, N8), (N10, N6), (N7, N11), (N8, N12), (N9, N10), (N13, N9)\}$ .

Hay tres nodos de flujo positivo (que se corresponde con la salida de mercancía), representados en azul:  $N1, N10$ , y  $N13$ , con valores de flujo respectivos 10, 7, y 10. Hay tres nodos de flujo negativo (que se corresponde con la llegada de mercancía), representados en verde:  $N8, N11$ , y  $N12$ , con valores de flujo respectivos  $-7, -5$ , y  $-15$ . El flujo en el resto de nodos lo consideramos nulo. Observamos que la red preexistente es compatible con estos flujos.

Los costes fijos de los arcos se indican en la representación gráfica y para los costes de transporte variables realizamos una partición de los arcos en cuatro conjuntos:  $C_1, C_2, C_3$  y  $C_4$ , indicada también en la representación gráfica. En cada conjunto los costes varían dentro de un intervalo:

Conjunto	Intervalo del coste de transporte
C1	[600, 950]
C2	[2700, 3500]
C3	[1000, 3300]
C4	[200, 550]

El poliedro de costes de transporte  $C$  así definido es un hipercubo en  $\mathbb{R}^{27}$ . Por la construcción del modelo, el punto extremo cuyas componentes son los costes máximos de cada intervalo dominará a todo el conjunto poliédrico  $C$  y el algoritmo tomará siempre este punto para los costes de transporte del peor escenario, perdiéndose así su variabilidad. Esto no será cierto en general puesto que el modelo asume que el conjunto  $C$  puede ser un poliedro arbitrario. Para introducir esta característica de un modo sencillo en nuestro ejemplo, vamos a considerar una restricción adicional de la forma:

$$\sum_{(i,j) \in A} c_{ij} \leq M$$



con  $M \in \mathbb{R}_+$ . Podemos visualizar esta situación en  $\mathbb{R}^2$ :

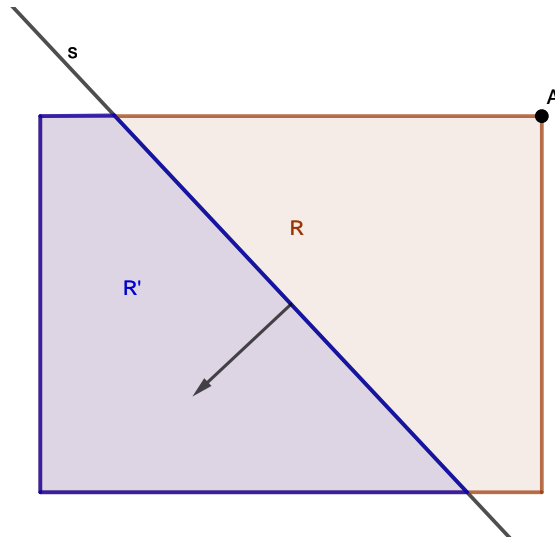


Figura 5.2

si el conjunto poliédrico de costes de transporte fuera el rectángulo  $R$  de la figura 5.2, el punto  $A$  dominaría todo el conjunto. Si introducimos el corte definido por la recta  $s$  y el sentido del vector normal que aparece en la figura, no hay ningún vector de costes que domine todo el conjunto resultante  $R'$ .

En nuestro caso introduciremos el corte

$$\sum_{(i,j) \in A} c_{ij} \leq 35000.$$

Este corte impide que el algoritmo seleccione para los costes de transporte el punto extremo cuyas componentes son los costes máximos posibles (por ejemplo,  $c_{ij} = 950$  si el arco  $(i, j)$  pertenece al grupo  $C_1$ ) puesto que, considerando  $c_{ij}$  con el máximo valor posible para cada arco  $(i, j)$ , se tiene que

$$\sum_{(i,j) \in A} c_{ij} = 39550.$$

## 5.2. Resolución computacional con AMPL

### 5.2.1. Fichero .dat

Introducimos los datos en AMPL (archivo .dat):

```
data;

set nodos:= N1 N2 N3 N4 N5 N6 N7 N8 N9 N10 N11 N12 N13 N14 N15 N16;

param: arcos: valor:=
N1 N2 0      N2 N3 0      N3 N4 0      N1 N5 1
N3 N7 0      N4 N8 0      N5 N6 1      N6 N7 1
N7 N8 1      N10 N6 1     N7 N11 1     N8 N12 1
N9 N10 1     N10 N11 0     N13 N9 1     N15 N11 0
N16 N12 0    N13 N14 0    N14 N15 0    N15 N16 0;

param demanda:=
N1 10 N2 0 N3 0 N4 0
N5 0 N6 0 N7 0 N8 -7
N9 0 N10 7 N11 -5 N12 -15
N13 10 N14 0 N15 0 N16 0;

param costesfijos:=
N1 N2 1500  N2 N3 1500  N3 N4 1500  N1 N5 500
N3 N7 1500  N4 N8 1500  N5 N6 500  N6 N7 500
N7 N8 500   N10 N6 500  N7 N11 500  N8 N12 500
N9 N10 500  N10 N11 7000 N13 N9 500  N15 N11 800
N16 N12 800 N13 N14 800 N14 N15 800 N15 N16 800;

param bloque:=
N1 N2 1      N2 N3 1      N3 N4 4      N1 N5 1
N3 N7 1      N4 N8 4      N5 N6 1      N6 N7 2
N7 N8 1      N10 N6 1     N7 N11 1     N8 N12 1
N9 N10 3     N10 N11 3     N13 N9 3     N15 N11 3
N16 N12 3    N13 N14 3    N14 N15 3    N15 N16 3;
```

A través del parámetro `valor` definimos la red preexistente: asignamos 1 a los arcos

que la forman y 0 a los que no. Mediante el parámetro `bloque` asignamos a cada arco el correspondiente conjunto de la partición hecha para los costes de transporte.

### 5.2.2. Fichero `.mod`

Modelamos el problema (archivo `.mod`):

```
set nodos;
set arcos within {nodos,nodos};

param demanda{nodos};
param bloque{arcos};
param costesfijos{arcos};
param valor{arcos};
param D:= sum{i in nodos: demanda[i] >= 0} demanda[i];
set variables ={(i,j) in arcos: valor[i,j] <= 0};
```

El parámetro `D` representa una cota superior sobre la suma de las demandas, en nuestro caso será  $D = 27$ . El conjunto de arcos que no están en la red preexistente lo definimos como `variables`.

```
param build{variables} binary;
var u{nodos};
var v{arcos} <= 0;
var costes{arcos} >= 200 <= 3500;
```

Para el subproblema definimos las variables `u{nodos}` (sin restringir), `v{arcos}` (no positiva), y `costes{arcos}` y establecemos el parámetro `build{variables}`, que tomará los valores del vector de variables de diseño `y{variables}` que proporciona el problema maestro.

```
maximize SP:
sum{i in nodos} demanda[i]*u[i] + sum{(i,j) in variables} D*build[i,j]*v[i,j]
+ sum{(i,j) in arcos: valor[i,j] = 1} D*v[i,j] ;
subj to R1 {(i,j) in arcos}:
u[i] - u[j] + v[i,j] <= costes[i,j];
subj to C1 {(i,j) in arcos: bloque[i,j] = 1}:
600 <= costes[i,j] <= 950;
subj to C2 {(i,j) in arcos: bloque[i,j] = 2}:
costes[i,j] >= 2700;
```

```

subj to C3 {(i,j) in arcos: bloque[i,j] = 3}:
1000 <= costes[i,j] <= 3300;
subj to C4 {(i,j) in arcos: bloque[i,j] = 4}:
costes[i,j] <= 550;
subj to corte:
sum{(i,j) in arcos} costes[i,j] <= 35000;

```

Introducimos el subproblema (*SP*), en particular:

$$\text{Maximizar} \quad \sum_{i \in N} d_i u_i + \sum_{(i,j) \in A} (Db_{ij}) v_{ij}$$

$$\text{sujeto a} \quad u_i - u_j + v_{ij} \leq c_{ij} \quad \forall (i,j) \in A, \quad (5.1)$$

$$600 \leq c_{ij} \leq 950, \quad \forall (i,j) \in C_1 \quad (5.2)$$

$$2700 \leq c_{ij} \leq 3500, \quad \forall (i,j) \in C_2 \quad (5.3)$$

$$1000 \leq c_{ij} \leq 3300, \quad \forall (i,j) \in C_3 \quad (5.4)$$

$$200 \leq c_{ij} \leq 550, \quad \forall (i,j) \in C_4 \quad (5.5)$$

$$\sum_{(i,j) \in A} c_{ij} \leq 35000, \quad (5.6)$$

$$u_i \text{ s.r. } \forall i \in N,$$

$$v_{ij} \leq 0, \quad \forall (i,j) \in A.$$

Donde  $c$  se corresponde con el vector de variables `costes`,  $d$  con el vector de parámetros `demanda` y  $b$  con el vector de parámetros `build`.

Las restricciones (5.1), (5.2), (5.3), (5.4), (5.5) y (5.6) se corresponden respectivamente con las restricciones definidas como `R1`, `C1`, `C2`, `C3`, `C4` y `corte` en el código.

```

param ncorte >= 0 integer;
var y{variables} binary;
var z >= 0;
param U{nodos,1..ncorte};
param V{arcos,1..ncorte} <= 0.000001;

```

Para el problema maestro definimos el parámetro `ncorte` que nos permitirá ir construyendo los cortes del problema maestro y al finalizar indicará el número de cortes construidos en total. Definimos el vector de variables de diseño `y{variables}` (solo consideramos los arcos que no están en la red preexistente porque los otros se considerarán fijos), la variable auxiliar continua `z` y los parámetros `U{nodos,1..ncorte}` y

$V\{\text{arcos}, 1..ncorte\}$  mediante los cuales construimos el corte número  $ncorte$  identificándolos con las soluciones  $u$  y  $v$  obtenidas en el subproblema.

```

minimize CosteTotal:
sum{(i,j) in variables} costesfijos[i,j]*y[i,j] + z
+ sum{(i,j) in arcos: valor[i,j] = 1} costesfijos[i,j];
subj to R2 {k in 1..ncorte}:
z >= sum{i in nodos} demanda[i]*U[i,k] + sum{(i,j) in variables}
D*V[i,j,k]*y[i,j] + sum{(i,j) in arcos: valor[i,j]=1} D*V[i,j,k];

```

Introducimos la formulación del problema maestro:

$$\begin{aligned}
& \text{Minimizar} && \sum_{(i,j) \in A} f_{ij} y_{ij} + z \\
& \text{sujeto a} && \sum_{i \in N} d_i u_i^k + \sum_{(i,j) \in A} D v_{ij}^k y_{ij} \leq z, \quad k = 0, 1, \dots \\
& && y \in Y.
\end{aligned}$$

Donde  $f$  se corresponde con el vector de parámetros `costesfijos`, e  $Y \subset \{0, 1\}^{11}$  es el conjunto de posibles rediseños de la red fijando la ya existente. Obsérvese que, incluso para un ejemplo tan sencillo como el que se representa en la figura 5.1, tenemos más de dos mil posibles diseños de la red mejorada. Sin embargo, si consideramos este problema incluyendo la posibilidad de cambiar la tecnología usada en los arcos de la red existente, el número de diseños potenciales ascendería a más de un millón.

La restricción  $y \in Y$  la introducimos en el código imponiendo que  $y_{ij} = 1$  en aquellos arcos  $(i, j)$  que pertenezcan a la red (en el código esto se corresponde con `valor[i,j]=1`). El resto de restricciones se corresponden en el código con el conjunto de las `ncorte` restricciones definidas en `R2`.

### 5.2.3. Fichero .run

Creamos el fichero ejecutable (archivo `.run`):

```

model experimento.mod;
data experimento.dat;

option omit_zero_rows 0;
check: sum {i in nodos} demanda[i] = 0;

```

La opción `omit_zero_rows 0` permite que al mostrar los vectores de soluciones no se omitan las componentes que valen 0. Comprobamos que la suma de todas las demandas sea 0.

```
problem Maestro: y, z, CosteTotal, R2;
    option display_1col 20;
    option solver gurobi;

problem Sub: u, v, costes, SP, R1, C1, C2, C3, C4, corte;
    option solver gurobi;
```

La clave de la formulación del esquema de descomposición es que en AMPL podemos definir objetos de tipo problema a través de `problem`. Definimos el problema maestro y el subproblema indicando sus variables, su función objetivo y sus restricciones. La opción `display_1col 20` permite que al mostrar las soluciones el número máximo de elementos por columna sea 20, que es el número de arcos con el que trabajamos. Utilizamos el solver Gurobi para ambos problemas.

```
printf "\n";
display D;
let ncorte := 0;
let z := 0;
let {(i,j) in variables} build[i,j] := 1;
```

Por cuestión de estética, para que la salida aparezca separada del texto que haya previamente a la ejecución en la ventana de consola, pedimos que al mostrar la salida comience con un salto de línea. También pedimos que nos muestre el valor del parámetro `D`, que en nuestro caso debe ser 27. Antes de comenzar con las iteraciones inicializamos el número de cortes y la variable auxiliar `z` a 0. Además, en el primer subproblema comenzamos suponiendo construidas todas las conexiones.

```
param sup default Infinity;
param inf default 0;
param costeredprevia := sum{(i,j) in arcos: valor[i,j] = 1} costesfijos[i,j];
param objoriginal;
display costeredprevia;
```

Definimos los parámetros cuyos valores serán las cotas superiores e inferiores: `sup` (con valor infinito por defecto) e `inf` (con valor 0 por defecto, ya que todos los costes

considerados son positivos) respectivamente. Con el objetivo de reducir el número de operaciones, definimos los parámetros:

- **costeredprevia**: su valor es el gasto en costes fijos de la red original. Pedimos que nos lo muestre.
- **objoriginal**: su valor en cada iteración será el valor objetivo del problema original. Es necesario para determinar la cota superior.

```
repeat{printf "\nITERATION %d\n\n",ncorte+1;
  printf "\nSUBPROBLEMA: \n";
  solve Sub;
  let objoriginal := SP + sum{(i,j) in variables}
    y[i,j]*costesfijos[i,j] + costeredprevia;
  if objoriginal < sup && ncorte > 0 then{
    let sup := objoriginal;};
  display objoriginal;
  display sup, inf;
  display R1.dual,costes, corte.body;
  if sup > inf + 0.0001 then{
    let ncorte := ncorte +1;
    let {i in nodos} U[i,ncorte] := u[i];
    let {(i,j) in arcos} V[i,j,ncorte] := v[i,j];
  }
  else break;
  printf "\nPROBLEMA MAESTRO: \n";
  solve Maestro;
  if CosteTotal > inf then{
    let inf := CosteTotal;};
  expand;
  printf "\n";
  display y;
  let {(i,j) in variables} build[i,j] := y[i,j];
};
```

Este es el bucle principal. Seguimos el esquema del algoritmo de Benders descrito en el capítulo anterior: en cada iteración resolvemos el subproblema (`solve Sub;`) cuya solución utilizamos para construir los cortes del problema maestro, que nos proporciona una cota inferior del problema original. A través del comando `display` (en la primera vez que lo utilizamos en el bucle) pedimos que nos muestre en pantalla:

- `objoriginal`: el valor objetivo del problema original en esa iteración.
- `sup` e `inf`: cota superior y cota inferior respectivamente.
- `R1.dual`: valor de las variables duales asociadas a las restricciones `R1`, que representan el flujo en cada arco.
- `costes`: vector de costes obtenido en el subproblema.
- `corte.body`: valor que toma la suma de la restricción `corte` al resolver el subproblema, es decir, el valor de  $\sum_{(i,j) \in A} c_{ij}$  con los  $c_{ij}$  obtenidos en el subproblema.

Para el test de salida consideramos un *gap* de  $10^{-5}$ , cuando las cotas superior e inferior se encuentren suficientemente próximas se detiene el algoritmo. Esto lo indicamos en las líneas:

```
if sup > inf + 0.00001 then{...}
else break;
```

Si tras resolver el subproblema el bucle no se detiene, introducimos el corte correspondiente para el problema maestro haciendo uso de las variables obtenidas en el subproblema:

```
let ncorte := ncorte +1;
let {i in nodos} U[i,ncorte] := u[i];
let {(i,j) in arcos} V[i,j,ncorte] := v[i,j];
```

En cada iteración introducimos un corte más al problema maestro, por lo que aumentamos en una unidad el parámetro `ncorte` y en las siguientes dos líneas proporcionamos los coeficientes que definen dicho corte, que se corresponden con las variables obtenidas en el subproblema. Después de esto, resolvemos el problema maestro (`solve Maestro`) y utilizamos el comando `expand` para mostrar en pantalla la formulación de este problema, lo que nos permite ver los cortes que se van construyendo (en las últimas iteraciones la formulación será muy extensa puesto que los cortes se van acumulando). Finalmente, actualizamos los valores del vector de parámetros `build` con los valores obtenidos en el vector de diseño y del problema maestro, que además mostramos en pantalla (`display y`).

Recordamos que la cota superior la proporciona el valor objetivo original del problema:

```
let objoriginal := SP + sum{(i,j) in variables}
  y[i,j]*costesfijos[i,j] + costeredprevia;
if objoriginal < sup && ncorte > 0 then{
  let sup := objoriginal;};
```



y la cota inferior la proporciona el valor objetivo del problema maestro:

```
if CosteTotal > inf then{
  let inf := CosteTotal;};
```

Finalmente,

```
param solucion{arcos};
let {(i,j) in variables} solucion[i,j] := y[i,j];
let {(i,j) in arcos: valor[i,j] = 1} solucion[i,j] := valor[i,j];
display solucion;
```

Mostramos `solucion`, que es el vector de las variables de diseño (visualizando también las componentes fijas de los arcos que ya estaban en la red).

#### 5.2.4. Salida

Al ejecutar el fichero `.run`, obtenemos que el algoritmo se detiene tras 66 iteraciones (se construyen 65 cortes), de las cuales mostramos las dos primeras y las dos últimas (en la penúltima no expandiremos el problema maestro debido a su extensión):

D = 27

costeredprevia = 4500

ITERATION 1

SUBPROBLEMA:

Gurobi 10.0.0: Gurobi 10.0.0: optimal solution; objective 184300

4 simplex iterations

objoriginal = 188800

sup = Infinity

inf = 0

```
:      R1.dual costes      :=
N1  N2      10      950
N1  N5       0      600
```

```

N10 N11    2.5    1850
N10 N6     9.5     950
N13 N14     5     3300
N13 N9      5     3300
N14 N15     5     3300
N15 N11    2.5    1850
N15 N16    2.5    3300
N16 N12    2.5    3050
N2  N3     10     950
N3  N4     10     550
N3  N7      0     600
N4  N8     10     550
N5  N6      0     600
N6  N7     9.5    3500
N7  N11     0     600
N7  N8     9.5     950
N8  N12    12.5     950
N9  N10     5     3300
;

```

```

corte.body = 35000

```

```

PROBLEMA MAESTRO:

```

```

Gurobi 10.0.0:                Gurobi 10.0.0: optimal solution; objective 188800
0 simplex iterations

```

```

minimize CosteTotal:

```

```

1500*y['N1','N2'] + 1500*y['N2','N3'] + 1500*y['N3','N4'] +
1500*y['N3','N7'] + 1500*y['N4','N8'] + 7000*y['N10','N11'] +
800*y['N15','N11'] + 800*y['N16','N12'] + 800*y['N13','N14'] +
800*y['N14','N15'] + 800*y['N15','N16'] + z + 4500;

```

```

subject to R2[1]:

```

```

z >= 184300;

```

```

y :=

```

```

N1  N2    0
N10 N11   0

```

```

N13 N14 0
N14 N15 0
N15 N11 0
N15 N16 0
N16 N12 0
N2 N3 0
N3 N4 0
N3 N7 0
N4 N8 0
;

```

ITERATION 2

SUBPROBLEMA:

```

Gurobi 10.0.0:          Gurobi 10.0.0: optimal solution; objective 235550
9 simplex iterations
objoriginal = 240050

```

```

sup = 240050
inf = 188800

```

```

:          R1.dual costes      :=
N1 N2      0      950
N1 N5      10      950
N10 N11     0     3300
N10 N6      17      950
N13 N14     0     1000
N13 N9      10     3300
N14 N15     0     1000
N15 N11     0     3300
N15 N16     0     3300
N16 N12     0     3300
N2 N3       0      950
N3 N4       0      550
N3 N7       0      950
N4 N8       0      550
N5 N6      10      950

```

```

N6 N7      27      3500
N7 N11      5       950
N7 N8      22      950
N8 N12     15      950
N9 N10     10     3300
;

```

```

corte.body = 34950

```

```

PROBLEMA MAESTRO:

```

```

Gurobi 10.0.0:                Gurobi 10.0.0: optimal solution; objective 189600

```

```

0 simplex iterations

```

```

minimize CosteTotal:

```

```

1500*y['N1','N2'] + 1500*y['N2','N3'] + 1500*y['N3','N4'] +
1500*y['N3','N7'] + 1500*y['N4','N8'] + 7000*y['N10','N11'] +
800*y['N15','N11'] + 800*y['N16','N12'] + 800*y['N13','N14'] +
800*y['N14','N15'] + 800*y['N15','N16'] + z + 4500;

```

```

subject to R2[1]:

```

```

z >= 184300;

```

```

subject to R2[2]:

```

```

68850*y['N3','N7'] + 90450*y['N4','N8'] + 56700*y['N10','N11'] +
180900*y['N15','N11'] + 117450*y['N15','N16'] + z >= 235550;

```

```

y :=

```

```

N1 N2      0
N10 N11     0
N13 N14     0
N14 N15     0
N15 N11     1
N15 N16     0
N16 N12     0
N2 N3       0
N3 N4       0
N3 N7       0
N4 N8       0

```

;

ITERATION 65

SUBPROBLEMA:

Gurobi 10.0.0:

Gurobi 10.0.0: optimal solution; objective 193175

3 simplex iterations

objoriginal = 213175

sup = 204450

inf = 204300

: R1.dual costes :=

N1	N2	10	950
N1	N5	0	600
N10	N11	2.5	2025
N10	N6	9.5	950
N13	N14	5	3300
N13	N9	5	3300
N14	N15	5	3300
N15	N11	2.5	2025
N15	N16	2.5	3050
N16	N12	2.5	3300
N2	N3	10	950
N3	N4	0	200
N3	N7	10	950
N4	N8	0	200
N5	N6	0	600
N6	N7	9.5	3500
N7	N11	0	600
N7	N8	19.5	950
N8	N12	12.5	950
N9	N10	5	3300

;

cor.te.body = 35000

PROBLEMA MAESTRO:

Gurobi 10.0.0:  
60 simplex iterations  
1 branching nodes

Gurobi 10.0.0: optimal solution; objective 204450

y :=

N1 N2 1  
N10 N11 0  
N13 N14 1  
N14 N15 1  
N15 N11 1  
N15 N16 0  
N16 N12 0  
N2 N3 1  
N3 N4 1  
N3 N7 0  
N4 N8 1  
;

ITERATION 66

SUBPROBLEMA:

Gurobi 10.0.0:  
11 simplex iterations  
objoriginal = 204450

Gurobi 10.0.0: optimal solution; objective 191550

sup = 204450

inf = 204450

: R1.dual costes :=  
N1 N2 10 950  
N1 N5 0 600  
N10 N11 0 1000  
N10 N6 12 950  
N13 N14 5 3300  
N13 N9 5 3300

N14	N15	5	3300
N15	N11	5	3300
N15	N16	0	3300
N16	N12	0	2450
N2	N3	10	950
N3	N4	10	550
N3	N7	0	600
N4	N8	10	550
N5	N6	0	600
N6	N7	12	3500
N7	N11	0	600
N7	N8	12	950
N8	N12	15	950
N9	N10	5	3300

;

cor.te.body = 35000

solucion :=

N1	N2	1
N1	N5	1
N10	N11	0
N10	N6	1
N13	N14	1
N13	N9	1
N14	N15	1
N15	N11	1
N15	N16	0
N16	N12	0
N2	N3	1
N3	N4	1
N3	N7	0
N4	N8	1
N5	N6	1
N6	N7	1
N7	N11	1
N7	N8	1
N8	N12	1
N9	N10	1

;

R1.dual :=

N1	N2	10
N1	N5	0
N10	N11	0
N10	N6	12
N13	N14	5
N13	N9	5
N14	N15	5
N15	N11	5
N15	N16	0
N16	N12	0
N2	N3	10
N3	N4	10
N3	N7	0
N4	N8	10
N5	N6	0
N6	N7	12
N7	N11	0
N7	N8	12
N8	N12	15
N9	N10	5

;



En la figura siguiente podemos visualizar el rediseño de la red obtenido, las nuevas conexiones construidas las representamos en morado.

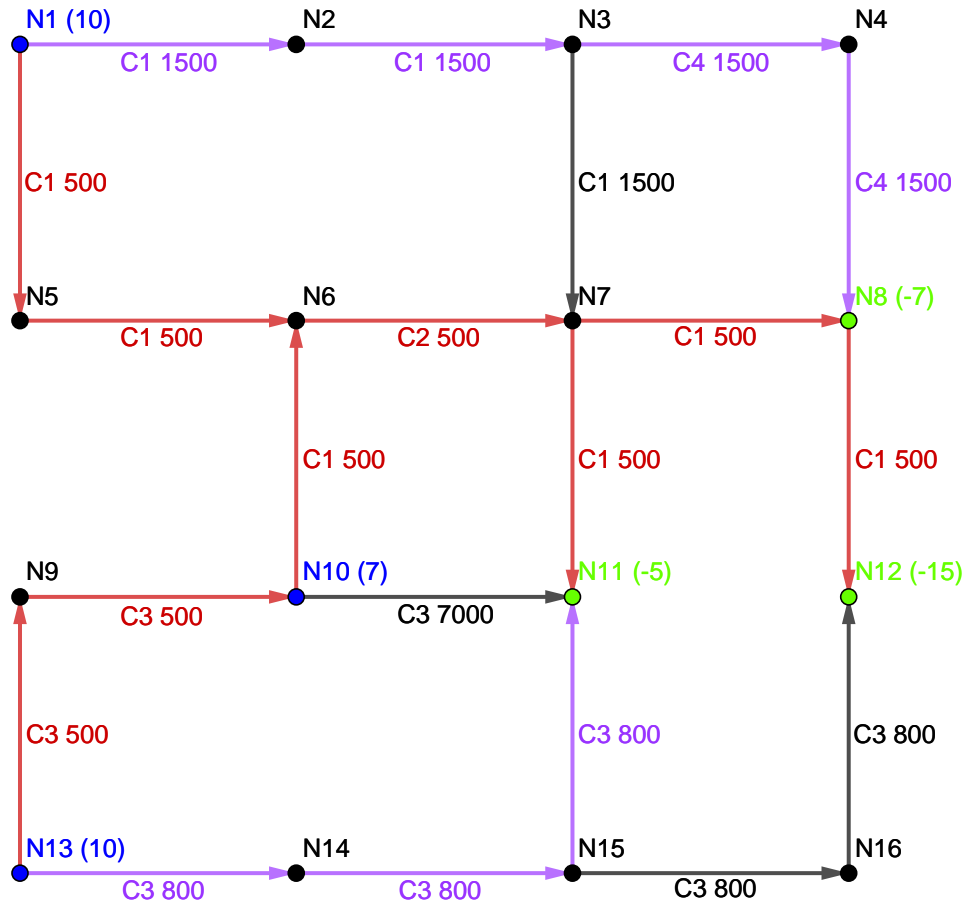


Figura 5.3: Red mejorada

Teniendo en cuenta los flujos óptimos ( $R1.dual$ ) vemos que las conexiones de la red preexistente (N1,N5) y (N5,N6) quedan sin utilizar, a pesar de que los costes de transporte son los mismos que los de las conexiones que las sustituyen. Esto es debido a que la conexión (N6,N7), por la que debería circular la mercancía si se utilizasen los dos arcos anteriores tiene costes de transporte muy elevados. Observamos también que el elevado coste de esa conexión fuerza a que la conexión (N7,N11) también quede sin utilizar.

Recordemos que, dado un diseño de la red, el subproblema (SP) (a través de su versión dual) determina el flujo óptimo y proporciona el gasto en costes de transporte en el peor de los casos para este flujo. Después, el problema maestro genera otro diseño de red tratando de minimizar los costes fijos y teniendo en cuenta la información proporcionada por el subproblema a través de cortes. En nuestro experimento, la red

inicial que proporcionamos al subproblema es aquella en la que están disponibles todas las conexiones posibles (todas las componentes del vector de diseño  $y$  valen 1), por lo que el valor objetivo de este primer subproblema va a ser una cota inferior para el del resto de subproblemas, en los que puede ser que no estén disponibles todas las conexiones posibles. Además, el valor de `objoriginal` en la primera iteración no se puede utilizar para obtener una cota superior del problema original pues todavía el vector de variables  $y$  es nulo y no se tendrían en cuenta los costes fijos correspondientes.

Por otro lado, es natural que el primer problema maestro proporcione la red original, porque la única restricción que tiene es una cota inferior sobre  $z$  y como el objetivo es minimizar  $\sum_{(i,j) \in A} f_{ij}y_{ij} + z$ , el valor óptimo se alcanza tomando  $y$  con el mayor número posible de componentes nulas, lo que se corresponde con la red preexistente.

Como consecuencia de la proposición 4.2.4, el valor objetivo óptimo del último problema maestro y el del último subproblema resuelto deben diferenciarse en el gasto en costes fijos de la red final. Mediante una simple comprobación vemos que esto se verifica: el valor objetivo óptimo del problema maestro es 204450 y el del último subproblema es 191550,

$$\text{sum}\{(i,j) \text{ in arcos}\} \text{ costesfijos}[i,j]*\text{solucion}[i,j] = 12900$$

$$204450 - 191550 = 12900.$$

Si suponemos que en la red preexistente el gasto en costes de transporte está sujeto a las restricciones  $\sum_{(i,j) \in A} c_{ij} \leq 35000$ ,  $c \in C$  y evaluado en el peor de los casos, podemos ver cuánto ha mejorado la red: el valor objetivo del problema original en la segunda iteración es el coste total de la red preexistente, 240.050, y el valor objetivo del problema original en la última iteración es el coste total de la red mejorada, 204.450. Por tanto,

- Diferencia:  $240,050 - 204,450 = 35,600$ .

En la siguiente figura podemos visualizar la evolución de las cotas:

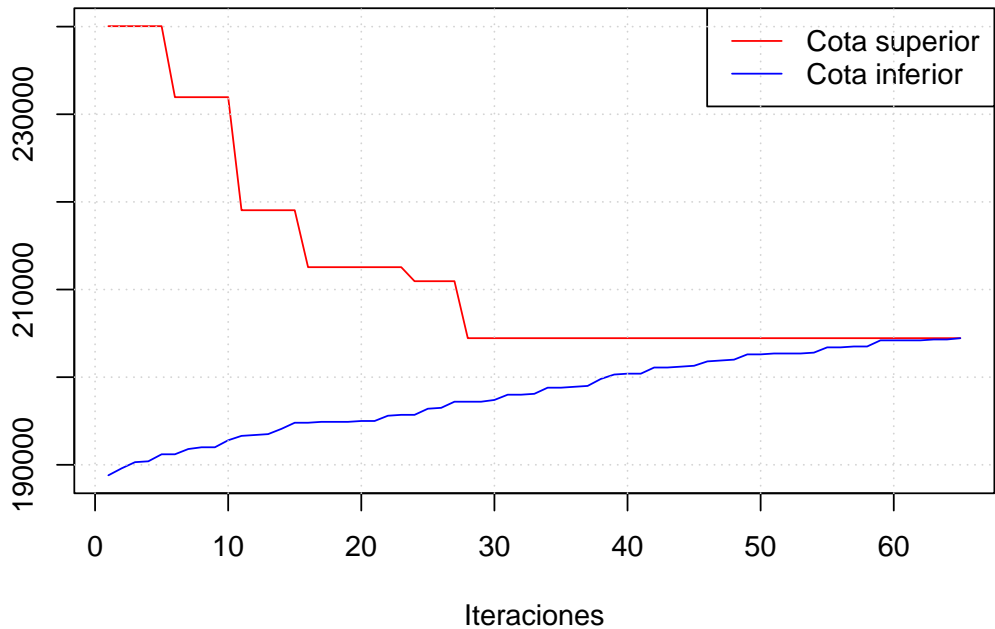


Figura 5.4: Evolución de las cotas.

**Observación 5.2.1.** Dado que partimos de una red preexistente, el subproblema tiene siempre solución óptima finita y en cada iteración se genera un corte para el problema maestro. Así, en la segunda iteración vemos que hay dos cortes y en la penúltima iteración habrá 65 cortes. En la última no se genera ningún corte porque el algoritmo termina al ser suficientemente próximas las cotas superior e inferior.

**Observación 5.2.2.** Observamos que hay iteraciones en las que algunas componentes de  $R1.dual$  (que se corresponden con las variables de flujo  $x_{ij}$ ) no son enteras. Esto se debe a que hay solución múltiple y AMPL no muestra por defecto la solución entera, pero sabemos que esta existe según se estableció en el capítulo anterior.



## Capítulo 6

# Conclusiones

Como ya se mencionó anteriormente, el objetivo de este trabajo ha sido introducir y describir el método de descomposición de Benders y su aplicación a los problemas de diseño de redes con carga fija.

Hemos visto que la teoría matemática sobre la que se fundamenta este método es relativamente sencilla, solo requiere de conocimientos básicos sobre Programación Matemática, en especial, de la teoría de dualidad lineal. Lo revolucionario de este método es la idea sobre la que se basa: aprovechar la estructura del problema, en nuestro caso el de diseño de redes con carga fija, para obtener dos problemas mucho más sencillos de resolver que el original, y que a través de la implementación del correspondiente algoritmo proporcionan la solución de este. Por otro lado, aunque la implementación básica del esquema se base en un método de corte, si se desea mejorar la eficiencia de estos cortes sí es necesario recurrir a construcciones más complejas como las que garantizan cortes Pareto-óptimos.

También hemos mostrado la potente capacidad de cálculo de este método: en el capítulo 4 hemos visto cómo la descomposición de Benders nos permite resolver un problema inicialmente de objetivo cuadrático no convexo mediante la resolución de varios problemas de carácter lineal. Los problemas de objetivo cuadrático no convexo se sabe que son bastante difíciles, en términos teóricos, se caracterizan como NP-difíciles. En el experimento computacional hemos mostrado que al resolver un problema de este carácter a través de AMPL, el algoritmo correspondiente a este método de descomposición ha proporcionado una solución en 66 iteraciones.

Cabe destacar también la gran diversidad de situaciones que se pueden modelar a través de las formulaciones de los problemas de diseño de redes con carga fija, tanto a nivel teórico como se ha visto en el capítulo 3, como a nivel práctico, como el problema actual de la red ferroviaria de alta velocidad entre España y Francia expuesto en el

capítulo 4.

En definitiva, la potencia de cálculo de este método, junto con la flexibilidad que demuestra en la aplicación a diferentes modelos de optimización, han hecho que este método de Programación Matemática haya resurgido en las últimas décadas al aumentar los problemas de planificación de redes.

# Referencias

- [1] Geoffrion AM. Generalized Benders decomposition. *Journal of Optimization Theory and Applications*, 10:60-237, 1972.
- [2] Magnanti T, Wong RT. Accelerating Benders decomposition: algorithmic enhancement and model selection criteria. *Operations Research*, 23:84-464, 1981.
- [3] Magnanti T, Mireault P, Wong RT. Tailoring Benders decomposition for uncapacitated network design. *Mathematical Programming Study*, 26:54-112, 1986.
- [4] Boyce DE, Farhi A, Weischedel R. Optimal network problem: a branch-and-bound algorithm. *Environment and Planning*, 5:33-519, 1973.
- [5] Gutierrez GJ, Kouvelis P, Kurawarwala AA. A robustness approach to uncapacitated network design problems. *European Journal of Operations Research*, 94:76-362, 1996.
- [6] Randazzo CD, Luna HPL. A comparison of optimal methods for local access uncapacitated network design. *Annals of Operations Research*, 106:86-263, 2001.
- [7] Gavish B. Topological design of centralized computer networks—formulations and algorithms. *Networks*, 12:77-355, 1982.
- [8] Randazzo CD, Luna HPL, Mahey P. Benders decomposition for local access network design with two technologies. *Discrete Mathematics and Theoretical Computer Science*, 4:46-235, 2001.
- [9] Sridhar V, Park JS. Benders-and-cut algorithm for fixed-charge capacitated network design problem. *European Journal of Operations Research*, 125:32-622, 2000.
- [10] Magnanti T, Mirchandani P, Vachani R. Modeling and solving the two-facility capacitated network loading problem. *Operations Research*, 43:57-142, 1995.
- [11] Bazaraa MS, Jarvis JJ, Sherali HD. *Linear Programming and Network Flows*. Wiley, 2010.
- [12] Gubern A. (21 de marzo de 2023). *Francia desconecta a España de la alta velocidad ferroviaria en Europa*. ABC. <https://amp.elperiodico.com/es/economia/20230409/francia-conexion-ave-espana-85797543>

- [13] Ahuja RK, Magnanti TL, Orlin JB. Network Flows: Theory, Algorithms, and Applications. Prentice Hall, 1993.
- [14] Wolsey LA. Integer Programming. Wiley, 1998.
- [15] Balakrishnan N, Wong RT. A network model for the rotating workforce scheduling problem. *Networks*, 20:25-42, 1990.
- [16] Cordeau JF, Stojkovic G, Soumis F, Desrosiers J. Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Science*, 35:375-388, 2001.
- [17] Fontaine P, Minner S. Benders decomposition for the hazmat transport network design problem. *European Journal of Operational Research*, 267(1):996-1002, 2018.
- [18] Zarrinpoor N, Fallahnezhad MS, Pishvae MS. The design of a reliable and robust hierarchical health service network using an accelerated Benders decomposition algorithm. *European Journal of Operational Research*, 265(3):1013-1032, 2018.
- [19] Benders JF. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238-262, 1962.