BFC: UNA EFICIENTE METODOLOGÍA EN LA SOLUCIÓN DE MODELOS DE OPTIMIZACIÓN

Tania Fuertes Martín

Trabajo de investigación 012/015

Master en Banca y Finanzas Cuantitativas

Directora: Mª Araceli Garín

Universidad del País Vasco

Universidad de Castilla-La Mancha

Universidad Complutense de Madrid

Universidad del País Vasco

Universidad de Valencia

www.finanzascuantitativas.es

Índice general

1. Introducción **5**

2. Problemas de optimización estocástica mixtos 0-1 de dos etapas **8**

2.1. Introducción 8

2.2. Modelización de la incertidumbre 9

2.3. Representaciones del DEM 13

2.4. Técnicas de resolución 15

2.4.1. Técnicas para la representación compacta 15

2.4.2. Técnicas para la representación extendida 18

2.4.3. Software de optimización 19

2.5. Aplicaciones 22

2.5.1. Modelos de planificación estocástica multiperiodo 24

2.5.2. Modelo de gestión de activos y pasivos 25

2.5.3. Estructuración de carteras de títulos con garantía hipotecaria 26

3. Algoritmo BFC (Branch-and-Fix Coordination) **28**

3.1. Definiciones previas 29

3.2. Bases teóricas 34

3.3. Algoritmo 38

4. Experiencia computacional **40**

4.1. Casos de estudio 40

4.2. Caso ilustrativo 41

4.3. Resultados numéricos 43

5. Conclusiones **50**

6. Comentarios y trabajo futuro **52**

A. Programación **56**

Índice de tablas

4.1. Dimensión de los modelos 41

4.2. Resultados para la representación compacta (COIN-OR y CPLEX) . 44

4.3. Resultados BFC (con cota superior) bajo COIN-OR y CPLEX . . . 46

4.4. Resultados BFC sin y con cota superior bajo CPLEX 47

4.5. Resultados BFC sin y con cota superior bajo CPLEX para el caso P5 48

4.6. Resultados BFC bajo CPLEX para problemas de altas dimensiones 49

Índice de figuras

2.1. Árbol de escenarios 11

2.2. Partición en racimos de escenarios 12

3.1. Variable común 30

3.2. Nodos gemelos 31

3.3. Nodos no gemelos 31

3.4. Familia de nodos gemelos 32

4.1. Ejemplo ilustrativo 42

Capítulo 1

Introducción

Desde los años 70 el dominio de las operaciones financieras ha supuesto una gran transformación. La liberación de los mercados financieros y la inflación, y la crisis del petróleo al mismo tiempo, conducen a aumentar la volatilidad de los tipos de interés.

La incertidumbre crea creatividad y como resultado, hemos visto un uso incrementado de técnicas analíticas avanzadas en la forma de modelos de optimización para aspectos muy diversos de las operaciones financieras. Los modelos para la estimación de la estructura temporal de los tipos de interés (ETTI), la celebrada fórmula de Black-Scholes para la valoración de opciones, y otros instrumentos complejos, se añaden a la larga lista de contribuciones desde el trabajo inicial de Markowitz en el análisis de media-varianza para rendimientos de stocks en 1952.

A lo largo del mismo periodo, las herramientas para la investigación sobre la gestión de operaciones alcanzan una etapa de sofisticación que atrae la atención de investigadores que desarrollan su actividad en este entorno dinámico. Los analistas de investigación de operaciones encuentran un campo de problemas muy apasionantes donde sus herramientas pueden tener un gran impacto. Los avances en la tecnología de computación facilitan el desarrollo y la validación de modelos. Como resultado, los modelos resultan ser herramientas indispensables en muchos campos de operaciones financieras.

Por su parte, los algoritmos matemáticos son necesarios para obtener soluciones a estos modelos matemáticos. Afortunadamente, hay algoritmos que tienen amplias aplicaciones y dan soluciones para toda clase de modelos matemáticos. Sin embargo, hay ejemplos de modelos a gran escala para los que los algoritmos de solución estándar no son suficientes, y hay que desarrollar algoritmos especiales.

Un modelo matemático es un modelo abstracto. Es la descripción de alguna parte del mundo real expresado en leguaje matemático. Un tipo de modelo matemático

se denomina *modelo de optimización*. Estos modelos son un instrumento valioso para dar solución a muchos tipos de problemas de la vida real, como son los de planificación, problemas de contratación, selección de carteras, etc.

En general, la combinación de los modelos y los ordenadores mejoran la rapidez en la obtención de la solución de los distintos problemas. Los ordenadores te dan la facilidad de manipular rápidamente más información de la que podrías manejar manualmente.

Dentro de los modelos de optimización podemos encontrar desde el más sencillo, el problema de optimización determinista lineal, que sólo contempla variables continuas en un momento del tiempo y sus parámetros son perfectamente conocidos; hasta uno de los más elaborados, como es el problema estocástico entero-mixto multietapa, que incluye varios momentos de decisión y en cada uno de ellos puede presentar variables tanto enteras como continuas, además de incertidumbre en todos o algunos de los parámetros.

Un problema de optimización puede involucrar parámetros cuyo valor no es conocido exactamente en el momento de decidir. Cuando existe incertidumbre sobre los datos del problema se habla de optimización estocástica, a diferencia de la optimización determinista en la que todos los parámetros son conocidos en el momento de tomar la decisión. Suele considerarse que el proceso de toma de decisiones está dividido en etapas, entre las cuales tiene lugar la realización de una parte de la incertidumbre. Habitualmente estas etapas de decisión hacen referencia a periodos temporales, aunque no tiene que ser necesariamente así.

Por tanto, debido a la gran variedad de modelos que podemos crear, la optimización es una herramienta de ayuda en la toma de decisiones que permite escoger la mejor estrategia para alcanzar un objetivo. Para esto es necesario modelar como problema de optimización el entorno en el que se produce esa toma de decisión. De ahí que vayan de la mano la modelización del problema y el desarrollo de algoritmos de resolución.

En este caso nos centraremos en un caso particular de los problemas enteros-mixtos multietapa: el problema estocástico mixto 0-1 de dos etapas. Se trata de un problema de decisión bajo incertidumbre en dos fases en cualquiera de la cuales pueden aparecen variables continuas y variables 0-1.

El objetivo concreto de este trabajo es recoger la metodología de *Branch & Fix Coordination* (BFC) para resolver este tipo de problemas y estudiar su eficiencia cuando nos enfrentamos a problemas estocásticos mixtos 0-1 dos etapas de altas dimensiones.

Al abordar esta técnica de resolución, así como cualquiera de las que se han desarrollado en la literatura, se hace necesario el uso de los ordenadores. Por esta razón se han producido grandes avances en el desarrollo de técnicas de resolución en las últimas décadas. Puede considerarse un punto de partida de este desarrollo

la aparición de las OSL (*Optimization Subroutine Library*). Se trata de un conjunto de subrutinas que se incorporan al código de programas que resuelven problemas de optimización más complicados.

A finales de 1990 IBM presenta una versión escrita en FORTRAN, y a principios de 2000 se traslada al lenguaje de programación C integrádolo en el proyecto *COmputational INfrastructure for Operations Research* (COIN-OR), ver [16].

Al mismo tiempo, durante los años 80 Robert Bixby desarrolla el optimizador CPLEX. El optimizador IBM ILOG CPLEX resuelve problemas de optimización entera, problemas de optimización lineal de grandes dimensiones y aún continúa desarrollándose bajo IBM. Este optimizador también se puede utilizar bajo el entorno de COIN-OR, ver [15].

Compararemos la eficiencia del algoritmo propuesto bajo los dos optimizadores: COIN-OR y CPLEX.

El trabajo está organizado de la siguiente manera. En el Capítulo 2 describimos los modelos estocásticos y cómo se modeliza la incertidumbre, así como las distintas representaciones que admiten estos modelos. Además, recogemos las diversas técnicas de resolución y sus aplicaciones más importantes. En el siguiente capítulo, detallamos el esquema de descomposición BFC. En el Capítulo 4 presentamos los resultados más relevantes de la aplicación de este algoritmo a distintos casos de estudio. En primer lugar, detallamos paso a paso el algoritmo con un caso ilustrativo, y a continuación destacamos los resultados computacionales más importantes para casos de grandes dimensiones. Finalmente, en el último capítulo recogemos las principales conclusiones del trabajo y futuras líneas de investigación.

Capítulo 2

Problemas de optimización estocástica mixtos 0-1 de dos etapas

2.1. Introducción

Un problema estocástico de dos etapas es un problema de decisión bajo incertidumbre en dos fases en el que en la segunda etapa se recoge dicha incertidumbre considerando distintos escenarios posibles. En una primera etapa se decide el valor de un conjunto de variables a partir de parámetros conocidos, y en la segunda se eligen los valores para el resto de variables en función del escenario que haya ocurrido, dentro de los posibles.

El carácter estocástico del problema puede aparecer en cualquier parte del modelo. Sin embargo, la incertidumbre se modeliza mediante un conjunto finito de escenarios $\omega = 1, ..., |\Omega|$, que cada uno de ellos lleva asociada una probabilidad de ocurrencia w^ω, $\omega \in \Omega$.

Las decisiones de primera etapa no dependen del escenario que ocurra, por lo que las variables han de tomar el mismo valor en cada uno de ellos. Más adelante nos referiremos a esta condición como *restricción de no anticipatividad*, que asegura que se toman las decisiones de primera etapa sin conocer con exactitud la realización de la incertidumbre que se va a producir en la siguiente etapa, es decir, que fuerza a que se tome una única decisión para todos los escenarios. Por otro lado, los valores de las variables de la primera etapa influyen en los valores de las variables de segunda etapa tanto como el escenario en el que nos encontremos; y así estas variables de segunda etapa no tienen por qué tener el mismo valor en cada uno de los escenarios.

El modelo más simple es aquél que sólo contiene variables 0-1 en la primera

etapa y variables continuas en la segunda etapa, ver en [4] la primera descripción de la metodología de Ramificación y Fijación Coordinada (BFC, de sus siglas en inglés, Branch & Fix Coordination). También se han abordado problemas en los que en la primera etapa las variables son puramente 0-1 o continuas, y en la segunda etapa hay variables mixtas. Para el caso en el que las variables de primera etapa son únicamente binarias se han desarrollado algoritmos específicos, por ejemplo [27].

De igual manera se han propuesto esquemas algorítmicos de resolución para problemas estocásticos en los que las variables de ambas etapas son mixtas, pero se han dado resultados para casos en los que las dimensiones eran reducidas. Además, aparecen en la literatura problemas que consideran variables enteras y continuas en ambas etapas, como en [14] donde se proponen algoritmos de Ramificación y Acotación (Branch & Bound).

Todo problema de optimización estocástica admite un modelo determinista equivalente DEM (del inglés, Deterministic Equivalent Model), ver [32]. Dependiendo de la forma en que se formulan las condiciones de no anticipatividad, ver [33], aparecen distintas representaciones del DEM, compacta, extendida o en variables divididas, o en variables divididas por clusters o racimos de escenarios.

En este capítulo presentamos el modelo de optimización estocástica mixto 0-1 partiendo de su versión determinista e introduciendo la incertidumbre por medio de un conjunto de escenarios. Una vez hecho este planteamiento presentamos las diferentes representaciones del DEM. Por último, consideramos las distintas técnicas de resolución adecuadas para cada formulación del DEM y señalamos algunas aplicaciones de los modelos de optimización estocástica.

2.2. Modelización de la incertidumbre

Un problema de optimización determinista es aquél en el que ningún parámetro del modelo es aleatorio, esto es, todos son conocidos con certeza y los valores de las variables no dependen de ninguna realización del azar. Estos valores se eligen basándose únicamente en la función objetivo y en las restricciones que deben satisfacer.

Un *problema determinista mixto 0-1 de dos etapas* se puede representar de la siguiente manera:

$$
\begin{array}{rlll}
(MIP): \ z_{MIP} = & \text{mín} & c_1\delta + c_2 x + q_1\gamma + q_2 y & \\
& s.a. & b_1 \leq A \begin{pmatrix} \delta \\ x \end{pmatrix} \leq b_2 & \\
& & h_1 \leq T \begin{pmatrix} \delta \\ x \end{pmatrix} + W \begin{pmatrix} \gamma \\ y \end{pmatrix} \leq h_2, & (2.1) \\
& & x, y \geq 0 & \\
& & \delta, \gamma \in \{0,1\} &
\end{array}
$$

donde c_1 y c_2 son los vectores de los coeficientes de la función objetivo para las variables enteras (δ) y continuas (x) de primera etapa, respectivamente. b_1 y b_2 son los límites inferiores y superiores, respectivamente, de las restricciones de primera etapa, y A es la matriz de coeficientes de las restricciones de primera etapa. q_1 y q_2 son los coeficientes en la función objetivo de las variables enteras (γ) y continuas (y), respectivamente, de segunda etapa, y h_1 y h_2 son los límites inferiores y superiores, respectivamente, de las restricciones de segunda etapa, mientras que T es la denominada *matriz tecnológica* y W la *matriz de recurso*.

Es habitual que las etapas que modeliza el problema hagan referencia a periodos de planificación temporales; sin embargo, no tiene que ser forzosamente así. De todas maneras, nosotros consideramos las etapas de decisión como fases temporales.

El modelo determinista no es suficiente para recoger la incertidumbre que puede aparecer en algún punto de decisión. Esta incertidumbre se puede reflejar en cualquiera de los parámetros del problema haciendo que éstos sean aleatorios, y se recoge mediante la definición de un conjunto de escenarios.

Cada escenario $\omega \in \Omega$ es una realización de los parámetros aleatorios y deterministas a lo largo de las diversas etapas del horizonte temporal, ψ^ω. A cada uno de los escenarios, $\omega = 1, ..., |\Omega|$, se le asocia una probabilidad de ocurrencia w^ω, $\omega \in \Omega$ de manera que $\sum_{\omega \in \Omega} w^\omega = 1$, donde Ω es el conjunto de escenarios.

Las decisiones tomadas en la primera etapa no dependen del escenario final que acontezca, por lo que las variables δ y x han de tomar el mismo valor en cada uno de ellos. Esta relevante condición se conoce como *restricción de no anticipatividad*, introducida en 1974 por Roger J-B Wets [33], y enunciada en 1991 por Rockafellar y Wets [23] de la siguiente manera:

> "Si dos escenarios, ω y ω', son idénticos considerando la información disponible sobre ellos desde la primera etapa hasta la etapa t incluida, entonces las decisiones a tomar bajo esos escenarios hasta la etapa t deben ser las mismas."

Las restricciones de no anticipatividad se pueden expresar de la siguiente manera:

$$\delta^\omega - \delta^{\omega'} = 0, \quad \forall \omega, \omega' \in \Omega, \quad \omega \neq \omega' \tag{2.2}$$

$$x^\omega - x^{\omega'} = 0, \quad \forall \omega, \omega' \in \Omega, \quad \omega \neq \omega' \tag{2.3}$$

Estas condiciones aseguran que se toman las decisiones de primera etapa sin conocer con exactitud la realización de la incertidumbre que se va a producir en la siguiente etapa, es decir, que fuerza a que se tome una única decisión para todos los escenarios.

En la Figura 2.1 podemos ver cómo se estructura la información en los distintos periodos, en este caso para $|\Omega| = 6$ escenarios. Cada nodo del árbol se asocia con un

grupo de escenarios, $g \in \mathcal{G}$, donde $\mathcal{G}$ representa el conjunto de grupos de escenarios. Definimos *un grupo de escenarios* para una etapa como un conjunto de escenarios cuya realización de los parámetros inciertos es la misma hasta dicha etapa. En cada etapa t aparece un subgrupo de escenarios $\mathcal{G}_t$ de manera que $\mathcal{G} = \sum_{t \in T} \mathcal{G}_t$.

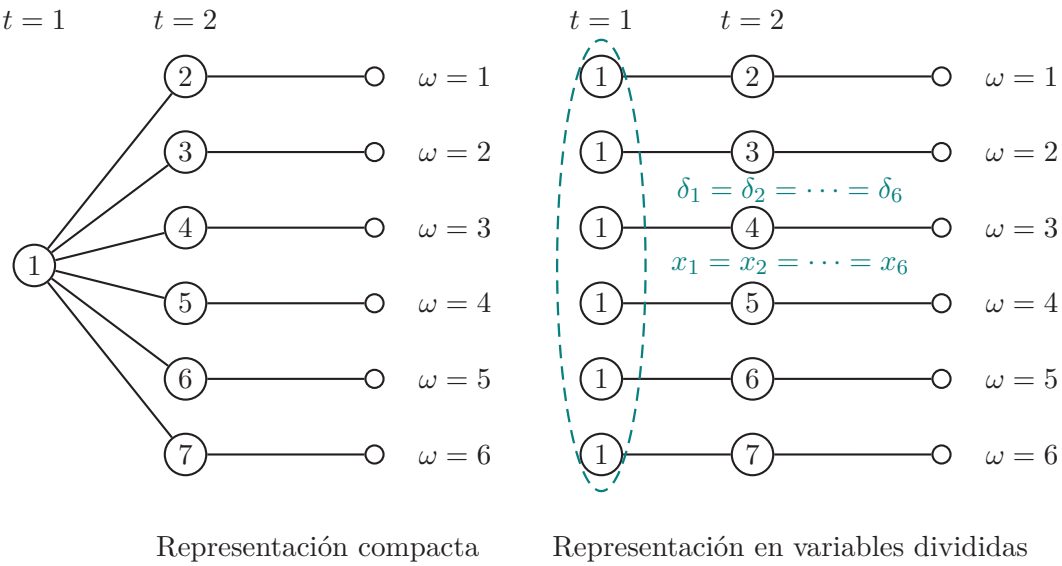

Figura 2.1: Árbol de escenarios

En los árboles de la Figura 2.1, en la primera etapa el subconjunto de escenarios está compuesto por un solo elemento $\mathcal{G}_1 = \{1\}$ y en la segunda etapa hay un elemento por cada escenario $\mathcal{G}_2 = \{2, 3, ..., 7\}$. Se observa que el concepto de grupo de escenarios corresponde con el concepto de nodos en el árbol de escenarios. Sin embargo, a veces nos interesa trabajar con el concepto de grupo, por ejemplo cuando queramos dividir el grupo de escenarios en distintos subgrupos para construir los clusters o racimos de escenarios.

En el árbol de la izquierda aparece una sola réplica de las variables de primera etapa para todos los escenarios. Se dice entonces que la no anticipatividad es implícita y la representación correspondiente es compacta.

En el árbol de la derecha se introduce una réplica de las variables de primera etapa para cada escenario, por lo que hay que añadir explícitamente las condiciones de igualdad de las variables en cada uno de los escenarios, esto es, $\delta_1 = \delta_2 = ... = \delta_6$ y $x_1 = x_2 = ... = x_6$. En este caso la representación es extendida o en variables divididas.

En ocasiones nos interesa formar *clusters* o *racimos de escenarios*. Éstos son particiones de los grupos de escenarios que permiten combinar la representación compacta y en variables divididas en las distintas etapas del problema.

Definición 2.1 *Un cluster o racimo de escenarios es un conjunto de escenarios cuyas restricciones de no anticipatividad se consideran de forma implícita en el modelo asociado.*

En la Figura 2.2 se representa el ejemplo anterior con dos elecciones diferentes en el número de clusters.

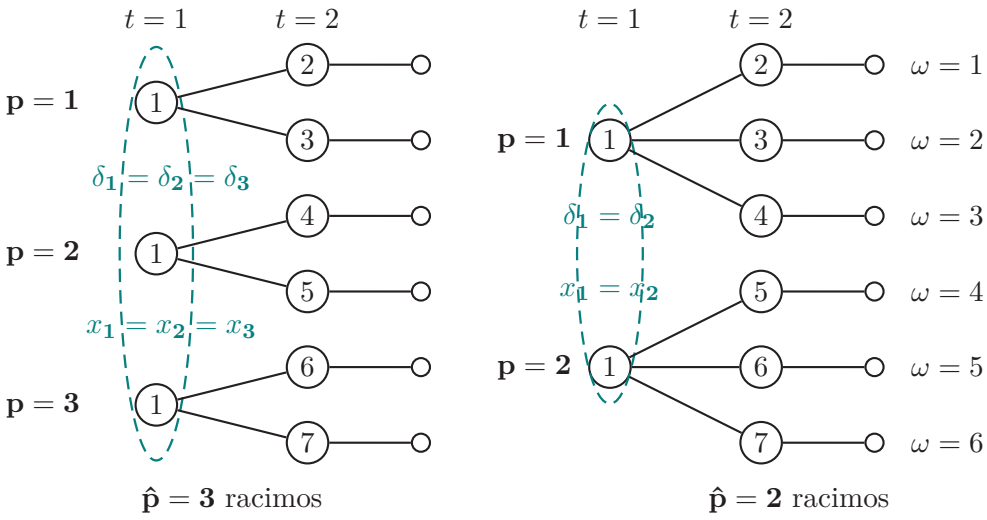

Figura 2.2: Partición en racimos de escenarios

En el árbol de la izquierda se han seleccionado $\hat{p} = 3$ racimos de escenarios, de modo que se han introducido tres copias de las variables de primera etapa, una para cada cluster, que se relacionan entre sí por las restricciones de no anticipatividad, es decir se cumple que $\delta_1 = \delta_2 = \delta_3$ y $x_1 = x_2 = x_3$. En cada cluster hay una única copia de las variables de primera etapa, por lo que la representación interna es compacta. El círculo punteado indica que las realizaciones de los parámetros inciertos son iguales bajo cualquiera de los clusters. Cada cluster está compuesto por dos escenarios, de manera que $\Omega^1 = \{1, 2\}$, $\Omega^2 = \{3, 4\}$ y $\Omega^3 = \{5, 6\}$.

El árbol de la derecha de la Figura 2.2 se ha descompuesto en $\hat{p} = 2$ clusters, donde cada uno de ellos tiene tres escenarios posibles, $\Omega^1 = \{1, 2, 3\}$ y $\Omega^2 = \{4, 5, 6\}$, por lo que se introducen dos copias de las variables de primera etapa. Estos escenarios están relacionados mediante las restricciones de no anticipatividad que aseguran que $\delta_1 = \delta_2$ y $x_1 = x_2$.

Se observa además que la Figura 2.1 también puede considerarse como distintas particiones en racimos de escenarios. La parte de la derecha sería una descomposición en $\hat{p} = |\Omega| = 6$ clusters y la parte de la izquierda en $\hat{p} = 1$.

2.3. Representaciones del DEM

Cualquier *problema estocástico mixto 0-1 de dos etapas* admite un *modelo determinista equivalente* o DEM (Deterministic Equivalent Model), término introducido por Wets en [32].

Para definir el problema de optimización, además de las variables a determinar (de primera y segunda etapa), hay que definir la función objetivo y las restricciones que han de cumplir dichas variables.

El DEM de un problema estocástico de dos etapas admite distintas representaciones. Una de ellas es la denominada *representación compacta*, dada por:

$$\begin{array}{rll} z_{MIP} = & \text{mín} & c_1\delta + c_2 x + E_{\psi^\omega}[\text{mín}\, q_1^\omega \gamma^\omega + q_2^\omega y^\omega] \\ & s.a. & b_1 \leq A \begin{pmatrix} \delta \\ x \end{pmatrix} \leq b_2 \\ & & h_1^\omega \leq T^\omega \begin{pmatrix} \delta \\ x \end{pmatrix} + W^\omega \begin{pmatrix} \gamma^\omega \\ y^\omega \end{pmatrix} \leq h_2^\omega, \quad \omega \in \Omega \\ & & x, y^\omega \geq 0, \quad \omega \in \Omega \\ & & \delta, \gamma^\omega \in \{0,1\}, \quad \omega \in \Omega \end{array} \tag{2.4}$$

donde c_1 y c_2 son los vectores de los coeficientes de la función objetivo para las variables δ y x de primera etapa, respectivamente. b_1 y b_2 son los límites inferiores y superiores, respectivamente, de las restricciones de primera etapa, y A es la matriz de coeficientes de las restricciones de primera etapa. Para cada escenario ω, q_1^ω y q_2^ω son los coeficientes en la función objetivo de las variables γ y y, respectivamente, de segunda etapa, y h_1^ω y h_2^ω son los límites inferiores y superiores, respectivamente, de las restricciones de segunda etapa, mientras que T^ω es la denominada *matriz tecnológica* y W^ω es la *matriz de recurso* bajo cada escerario ω, $\omega \in \Omega$, donde Ω es el conjunto de escenarios posibles considerados. Finalmente, E_{ψ^ω} representa la esperanza matemática con respecto a ψ^ω sobre el conjunto de escenarios.

El vector ψ^ω sobre el que se calcula la esperanza matemática está formado por las componentes estocásticas del problema, esto es:

$$\psi^\omega = (q_1^\omega, q_2^\omega, h_1^\omega, h_2^\omega, T^\omega, W^\omega)\,.$$

De esta forma, podemos reescribir la expresión de esta esperanza resultando la siguiente representación compacta del DEM:

$$\begin{array}{rll} z_{MIP} = & \text{mín} & c_1\delta + c_2 x + \displaystyle\sum_{\omega\in\Omega} w^\omega [q_1^\omega \gamma^\omega + q_2^\omega y^\omega] \\ & s.a. & b_1 \leq A \begin{pmatrix} \delta \\ x \end{pmatrix} \leq b_2 \\ & & h_1^\omega \leq T^\omega \begin{pmatrix} \delta \\ x \end{pmatrix} + W^\omega \begin{pmatrix} \gamma^\omega \\ y^\omega \end{pmatrix} \leq h_2^\omega, \quad \omega \in \Omega \\ & & x, y^\omega \geq 0, \quad \omega \in \Omega \\ & & \delta, \gamma^\omega \in \{0,1\}, \quad \omega \in \Omega \end{array} \tag{2.5}$$

donde w^ω es la *verosimilitud* o *probabilidad de ocurrencia* de cada escenario.

Una representación alternativa del DEM, adecuada para emplear técnicas de descomposición, es la denominada *extendida* o *en variables divididas*. Surge al introducir en el problema una réplica para cada variable de primera etapa y cada escenario, y añadir las condiciones de no anticipatividad de forma explícita en el modelo.

$$\begin{array}{rll} z_{MIP} = & \text{mín} & \displaystyle\sum_{\omega\in\Omega} w^\omega[c_1\delta^\omega + c_2x^\omega + q_1^\omega\gamma^\omega + q_2^\omega y^\omega] \\ & s.a. & b_1 \le A\begin{pmatrix}\delta^\omega\\ x^\omega\end{pmatrix} \le b_2, \quad \omega\in\Omega \\ & & h_1^\omega \le T^\omega\begin{pmatrix}\delta^\omega\\ x^\omega\end{pmatrix} + W^\omega\begin{pmatrix}\gamma^\omega\\ y^\omega\end{pmatrix} \le h_2^\omega, \quad \omega\in\Omega \\ & & \delta^\omega - \delta^{\omega'} = 0, \quad \forall\omega,\omega'\in\Omega, \quad \omega\neq\omega' \\ & & x^\omega - x^{\omega'} = 0, \quad \forall\omega,\omega'\in\Omega, \quad \omega\neq\omega' \\ & & x^\omega \ge 0, \delta^\omega\in\{0,1\}, \quad \omega\in\Omega \\ & & y^\omega \ge 0, \gamma^\omega\in\{0,1\}, \quad \omega\in\Omega. \end{array} \tag{2.6}$$

Hay que destacar que la relajación del conjunto de restricciones adicionales,

$$\begin{array}{rcl} \delta^\omega - \delta^{\omega'} & = & 0, \quad \forall\omega,\omega'\in\Omega, \quad \omega\neq\omega' \\ x^\omega - x^{\omega'} & = & 0, \quad \forall\omega,\omega'\in\Omega, \quad \omega\neq\omega' \end{array}$$

en el problema (2.6) hace que aparezcan $|\Omega|$ modelos mixtos 0-1 independientes, uno por escenario.

Por último podemos considerar la descomposición en clusters o racimos de escenarios. De este modo aparece una última representación del problema DEM, la *representación extendida sobre los racimos de escenarios*, que surge al incluir una copia de cada variable de primera etapa para cada cluster o racimo. Se observa que en este caso el número de copias introducidas es menor que en la representación en variables divididas sobre el conjunto de escenarios.

El modelo DEM en este caso tendría la siguiente representación:

$$\begin{array}{rll} z_{MIP} = & \text{mín} & \displaystyle\sum_{p=1}^{\hat{p}} w^p[c_1\delta^p + c_2x^p] + \sum_{p=1}^{\hat{p}}\sum_{\omega\in\Omega_p} w^\omega[q_1^\omega\gamma^\omega + q_2^\omega y^\omega] \\ & s.a. & b_1 \le A\begin{pmatrix}\delta^p\\ x^p\end{pmatrix} \le b_2, \quad p=1,...,\hat{p} \\ & & h_1^\omega \le T^\omega\begin{pmatrix}\delta^p\\ x^p\end{pmatrix} + W^\omega\begin{pmatrix}\gamma^\omega\\ y^\omega\end{pmatrix} \le h_2^\omega, \quad \omega\in\Omega^p,\ p=1,...,\hat{p} \\ & & \delta^p - \delta^{p'} = 0, \quad \forall p,p'\in\{1,...,\hat{p}\},\ p\neq p' \\ & & x^p - x^{p'} = 0, \quad \forall p,p'\in\{1,...,\hat{p}\},\ p\neq p' \\ & & x^p \ge 0, \delta^p\in\{0,1\}, \quad p=1,...,\hat{p} \\ & & y^\omega \ge 0, \gamma^\omega\in\{0,1\}, \quad \omega\in\Omega^p,\ p=1,...,\hat{p}. \end{array} \tag{2.7}$$

2.4. Técnicas de resolución

Asociado a cada modelo estocástico mixto 0-1 siempre se define su correspondiente modelo determinista equivalente o DEM. Se trata de un modelo de optimización mixto 0-1, en general de grandes dimensiones al aumentar el número de escenarios.

Entre las distintas técnicas conocidas para la resolución de este tipo de problemas, podemos distinguir entre las desarrolladas para la resolución de problemas DEM en representación compacta y en su representación extendida.

Dentro de los algoritmos para resolver problemas en representación compacta el más antiguo es el *método Simplex*, empleado para solucionar problemas de optimización lineal y que se emplea en la mayoría de software de optimización lineal. Cuando el problema lineal es de grandes dimensiones es habitual recurrir a algoritmos de descomposición, como por ejemplo el *método de descomposición de Benders*.

Para resolver problemas enteros en representación compacta uno de los algoritmos más conocidos es el de *Ramificación y Acotación* conocido como *Branch & Bound*, que resuelve, en cada nodo del árbol de búsqueda, los submodelos lineales que se generan al relajar las restricciones enteras, bifurcando sobre el valor de dichas variables. Esta relajación permite obtener cotas para el valor de la función objetivo.

En los últimos años se ha investigado mucho sobre los algoritmos de resolución para problemas en representación extendida. Se utilizan algoritmos de descomposición, como el *método de descomposición Lagrangiana* o el *método de Bifurcación y Fijación Coordinada*, conocido como BFC (*Branch & Fix Coordination*). Este último lo desarrollamos en profundidad en el Capítulo 3.

Para la implementación de cualquiera de estas técnicas es necesario el uso de un solver de optimización. En esta área también se ha avanzado mucho y se han incorporado técnicas computacionales que mejoran en gran medida los software existentes hasta el momento. Como primer solver de optimización utilizamos COIN-OR, que comparamos con CPLEX. Este último incorpora técnicas de preproceso y paralelización, que hacen que la velocidad de resolución sea considerablemente mayor respecto a COIN-OR.

En esta sección presentamos los algoritmos más destacados para resolver cada tipo de problemas de optimización y describimos las principales diferencias entre los mismos.

2.4.1. Técnicas para la representación compacta

El problema DEM en representación compacta puede representar un problema estocástico mixto 0-1 de dos etapas, pero de igual manera podría representar un

problema lineal.

En el caso de los problemas lineales, el método de resolución más utilizado es el Simplex, que fue desarrollado en el año de 1947 por el estadounidense George Bernard Dantzig y el ruso Leonid Vitalievich Kantorovich (que comparte premio Nobel de Economía con T.J. Koopmans, 1976). Su intención era crear un algoritmo capaz de solucionar problemas de optimización de m restricciones y n variables.

El método Simplex es un método analítico de solución de problemas de optimización lineal capaz de resolver modelos más complejos que los resueltos mediante el método gráfico[1] sin restricción en el número de variables. Se trata de un método iterativo que permite ir mejorando la solución en cada paso. El proceso concluye cuando no es posible seguir mejorando más dicha solución.

Este algoritmo, usando el criterio del peor caso, está clasificado como un algoritmo de complejidad de tipo exponencial, es decir, que el tiempo de ejecución puede llegar a crecer exponencialmente en función del problema a resolver. El Simplex necesita a lo sumo $2n - 1$ iteraciones para encontrar la respuesta (n es el número de variables). El número de soluciones básicas factibles es menor o igual a

$$\left(\begin{array}{c} m \\ n \end{array} \right) = \frac{m!}{n!(m-n)!}$$

Sin embargo, ha resultado ser un método tremendamente eficiente en la práctica.

Generalmente los problemas de optimización que nos encontramos en la vida real no incluyen sólo variables continuas, sino que en su mayoría aparecen variables enteras que dificultan la resolución de estos problemas. La técnica más utilizada para resolverlos es el método Branch & Bound. Éste realiza una búsqueda de la solución en una secuencia de procesos en los que se intercalan dos fases: la bifurcación sobre los valores de una variable de decisión entera, y la acotación del espacio de búsqueda. Para ello realiza relajaciones lineales del problema, lo que permite ir descomponiéndolo en subproblemas de menor tamaño, descartando paulatinamente aquéllos cuya función objetivo exceda el valor de la mejor solución encontrada hasta ese momento (en caso de problemas de minimización). Es decir, a lo largo de la ejecución se recorre un árbol de búsqueda, que se va generando al relajar las restricciones enteras y resolver en cada nodo subproblemas de programación lineal.

La relajación lineal y la búsqueda son, en efecto, los dos procesos fundamentales del método de Branch & Bound y los que determinan la eficiencia del mismo.

La relajación lineal permite ir obteniendo límites para los valores de la función objetivo. El árbol se construye y se recorre siguiendo los siguientes criterios:

- Mientras que la solución obtenida en la resolución de la relajación no satisfaga

[1] El método gráfico resuelve problemas lineales mediante la representación gráfica de las restricciones y la función objetivo. Para modelos con tres o más variables este método se vuelve ineficaz.

todas las restricciones del problema original se generan nuevas ramas en el árbol de búsqueda, bifurcándose el nodo en los valores enteros más cercanos a las variable continuas halladas.

- Un nodo es abandonado cuando el valor de la función objetivo de la relajación es peor que la mejor solución entera obtenida hasta ese momento, o bien cuando el subproblema es infactible.
- Si la solución relajada satisface todas las restricciones del problema original y es mejor que la mejor solución factible encontrada hasta el momento, entonces se actualiza el valor de la mejor solución factible.
- La búsqueda termina cuando se demuestra que no existe mejor solución que la encontrada.

Existen procedimientos para mejorar la eficiencia de esta búsqueda, como es la introducción de planos de corte, esto es el método Branch & Cut, lo que reduce el árbol de búsqueda, y acelera por tanto la obtención de la solución.

Estos algoritmos descritos son ineficientes cuando la dimensión del problema es muy grande. Para resolver problemas lineales de grandes dimensiones, en 1962 Benders (ver [6]) propuso el método denominado descomposición de Benders. Éste consiste en separar el modelo en dos problemas lineales, uno con un conjunto de restricciones generales, denominado *Problema Maestro Relajado* (PMR) y otro denominado *Problema Auxiliar* (PA). La idea principal de este procedimiento se basa en el flujo de información que circula de un problema a otro y que conduce a garantizar el óptimo. Una condición que garantiza el éxito del procedimiento, y que en la práctica suele ser olvidada, es la exigencia de que la región factible[2] del PMR sea acotada. Esta separación soluciona el problema de las dimensiones ya que conduce a modelos bastante más pequeños.

Además, la aplicación de esta técnica de descomposición a problemas estocásticos permite abordar problemas con un número de escenarios y etapas muy superior al que admite la optimización clásica, ya que adicionalmente el PA se puede descomponer en submodelos independientes. La descomposición de Benders propone separar en subproblemas las decisiones tomadas en diferentes etapas. Para ello se necesita que las decisiones de una etapa sólo dependan de las consecuencias de las decisiones tomadas en la etapa anterior. Con esta descomposición se plantea un problema por cada etapa, y en éste se incluye tanto la parte correspondiente a la etapa que estamos resolviendo como la parte que liga esta etapa a las decisiones tomadas en la etapa anterior.

[2]La región factible, determinada por el conjunto de restricciones, es el conjunto de puntos que satisfacen a la vez todas las restricciones.

2.4.2. Técnicas para la representación extendida

Cuando se trata con problemas de gran tamaño que no pueden ser resueltos en los equipos informáticos disponibles, suele recurrirse a técnicas de descomposición, que permiten fragmentar el problema y coordinar la resolución de los subproblemas para alcanzar la solución óptima del problema completo. En este sentido, las técnicas de descomposición se pueden ver como estrategias de partición del árbol de escenarios y de resolución coordinada de los fragmentos del mismo. Este proceso de resolución es de naturaleza iterativa y amplía el tiempo de resolución total, por lo que debe ser evitado siempre que sea posible la resolución directa. En el caso de los problemas de optimización estocástica, el empleo de técnicas de descomposición permite considerar una gran cantidad de escenarios o de problemas con una modelización más compleja.

Para poder aplicar estas técnicas de descomposición se debe introducir la representación en variables divididas y la representación extendida bajo los clusters del DEM. Los métodos de descomposición más relevantes son la descomposición Lagrangiana y el BFC (Branch & Fix Coordination).

La relajación Lagrangiana [13] trata de separar, dentro de cada etapa, las decisiones para grupos de variables que están relacionadas entre sí. Es decir, se pueden localizar conjuntos de variables que están muy conectadas con otras etapas, pero poco relacionadas con otras variables de la misma etapa. La relajación Lagrangiana consiste en eliminar las restricciones de complicación, de forma que se puedan resolver independientemente las distintas partes del problema. La coordinación de las soluciones de cada uno de estos subproblemas se lleva a cabo por un *problema maestro*, cuyo objetivo es forzar el cumplimiento de las restricciones de complicación. El método iterativo resuelve alternativamente el problema maestro y los subproblemas hasta que se cumplen las restricciones de complicación que se han eliminado al formular los subproblemas.

Cuando la estocasticidad se introduce en el problema mediante un árbol de escenarios, de manera natural surge la descomposición por escenarios a lo largo de todas las etapas. El árbol está formado por escenarios independientes salvo por el hecho de que comparten los nodos iniciales. Si no tuviera que cumplirse esta condición, podrían resolverse los problemas de cada escenario de forma independiente. La condición de que los escenarios compartan los nodos iniciales es lo que hemos denominado condición de no anticipatividad. Esta condición es la restricción de complicación que se elimina por medio de la relajación Lagrangiana.

En este trabajo nos centramos en los métodos desarrollados para la resolución de los problemas estocásticos mixtos de dos etapas, que dan un paso más allá en la programación ya que combinan las dificultades de los problemas clásicos: problemas estocásticos de grandes dimensiones, y las variables enteras de los problemas de optimización discreta.

Muchos investigadores han estudiado las propiedades y algunas aproximaciones

a soluciones para este tipo de problemas en la última década, Klein, van del Verk y Schultz han encontrado los mejores resultados en esta área, ver [17] y [25], entre otros.

La forma más simple de la programación entera dos etapas contiene variables 0-1 en la primera etapa y variables continuas en la segunda etapa. Birge y Louveaux y Laporte y Louveaux, en [7] y [20], aplican el procedimiento Branch & Cut para cada problema, basado en la descomposición de Benders [6]. En los artículos de Alonso-Ayuso, Escudero, Garín, Ortuño y Pérez ([2], [3] y [4]) se describe la metodología Branch & Fix Coordination (BFC), ver [5] para resolver problemas para las aplicaciones sobre la planificación de la producción, aunque esta aproximación no permite variables continuas en la primera etapa o variables 0-1 en la segunda.

Cuando las variables de primera etapa son todas enteras, se puede justificar fácilmente que se obtiene una solución en un número finito de pasos basándose en la idea de ramificar sobre todos y cada uno de los posibles valores de estas variables de primera etapa. En [22], [28] y [29] se proponen algoritmos de descomposición basados en la técnica Branch & Cut para resolver problemas estocásticos de dos etapas con variables de primera etapa puramente enteras y variables de segunda etapa mixtas 0-1. En [27] y [30], Sen y Sherali y Sherali y Zhu proponen una descomposición similar a la empleada en el algoritmo Branch & Cut donde se desarrolla el método de la descomposición de Benders modificado.

En [8] y [14] se diseña un algoritmo Branch & Bound para problemas con variables mixtas en ambas etapas. Sin embargo, sus aproximaciones se centran más en usar la relajación Lagrangiana para obtener mejores cotas, y menos en ramificar y fijar variables. En [10], Escudero, Garín, Merino y Pérez desarrollan una aproximación similar para problemas de dos etapas, donde las variables 0-1 y las variables continuas tienen elementos no nulos en la primera etapa y hay variables continuas en la segunda etapa. Esta aproximación utiliza la metodología BFC para satisfacer las restricciones de no anticipatividad de las variables 0-1 y a su vez se utiliza la descomposición de Benders para resolver los submodelos lineales definidos en cada Familia de Nodos Gemelos (TNF).

En este trabajo describimos la metodología BFC, que se introduce en [12] para resolver problemas estocásticos mixtos 0-1 de dos etapas. Se describe con detalle en el Capítulo 3.

2.4.3. Software de optimización

El acelerado proceso de mejora de los ordenadores ha permitido que se haya podido avanzar, casi a la misma velocidad, en la eficiencia de resolución de problemas de optimización. La computación es una parte muy importante en la obtención de la solución de estos problemas, ya que tener un buen algoritmo de resolución no lo es todo. Cuando nos enfrentamos ante un problema de optimización estocástica

hay un gran número de cálculos que hay realizar para llegar al óptimo, aunque las dimensiones de este problema no sean elevadas. De esta manera, si nos planteásemos un problema en el que las dimensiones creciesen ligeramente, resolverlo con lápiz y papel sería impensable.

En este contexto los ordenadores son de gran utilidad. Cuanto más avance la computación, los problemas que seamos capaces de resolver serán de mayores dimensiones. El objetivo de este trabajo es presentar un algoritmo con el que siempre lleguemos a la solución en caso de que exista, o que sea capaz de decir que el problema es infactible. Pero a su vez, se pretende mejorar continuamente el tiempo de ejecución. Para ello utilizamos dos optimizadores: COIN-OR y CPLEX.

Ambos optimizadores contienen la implementación de los algoritmos básicos que utilizamos para la resolución de los problemas lineales y enteros, Simplex y Branch & Bound, que hemos descrito anteriormente. Sin embargo, ambos optimizadores tienen características completamente distintas. La principal diferencia radica en que CPLEX incorpora técnicas de preproceso y paralelización, que disminuyen considerablemente el tiempo de resolución.

El optimizador COIN-OR nos permite resolver problemas de optimización lineal y entera. Sin embargo, presenta una serie de limitaciones en cuanto al tiempo de resolución y dimensiones del problema. A medida que el problema entero crece en dimensiones, el tiempo que tarda en encontrar la solución óptima, en caso de tenerla, aumenta mucho. Esto es lógico, como ya hemos comentado, ya que la dificultad del problema aumenta.

Esta limitación ha hecho que se sigan desarrollando algoritmos de optimización con el fin de disminuir el tiempo de ejecución. Los problemas que se necesitan resolver en la realidad presentan un gran número de variables y de restricciones, es decir, son de altas dimensiones.

Por su parte, CPLEX dispone de un motor de resolución que está reconocido por la comunidad de investigación en optimización matemática y usuarios de grandes compañías como el motor de optimización más eficiente de programación lineal, tanto por la velocidad de ejecución como por el tamaño de los problemas que resuelve. La eficiencia de CPLEX es en efecto debida a la especialización de los algoritmos y preresolutores que utiliza, así como a la adaptación de dichos algoritmos a las nuevas arquitecturas de los procesadores.

La tecnología de programación matemática del optimizador CPLEX permite mejorar la eficiencia, reducir costes y aumentar la rentabilidad. Además, proporciona solvers de programación matemática (programación lineal, entera mixta, cuadrática, etc.) flexibles y de alto rendimiento. En este trabajo tan sólo se utiliza para resolver problemas de optimización lineal y enteros-mixtos.

Con este optimizador se pueden resolver problemas con un número elevado de variables y restricciones, como son los problemas que podemos encontrarnos en la

realidad. Para ello proporciona la potencia necesaria para su resolución, así como la velocidad requerida para el análisis interactivo que se lleva a cabo hoy en día en las aplicaciones de soporte de decisiones.

Este optimizador se puede utilizar a través del lenguaje de programación C++, que es el que se utiliza para implementar el algoritmo BFC en este trabajo.

CPLEX tiene dos ventajas claras sobre COIN-OR: el preproceso y la paralelización. Estos dos añadidos permiten que la resolución de los problemas se consiga en menor tiempo. El preproceso consiste en tratar previamente el problema para conseguir un problema equivalente y más sencillo a resolver, y la paralelización reparte el trabajo que tiene que hacer el ordenador consiguiendo que el tiempo se reduzca considerablemente.

El preproceso llevado a cabo por CPLEX se basa en simplificar restricciones, reducir las dimensiones del problema y eliminar redundancias. Con todo esto intenta reducir el tamaño del problema haciendo deducciones sobre la naturaleza de cualquiera de las soluciones óptimas del problema. Elimina variables y restricciones mediante sustitución. Este preproceso es beneficioso para la velocidad de resolución. Además la solución que proporciona CPLEX está en términos de la formulación original a pesar de que internamente reformule equivalentemente el problema.

A través del preproceso se pretenden alcanzar dos objetivos. En primer lugar se detecta si el problema tiene solución factible, y en segundo lugar se intenta reducir el número de familias de nodos activos. Si se consiguen estos dos objetivos el procesador no perderá tiempo tratando de resolver problemas imposibles y se dedicará exclusivamente a las partes significativas del problema.

La segunda ventaja a la que hacíamos mención es la paralelización. CPLEX explota cada vez más el paralelismo de los ordenadores actuales. Estas máquinas tienen unos procesadores que pueden constar de más de un núcleo, pudiendo aprovechar así las características del algoritmo Branch & Bound, haciendo que cada uno de los núcleos recorra una rama diferente del árbol, reduciendo de manera notable el tiempo de cómputo.

Las técnicas de preproceso y paralelización incluidas por CPLEX abren una puerta a futuras investigaciones sobre las técnicas de resolución de problemas de optimización estocástica. La idea es incorporar estas técnicas en los algoritmos de descomposición que hoy en día se están utilizando; de esta manera conseguiríamos que estos algoritmos mejorasen su eficiencia en problemas de grandes dimensiones.

2.5. Aplicaciones

Los modelos de optimización estocástica tienen una gran cantidad de aplicaciones en la vida real. Muchos de los problemas de selección generados en cualquiera de las áreas de trabajo remiten a un problema de optimización. En éste hay que seleccionar los mejores valores de una serie de variables para que el valor de una cierta función sea el máximo o mínimo posible.

Existen numerosas aplicaciones de los problemas de optimización estocástica e incluyen, entre otras, problemas de transporte y localización óptima, de asignación de aviones a rutas aéreas, de expansión de la capacidad de una firma, de planificación de la producción, de planificación de inversiones financieras, de generación y despacho económico de sistemas eléctricos y de inversiones y manejo óptimo de recursos naturales.

Una aplicación concreta es el problema de coordinación hidrotérmica a medio plazo, que se utiliza para la gestión de las centrales del sistema eléctrico. Este modelo se inscribe dentro de la jerarquía de modelos de planificación eléctrica.

En el campo financiero hay multitud de modelos bajo incertidumbre, por lo que es apropiado tratarlos como problemas estocásticos. Algunas de las aplicaciones en este campo son la gestión de activos y pasivos, la estructuración de una cartera de títulos hipotecarios, la planificación estocástica de carteras, la inmunización de carteras de bonos, el análisis del peor escenario para posiciones en opciones, los modelos de media-varianza o los modelos de utilidad, entre otros.

La mayoría de los modelos de optimización financiera se pueden clasificar en dos grandes grupos de acuerdo con su objetivo primario: *gestión de riesgos* e *ingeniería financiera*. Los modelos de gestión de riesgos se usan en la selección de carteras con exposición a diferentes riesgos. Los modelos de ingeniería financiera se utilizan para estructurar nuevos instrumentos financieros para un objetivo específico según las preferencias del inversor, o para coger ventaja de las oportunidades de arbitraje.

La gestión de riesgo está centrada, principalmente, en seleccionar riesgos a los que estás expuesto y de los que te quieres inmunizar. En segundo lugar, valora el riesgo de diferentes activos, y en tercer lugar, construye y mantiene las carteras con unas características de riesgo-rentabilidad específicas. El riesgo financiero es multidimensional. Entonces, un prerrequisito para seleccionar la exposición al riesgo es identificar las distintas formas de riesgo que se pueden presentar, entre las que se encuentran:

- Riesgo de mercado
- Riesgo de estructura (shape)
- Riesgo de volatilidad

- Riesgo de sector
- Riesgo de moneda
- Riesgo de crédito
- Riesgo de liquidez
- Riesgo residual

La gestión de riesgo no es más que tomar posiciones en asignaciones específicas o genéricas del activo. Un problema importante, en cambio, es que las características genéricas no se negocian en el mercado. El problema es aún más complicado cuando nos percatamos de que la mayoría de los inversores hacen frente a muchos requerimientos institucionales para establecer una cartera.

El escenario en el que se describen estos problemas explica el papel importante de los modelos de optimización en la gestión de riesgo. Las técnicas de optimización matemática pueden identificar de forma efectiva la solución de problemas de planificación de carteras con muchas restricciones. Usando optimización matemática podemos encontrar soluciones factibles al problema, o demostrar que una exposición a un riesgo es inaccesible.

Los nuevos productos creados por la ingeniería financiera se han introducido satisfactoriamente en los mercados financieros, ya que estos nuevos instrumentos generan algunas características de los riesgos genéricos, haciendo más fácil el control del riesgo. Como ya hemos discutido, la gestión del riesgo es un proceso de crear una cartera de activos con características concretas, a partir de paquetes existentes. Sin embargo, la cartera resultante no se vende como un producto estándar, por lo que el objetivo de la ingeniería financiera es diseñar productos que se añadan al mercado. Para crear los paquetes a partir de instrumentos existentes la ingeniería financiera mejora la capacidad de que los productos se negocien en el mercado y adecúa las necesidades de los inversores individuales.

Como en el caso de la gestión de riesgo, la mera existencia de restricciones hace que la optimización sea una herramienta clave en la ingeniería financiera. La optimización matemática asegura eficientemente la viabilidad e identifica las dificultades cuando no es viable. Sin embargo, también aparecen oportunidades genuinas para la optimización en la ingeniería financiera; la razón es que generalmente hay más flexibilidad en las restricciones estructuradas.

A continuación presentamos algunos modelos de optimización en el campo de las finanzas mencionados anteriormente.

2.5.1. Modelos de planificación estocástica multiperiodo

En el grupo de gestión de riesgos podemos encontrar los modelos de planificación estocástica multiperiodo, que se valen de las más novedosas técnicas de optimización estocástica para resolverse.

Muchos de los modelos económicos son estáticos y de un solo periodo (como pueden ser los problemas de inmunización de carteras de un bono, de inmunización ante un factor de riesgo, de elección de posiciones en opciones analizando el peor escenario, los modelos de media-varianza, los de utilidad, etc.). El modelo de utilidad considera varios periodos, pero las decisiones de inversión que se toman en el periodo t no dependen de las expectativas de ningún periodo a partir de $t + 1$. En los casos donde la incertidumbre aparece en cualquiera de los estados del horizonte de planificación (como suele ocurrir) y cuando se pueden llevar a cabo acciones de corrección entre los periodos t y $t+1$, los modelos de optimización estocástica suelen ser los más apropiados.

Algunos modelos no son muy comunes actualmente en la planificación financiera debido a su complejidad y a la necesidad de una cantidad de datos enorme para resolverse. Sin embargo, han aparecido muchos modelos interesantes en la literatura. Algunos de los más importantes se han desarrollado para problemas concretos en bancos, en gestión de activos, y se usan con datos reales para probar su superioridad sobre modelos más elementales, deterministas. Algunos ejemplos de modelos de planificación estocástica son:

1. El modelo dinámico para la gestión de carteras bancarias de Bradley y Crane (1972),
2. El modelo activo/pasivo bancario de Kusy y Ziembra (1986),
3. El modelo de gestión de cartera de activos de Mulvey y Vladimirou (1992),
4. El modelo de programación estocástica para la financiación respaldada por hipotecas de Zenios (1991), y
5. Los modelos de dedicación estocásticos de Hiller y Schaack (1990).

No existe un modelo de planificación estocástica unificado que abarque los detalles de todos estos modelos, pero se puede presentar una formulación bastante general, que es la que a continuación se detalla, y que se asemeja al planteamiento general que hemos hecho del DEM.

Siguiendo una notación similar a la estableciada hasta ahora, sea Ω el conjunto de escenarios posibles y definimos:

- x: vector de decisiones de inversión para el primer periodo, tomadas en base a la información conocida en $t = 0$,

- y^ω: decisiones del segundo periodo tomadas bajo cada escenario $\omega \in \Omega$,
- w^ω: probabilidad de ocurrencia del escenario $\omega \in \Omega$,
- $U(x, y^\omega)$: función utilidad de la rentabilidad, como función de las variables de decisión.

Obviamente $y^\omega, \forall \omega \in \Omega$ tienen que determinarse a priori, antes de que se dé cualquier escenario. Por otro lado, el modelo se simplifica a una secuencia de modelos uniperiodo resueltos después de que $\omega \in \Omega$ se haya observado. Las decisiones x e y^ω se toman de manera que se maximice una función de utilidad esperada (esta esperanza se calcula como la combinación de los valores de utilidad bajo todos los escenarios, cada uno ponderado por su probabilidad w^ω). Al mismo tiempo se deben satisfacer restricciones de contabilidad, política, diversificación y otras restricciones bajo cada escenario.

Usando las matrices A, T^ω y W^ω para representar las restricciones generales, el modelo de optimización DEM correpondiente se puede representar de la siguiente manera:

$$\begin{aligned} max_{x, y^\omega \in \mathcal{R}^I} \quad & \sum_{\omega \in \Omega} w^\omega U(x, y^\omega) \\ s.a. \quad & Ax = b \\ & T^\omega x + W^\omega y^\omega = h^\omega, \forall \omega \in \Omega \\ & x \geq 0 \\ & y^\omega \geq 0, \forall \omega \in \Omega \end{aligned} \tag{2.8}$$

Hay que destacar que el conjunto de restricciones se impone para cada escenario, de manera que el problema resultante es un programa no lineal de grandes dimensiones. Las decisiones de primera etapa, x, no dependen del conocimiento sobre el futuro y por tanto, son idénticas bajo cada uno de los escenarios.

La extensión más importante del modelo (2.8) está en la planificación multiperiodo. Desafortunadamente, el modelo contiene una explosión combinatoria de escenarios a medida que el número de periodos aumenta. Los modelos de planificación estocástica continúan estando infrautilizados en la modelización financiera. Sin embargo, la necesidad de capturar la incertidumbre de los caminos sitemáticos es de gran importancia en el entorno de las aplicaciones. Esto aporta una área abierta de gran interés para futuras investigaciones.

2.5.2. Modelo de gestión de activos y pasivos

El objetivo de este modelo es maximizar el valor neto descontado de los beneficios bancarios menos la penalización esperada de los costes por infactibilidad con respecto a restricciones débiles, sujeto a numerosas restricciones sobre los presupuestos, liquidez, estructura de flujos de caja, además de restricciones políticas y legales.

Hay diversas fuentes de incertidumbre, como los rendimientos, los tipos de interés, los depósitos de efectivo, etc.

Este modelo fue creado por Kusy y Ziemba (ver [19]) y centra la aleatoriedad en los depósitos de flujo dando valores deterministas fijos a los rendimientos y tipos de interés. En un primer momento se modelizó como un problema estocástico dos etapas multiperiodo lineal con recurso simple[3], que puede resolverse con las técnicas de resolución actuales si las variables aleatorias son discretas.

2.5.3. Estructuración de carteras de títulos con garantía hipotecaria

El algoritmo BFC se puso a prueba por primera vez en modelos dos etapas en [10], donde se resolvía el problema de estructurar una cartera de *títulos con garantía hipotecaria* (MBS, de sus siglas en inglés, Mortgage-Backed Securities). En adelante nos referiremos a este problema como aplicación MBS.

La titulización temporal está ideada para dividir el MBS y su devolución en pequeñas piezas como un activo financiero. El nuevo activo se devuelve como un préstamo hipotecario y su estructura de rendimiento tiene la forma de un conjunto de obligaciones a satisfacer a lo largo del tiempo. Hay que destacar que los títulos están cuasi-amortizados a lo largo del horizonte de planificación, el rendimiento de los mismos está basado normalmente en rentas ajustables a lo largo del tiempo, un principal de un préstamo puede ser totalmente prepagado y la duración de la cartera raramente iguala la duración de las obligaciones y entonces, hay un riesgo adicional de impago del rendimiento de la inversión. La complejidad del problema aumenta con la volatilidad de los tipos de interés y, por tanto, con el valor incierto de las carteras MBS.

Un título puede definirse como un activo que da el derecho a una ganancia variable a lo largo de un horizonte del tiempo. En este caso, al activo es un derecho financiero que incluye un principal y un rendimiento garantizado por una hipoteca, cuyo principal puede ser pagado por adelantado e igualmente puede ser retrasado.

Dado un conjunto de periodos de tiempo, un conjunto de activos, en concreto títulos con garantía hipotecaria, y un presupuesto o capital inicial para invertir, la aplicación MBS consiste en determinar el subconjunto de activos que serán incluidos en la cartera así como la fracción del valor nominal a considerar para cada uno, bajo incertidumbre en la trayectoria del tipo de interés a lo largo del horizonte de planificación.

Una estructuración factible de una cartera tiene que satisfacer dos tipos de res-

[3]Un modelo de recurso simple es aquél en el que todas las decisiones a tomar deben ir fijadas desde un principio, manteniéndose invariantes a pesar de que en periodos de tiempo posteriores se disponga más información sobre el escenario que puede acaecer.

tricciones:

1. Restricciones de primera etapa para relacionar los títulos. Por ejemplo, un límite superior en el número de títulos que se incluyan en la cartera, equilibrio en el valor nominal total de los diversos tipos de títulos, relaciones de exclusividad e implicación entre los distintos tipos, etc.

2. Restricciones de segunda etapa para analizar el resultado de invertir en una cartera de títulos a largo plazo sobre los distintos escenarios. Por ejemplo, límites inferiores y superiores del efectivo neto disponible en dichos periodos bajo cada escenario, el requisito de que el valor actual de la cartera no sea inferior al valor actual de las obligaciones bajo cada escenario, el requisito de que el valor absoluto de la diferencia entre la duración unitaria de la cartera y la duración unitaria del conjunto de títulos no superen unos valores prefijados, etc.

Además, la entrega o acuerdo de un MBS entre los comerciantes primarios - agencias federales y bancos de inversión- es liderada por un gran conjunto de normas y regulaciones para asegurarse prácticas uniformes. Aún si se ignora la oportunidad de beneficio, la necesidad de incluir los normas establecidas, junto con las restricciones comentadas, crea un problema complejo. Exponiendo una por una las bases, se desarrolla un modelo matemático.

La función de utilidad considerada por los autores en [10] es minimizar la diferencia esperada entre la duración de la cartera y la duración de las obligaciones, aunque hay muchas otras posibilidades. Esta elección es una forma de reducir o eliminar el riesgo de pérdida debido a movimientos adversos del precio de los activos, es decir, es una manera de proteger el rendimiento de la inversión frente a pequeños cambios en el tipo de interés a lo largo del horizonte temporal.

Este modelo planteado puede convertirse en un programa entero-mixto aún más complejo que no se puede resolver con los optimizadores existentes de programación entera. Actualmente se considera uno de los problema abiertos en ingeniería financiera. Generalmente se trata con procedimientos heurísticos para garantizar que se cumplen las restricciones. Este problema recibe la atención de investigadores para conseguir soluciones efectivas.

Capítulo 3

Algoritmo BFC (Branch-and-Fix Coordination)

El algoritmo de Ramificación y Fijación Coordinada BFC (de sus siglas en inglés, Brach and Fix Coordination) está diseñado para resolver problemas de optimización estocástica mixtos 0-1 de dos etapas, donde las variables continuas y enteras tienen elementos no nulos en las restricciones de cualquiera de las dos etapas. Este procedimiento permite resolver el problema aún cuando las variables continuas no están acotadas, y el carácter estocástico del problema, discretizado en un conjunto finito de escenarios, puede aparecer en cualquier parte del modelo.

Este esquema se describe originariamente en [4] para modelos multietapa enteros 0-1 y bietapa mixtos 0-1 donde tan solo en la primera etapa aparecen variables enteras. En [2] y [3] se pueden encontrar aplicaciones de este algoritmo a modelos mixtos 0-1 de dos etapas. Para el caso de problemas de optimización estocástica mixtos 0-1 donde en ambas etapas pueden aparecer tanto variables continuas como enteras, se describe por primera vez la extensión del BFC en [12].

El objetivo de este trabajo es resolver un modelo general similar al presentado en formulación extendida en (2.6), aunque utilizamos equivalentemente la formulación descrita en (2.7) y describimos el esquema del algoritmo BFC para los problemas relacionados con cada uno de los clusters o racimos de escenarios. De este modo nos aseguramos de que las variables δ sean enteras, de manera que las condiciones de no anticipatividad (2.2) se satisfagan mientras se seleccionan nodos y variables de ramificación. Teniendo en cuenta los valores que deben tomar todas las variables enteras, se construye el árbol con todas las posibles bifurcaciones de las variables enteras a 0 ó a 1, surgiendo el denominado *árbol de ramificación y fijación* para cada cluster o racimo de escenario.

Esta técnica basada en el Brach & Bound, coordina la selección de la Familia de Nodos Gemelos (TNF, de sus siglas en inglés *Twin Node Family*) y la selección de la

variable a ramificar en cada uno de los submodelos a optimizar. Estos subproblemas son modelos mixtos 0-1 que resultan del modelo (2.7) y de cada TNF entera, de manera que las restricciones de no anticipatividad (2.3) para las variables continuas de primera etapa también se satisfacen. Del mismo modo que coordina la selección de variables a fijar, también coordina el corte de las distintas ramas del árbol.

Esta aproximación no sólo selecciona nodos y variables sino que también corta el nodo BF asociado a un escenario, determina la variable fijada y las cotas de la función objetivo de los subproblemas correspondientes a los nodos. Se pueden encontrar aproximaciones de descomposiciones similares a ésta en la literaturas, ver [8], [14], [18], [21] y [24], entre otros. Sin embargo, éstas se centran más en usar la relajación Lagrangiana de las restricciones (2.2) para obtener buenas cotas inferiores, en vez de en ramificar y fijar variables. De todas maneras, también se puede añadir la relajación Lagrangiana y la descomposición de Benders al esquema que proponemos aquí. Ver para ello [10], [11] y [25].

A continuación se detallan los conceptos básicos para el desarrollo del esquema BFC, así como las bases teóricas. Al mismo tiempo se describen los pasos de este algoritmo y se presenta el pseudocódigo completo necesario para su implementación.

3.1. Definiciones previas

Comenzamos presentando las definiciones de los conceptos fundamentales, tomadas de [12], y los modelos de optimización que intervienen en la metodolgía BFC, para problemas asociados al escenario $\omega \in \Omega$ ignorando las restricciones de no anticipatividad (2.2) y (2.3) del problema (2.6). Este problema se puede expresar de la siguiente manera:

$$
\begin{array}{rlll}
(MIP^{\omega}): \ z^{\omega}_{MIP} = & \text{mín} & w^{\omega}[c_1\delta^{\omega} + c_2 x^{\omega} + q_1^{\omega}\gamma^{\omega} + q_2^{\omega} y^{\omega}] & \\
& s.t. & b_1 \leq A \begin{pmatrix} \delta^{\omega} \\ x^{\omega} \end{pmatrix} \leq b_2, & \\
& & h_1^{\omega} \leq T^{\omega} \begin{pmatrix} \delta^{\omega} \\ x^{\omega} \end{pmatrix} + W^{\omega} \begin{pmatrix} \gamma^{\omega} \\ y^{\omega} \end{pmatrix} \leq h_2^{\omega}, & (3.1) \\
& & x^{\omega}, y^{\omega} \geq 0, & \\
& & \delta^{\omega}, \gamma^{\omega} \in \{0,1\}. &
\end{array}
$$

Teniendo en cuenta los valores que deben tomar todas las variables enteras, se construye para cada escenario el árbol de Ramificación y Fijación (BF, de sus siglas en inglés, Branch & Fix) con todas las posibles bifurcaciones de las variables enteras a 0 ó a 1.

Para describir la aproximación BFC para resolver el problema (2.6), necesitamos unas definiciones previas. En primer lugar, $\mathcal{R}^{\omega}$ denota el árbol BF asociado al escenario ω, y $\mathcal{G}^{\omega}$ el conjunto de nodos activos en el árbol $\mathcal{R}^{\omega}$, $\omega \in \Omega$.

Definición 3.1 *Se dice que dos variables son variables comunes si para algún grupo de escenarios son variables con valores iguales según las restricciones de no anticipatividad.*

En nuestro modelo de optimización estocástica mixto 0-1 de dos etapas, las variables comunes son las variables de primera etapa, entre las que se encuentran las variables comunes enteras, δ, y las variables comunes continuas, x.

En las Figuras 3.1 hasta 3.4 recogemos una ilustración de ésta y las siguientes definiciones. Para todas estas figuras suponemos que hemos seleccionado $\hat{p} = 3$ racimos de escenarios, por lo que tenemos tres árboles BF. En cada uno de estos árboles contamos con tres variables enteras δ. En la Figura 3.1 vemos que la variable δ_2 toma el mismo valor, cero, en los tres árboles, por lo que a esta variable se le denomina variable común.

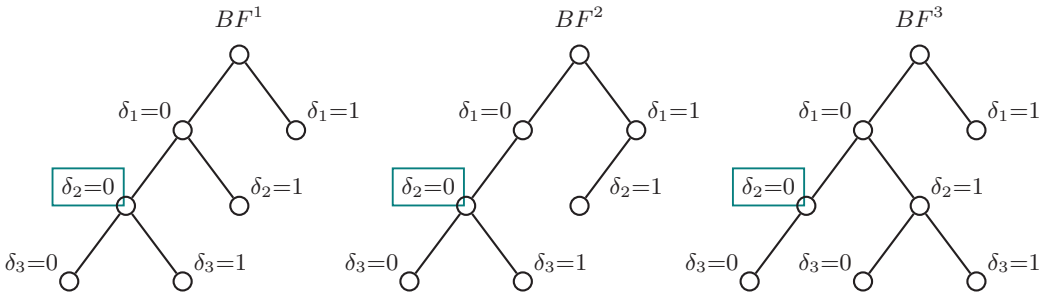

Figura 3.1: Variable común

Definición 3.2 *Dados dos nodos activos, $g \in \mathcal{G}^{\omega}$ y $g' \in \mathcal{G}^{\omega'}$, se dice que son nodos gemelos si cada uno de ellos es un nodo raíz o si los caminos desde el nodo raíz a cada uno de ellos en sus propios árboles BF, $\mathcal{R}^{\omega}$ y $\mathcal{R}^{\omega'}$, respectivamente, tienen la misma ramificación o han fijado los mismos valores para las variables δ^{ω} y $\delta^{\omega'}$, para $\omega, \omega' \in \Omega$.*

Hay que destacar que para satisfacer las restricciones de no anticipatividad, la ramificación y fijación en el valor 0 ó 1 debe ser la misma para las variables comunes enteras de nodos gemelos.

La Figura 3.2 ilustra esta definición. Vemos que el nodo resultande de ramificar δ_3 a cero es un nodo gemelo, ya que el camino seguido en cada uno de los árboles BF para llegar a él es el mismo. Sin embargo, vemos en la Figura 3.3 que el nodo resultante de bifurcar δ_3 a uno no es un nodo gemelo, ya que el camino seguido en el tercer árbol para llegar a este nodo no coincide con el camino seguido en los dos primeros árboles.

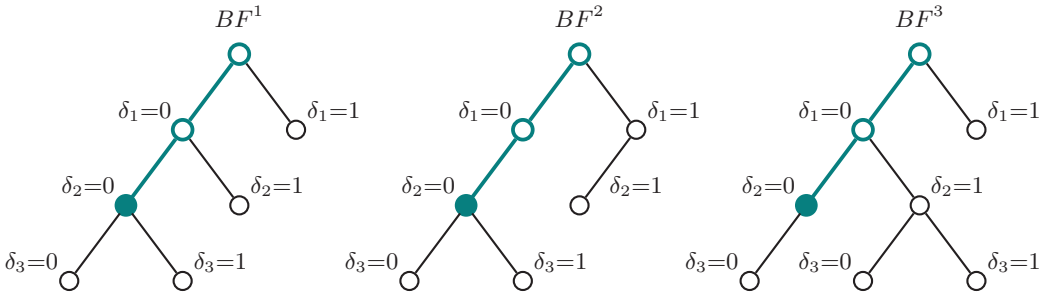

Figura 3.2: Nodos gemelos

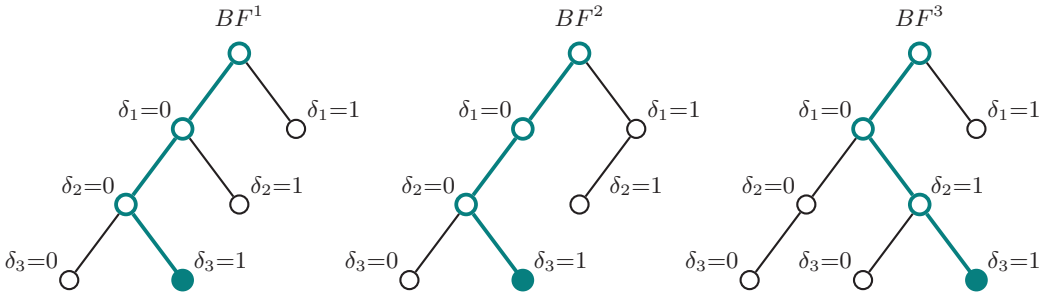

Figura 3.3: Nodos no gemelos

Definición 3.3 *Una familia de nodos gemelos (TNF), $\mathcal{H}_f$, es un conjunto de nodos tal que cada nodo es un nodo gemelo del resto de los nodos de la familia, para $f \in \mathcal{F}$, donde $\mathcal{F}$ es el conjunto de TNFs.*

Se observa que $g, g' \in \mathcal{H}_f$ para cualquier familia $f \in \mathcal{F}$ implica que $\omega \neq \omega'$ para $g \in \mathcal{G}^{\omega}$ y $g' \in \mathcal{G}^{\omega'}$, $\omega, \omega' \in \Omega$.

Definición 3.4 *Una TNF entera es un conjunto entero de nodos BF, uno por cada árbol BF, donde se satisfacen las condiciones de no anticipatividad (2.2) de las variables δ.*

En la Figura 3.4 vemos que una familia de nodos gemelos es cualquier rama de los árboles que está formada por nodos gemelos en estos árboles BF.

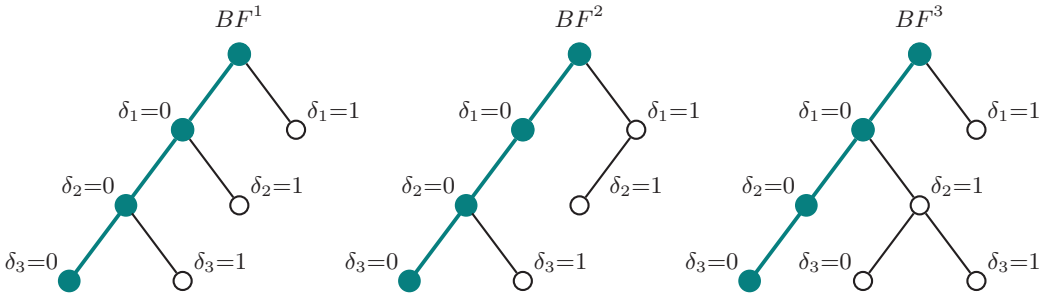

Figura 3.4: Familia de nodos gemelos

Cada TNF entera nos proporciona unos valores 0-1 de todas las variables δ. Si fijamos estos valores en el problema (2.5), y abusando ligeramente de la notación, llegamos al siguiente modelo mixto 0-1:

$$\begin{aligned}(MIP^{TNF}):\ z^{TNF} = c_1\overline{\delta} + \text{mín} \quad & c_2x + \sum_{\omega\in\Omega} w^\omega[q_1^\omega\gamma^\omega + q_2^\omega y^\omega] \\ s.t. \quad & b_1 \leq A\begin{pmatrix}\overline{\delta}\\ x\end{pmatrix} \leq b_2 \\ & h_1^\omega \leq T^\omega\begin{pmatrix}\overline{\delta}\\ x\end{pmatrix} + W^\omega\begin{pmatrix}\gamma^\omega\\ y^\omega\end{pmatrix} \leq h_2^\omega, \quad \omega\in\Omega \\ & x, y^\omega \geq 0, \gamma^\omega \in \{0,1\}, \quad \omega\in\Omega.\end{aligned} \tag{3.2}$$

donde $\overline{\delta}$ hace referencia a los valores fijados de las variables enteras en la TNF entera. De esta manera la parte de la función objetivo en la que intervienen las variables enteras de primera etapa resulta ser determinista, y además las δ dejan de ser variables a determinar ya que han sido fijadas con anterioridad.

Para una TNF entera consideramos un modelo mixto 0-1 particular para resolver. En este punto las varibles δ pueden tomar un valor real en el intervalo $[0,1]$, si dichas variables no han sido fijadas en la TNF que estamos considerando. Es decir, dejamos fijas aquellas variables enteras que hayan sido ramificadas y permitimos que el resto tomen valores en el intervalo $[0,1]$. Abusando de la notación, denotamos de nuevo como δ al subvector formado por las variables δ que no han sido fijadas en este punto. Este problema mixto 0-1 se puede escribir de la siguiente manera:

$$\begin{aligned}(MIP^f):\ z^f = K + \text{mín} \quad & c_1\delta + c_2x + \sum_{\omega\in\Omega} w^\omega[q_1^\omega\gamma^\omega + q_2^\omega y^\omega] \\ s.t. \quad & \overline{b}_1 \leq A\begin{pmatrix}\delta\\ x\end{pmatrix} \leq \overline{b}_2 \\ & \overline{h}_1^\omega \leq T^\omega\begin{pmatrix}\delta\\ x\end{pmatrix} + W^\omega\begin{pmatrix}\gamma^\omega\\ y^\omega\end{pmatrix} \leq \overline{h}_2^\omega, \quad \omega\in\Omega \\ & x \geq 0, 0 \leq \delta \leq 1 \\ & y^\omega \geq 0, \gamma^\omega \in \{0,1\}, \quad \omega\in\Omega,\end{aligned} \tag{3.3}$$

donde K es el valor de la función objetivo de las variables δ que ya han sido fijadas, y $\overline{b}_1$, $\overline{b}_2$, $\overline{h}_1$ y $\overline{h}_2$ son los vectores de los límites de las restricciones actualizados con

los valores fijados.

Está claro que la relajación de las condiciones de no anticipatividad (2.2) y (2.3) no es necesaria en cada par de escenarios. Para ganar eficiencia se propone la partición del árbol de escenarios en *clusters* o *racimos* donde el conjunto de escenarios Ω se divide en subconjuntos, Ω^1, Ω^2,..., $\Omega^{\hat{p}}$, donde $\hat{p}$ es el número de clusters considerados ($1 \leq \hat{p} \leq |\Omega|$).

El criterio para hacer racimos de escenarios, esto es, crear los subconjuntos $\Omega^1, \ldots, \Omega^{\hat{p}}$, puede basarse en criterios estadísticos como medir la desviación interna más pequeña de los parámetros de incertidumbre, la desviación más grande, etc. La determinación del mejor criterio de eficiencia dependerá en general del ejemplo que estemos considerando. En cualquier caso, hay que destacar que los racimos deben satisfacer las condiciones de una partición del conjunto de escenarios, esto es:

$$\Omega^p \cap \Omega^{p'} = \emptyset,\ p, p' = 1, \ldots, \hat{p} : p \neq p'$$

y

$$\Omega = \bigcup_{p=1}^{\hat{p}} \Omega^p.$$

Así, el problema a considerar bajo cada racimo de escenarios $p = 1, \ldots, \hat{p}$ se puede expresar en representación compacta de la sigueinte manera:

$$\begin{array}{rll} (MIP^p): \ z^p_{MIP} = \text{mín} & c_1\delta^p + c_2 x^p + \displaystyle\sum_{\omega \in \Omega^p} w^\omega [q_1^\omega \gamma^\omega + q_2^\omega y^\omega] & \\ s.a. & b_1 \leq A \begin{pmatrix} \delta^p \\ x^p \end{pmatrix} \leq b_2 & \\ & h_1^\omega \leq T^\omega \begin{pmatrix} \delta^p \\ x^p \end{pmatrix} + W^\omega \begin{pmatrix} \gamma^\omega \\ y^\omega \end{pmatrix} \leq h_2^\omega, \quad \forall \omega \in \Omega^p & \\ & x^p \geq 0, \delta^p \in \{0,1\}, & \\ & y^\omega \geq 0, \gamma^\omega \in \{0,1\}, \quad \forall \omega \in \Omega^p & \end{array} \tag{3.4}$$

Los $\hat{p}$ problemas (3.4) se relacionan mediante las siguientes restricciones de no anticipatividad:

$$\delta_i^p - \delta_i^{p'} = 0 \tag{3.5}$$

$$x_i^p - x_i^{p'} = 0, \tag{3.6}$$

$\forall i \in \{1, ..., n\}, p, p' = 1, \ldots, \hat{p} : p \neq p'$, obteniéndose la formulación del problema completo dada en (2.7).

Hay que notar que existen dos casos extremos en cuanto al número de clusters seleccionado. $\hat{p} = 1$ corresponde a considerar un único racimo, es decir, el modelo (3.4) coincide con el modelo inicial (2.5); y $\hat{p} = |\Omega|$ corresponde a considerar tantos clusters como escenarios, y por tanto el problema (3.4) coincide con el modelo (3.1).

Sobre el concepto de racimos de escenarios, las definiciones de la metodología BFC han de ser adaptadas y se enuncian de la siguiente manera:

Definición 3.5 *Dados dos nodos activos, $g \in \mathcal{G}^p$ y $g' \in \mathcal{G}^{p'}$, se dice que son nodos gemelos si cada uno de ellos es un nodo raíz o si los caminos desde el nodo raíz a cada uno de ellos en sus propios árboles BF, $\mathcal{R}^p$ y $\mathcal{R}^{p'}$, respectivamente, tienen la misma ramificación o han fijado los mismos valores para las variables δ^p y $\delta^{p'}$, para $1 \leq p,\ p \leq |\Omega|,\ p \neq p'$.*

Análogamente, se definen las TNF y TNF entera relativas a árboles asociados a clusters, a partir de las Definiciones 3.3 y 3.4, respectivamente.

Como veremos más adelante, la idea de considerar clusters es muy relevante en la mejora de la eficiencia computacional del algoritmo BFC, por lo que la elección del número de racimos de escenarios es decisiva a la hora de reducir el tiempo empleado por el algoritmo en encontrar la solución óptima.

3.2. Bases teóricas

El algoritmo BFC ramifica los árboles BF asociados a un escenario, o equivalentemente a un racimo de escenarios, sobre las variables 0-1, δ, de primera etapa a la vez que comprueba que se satisfacen las condiciones de no anticipatividad y de integrabilidad de las TNFs. Dicho de otra manera, el BFC ramifica los árboles BF sobre las variables δ resolviendo los submodelos mixtos 0-1 MIP bajo los racimos de escenarios.

Esta ramificación hace que el número de subproblemas a resolver sea elevado. A medida que el número de variables enteras de primera etapa aumenta, el número de nodos que tiene el árbol BF se hace más grande. En concreto el número total de nodos de cada árbol es 2^{n_δ}, donde n_δ es el número de variables enteras de primera etapa.

Recorrer todos los nodos del árbol implicaría la necesidad de mucho tiempo computacional; sin embargo, no es necesario resolverlos todos ya que existen cotas inferiores y superiores que permiten cortar la rama cuando se han alcanzado dichas cotas. A continuación, describimos la obtención de estas cotas y mencionamos sus principales propiedades. Para ver las demostraciones detalladas de estas relaciones acudir a [12].

Una primera cota inferior de la función objetivo del problema MIP es el valor de la función objetivo del problema lineal relajado. Es decir, si relajamos la integrabilidad de todas las variables enteras del modelo (2.5), tanto de primera como de segunda etapa, obtenemos un modelo estocástico lineal de dos etapas.

Al valor de la función objetivo de esta representación lineal lo denotamos por Z_{LP}, y claramente es una cota inferior del valor de la función objetivo del problema original (2.5). Es decir, se cumple que

$$Z_{LP} \leq z_{MIP}.$$

Adicionalmente, para cada nodo i en el árbol de enumeración BF definimos otra cota inferior, $\underline{Z}_i$, como

$$\underline{Z}_i = \sum_{p=1}^{\hat{p}} z_i^p, \ 1 \leq \hat{p} \leq |\Omega|$$

donde z_i^p es la solución del modelo MIP^p (3.4) en el que las i primeras variables δ se han fijado a 0 ó 1, es decir, es la solución del siguiente problema:

$$\begin{array}{lll} (MIP_i^p): \ MIP^p \ (3.4) & & \\ \quad s.a. & \delta_j^p = \overline{\delta}_j^p, & \forall j \in \{1,...,i\} \\ & \delta_j^p \in \{0,\ 1\}, & \forall j \in \{i+1,...,n\} \end{array} \tag{3.7}$$

El valor de esta cota inferior asociado al nodo raíz, $i = 0$, lo denotamos por $\underline{Z}_0$. En este caso ninguna de las variables δ han sido fijadas a un valor concreto previamente.

Al resolver la secuencia de problemas (3.7) para cada cluster o racimo p previamente a calcular la correspondiente cota $\underline{Z}_i$, las variables δ y/o x pueden satisfacer o no las condiciones de no anticipatividad (3.5) y (3.6), respectivamente. Si ambas variables satisfacen las condiciones de no anticipatividad se ha llegado a la solución óptima del problema. En el caso en el que las variables δ satisfagan la no anticipatividad, pero las x no, hay que resolver los dos problemas siguientes:

$$\begin{array}{lll} (MIP_i^{TNF}): \quad MIP^{TNF} & & \\ \quad s.a. & \delta_j = \overline{\delta}_j, & \forall j \in \{1,...,i\} \quad \text{(fijadas por la rama)} \\ & \delta_j = \overline{\delta}_j, & \forall j \in \{i+1,...,n\} \quad \text{(solución de } MIP_i^p\text{)} \end{array} \tag{3.8}$$

$$\begin{array}{lll} (MIP_i^f): \quad MIP^f & & \\ \quad s.a. & \delta_j = \overline{\delta}_j, & \forall j \in \{1,...,i\} \\ & \delta_j \in [0,1], & \forall j \in \{i+1,...,n\} \end{array} \tag{3.9}$$

A continuación detallamos las relaciones de desigualdad existentes entre las distintas soluciones y cotas planteadas. Éstas están definidas para un problema de minimización, para el caso de un problema de maximización la relaciones se cumplirán igualmente pero en sentido opuesto.

Proposición 3.1 *El valor de la solución del problema asociado al nodo raíz es una cota inferior del problema original. Es decir,*

$$\underline{Z}_0 \leq z_{MIP}.$$

Proposición 3.2 *El valor de la solución del problema lineal relajado es siempre menor que el valor de la solución del problema asociado al nodo raíz. Es decir,*

$$Z_{LP} \leq \underline{Z}_0.$$

De este modo, en vez de resolver la relajación lineal para obtener la cota inferior Z_{LP}, alternativamente podemos calcular

$$\underline{Z}_0 = \sum_{p=1}^{\hat{p}} z_0^p,$$

ya que es más cercana al valor óptimo del problema mixto 0-1.

Obviamente, calcular esta cota es computacionalmente más complicado que calcular el valor de la solución del problema lineal relajado. Sin embargo, el resultado global de la búsqueda de la solución óptima mejora, ya que el número de nodos BF que tiene que resolver es mucho menor.

Por otra parte, habrá que elegir el número de racimos en los que dividimos los escenarios. Cuanto mayor sea este número, $\hat{p}$, la cota inferior será peor, es decir, se alejará más del valor óptimo; pero el tiempo que tardará en encontrar esta cota será menor. Por lo tanto, hay que encontrar un equilibrio entre la bonanza de la cota y el esfuerzo que se necesita para obtenerla.

Proposición 3.3 *La cota inferior obtenida en el nodo i-ésimo del árbol BF es menor o igual que la cota inferior asociada a cualquiera de los subproblemas de sus nodos descendientes, debido a la ramificación de la variable δ $(i+1)$-ésima a 0 ó 1 para cualquier nodo $i \in \{0, 1, 2, \ldots, n\}$. Esto es,*

$$\underline{Z}_i \leq \underline{Z}_{i+1}.$$

Entonces, se satisface la siguiente cadena de desigualdades:

$$\underline{Z}_0 \leq \underline{Z}_1 \leq \underline{Z}_2 \leq \ldots \leq \underline{Z}_n.$$

Proposición 3.4 *Para cualquier nodo i del árbol BF se cumple que*

1. $\underline{Z}_i \leq z_i^{TNF}$
2. $z_i^f \leq z_i^{TNF}$ $\forall i \in \{1, 2, \ldots, n\}$.

Hay que destacar que la solución del problema (3.2) para una TNF entera determinada puede coincidir con la solución óptima del problema siempre y cuando las variables δ y x cumplan las condiciones de no anticipatividad y las variables δ sean

enteras. Sin embargo, esto no implica necesariamente que se deba cortar la rama, excepto si todas las variables han sido ramificadas en la familia. Por el contrario, se puede encontrar una solución mejor ramificando en las variables 0-1 que aún no han sido ramificadas, como asegura el apartado 2 de la Proposición 3.4.

z^{TNF} es el valor de la solución del problema (3.2), que satisface las condiciones de no anticipatividad (2.3) cuando han sido ramificadas las variables δ a sus valores 0-1 (por lo que la restricción de no anticipatividad (2.2) ya se satisface). La rama se puede cortar si $z^{TNF} = z^f$, donde z^f es el valor de la solución del modelo (3.3), donde las condiciones (2.2) y (2.3) se satisfacen, pero las variables δ no ramificadas pueden tomar valores reales en el intervalo [0,1]. En este caso, no existe una solución mejor que z^{TNF} para ningún nodo descendiente en la TNF entera.

Si $z_i^{TNF} > z_i^f$ y $z_i^f < \overline{Z}$, debemos seguir ramificando hacia el nodo $(i+1)$-ésimo, porque es posible encontrar una solución mejor en ese árbol.

Proposición 3.5 *Sea $z_{i_1}^f$ el valor de la solución del modelo $MIP_{i_1}^f$ para el nodo BFC $i_1 \in \{1, 2, \ldots, n-1\}$ en cualquier rama dada. Si seguimos ramificando en la misma rama, obtenemos que*

$$z_{i_1}^f \leq z_{i_2}^f$$

donde $i_2 > i_1$ es un nodo descendiente de i_1.

Proposición 3.6 (Factibilidad I) *Si los nodos descendientes de una rama dada no son infactibles, entonces se podrá encontrar una solución del problema original. Denotamos como $\overline{Z}$ al valor de la solución asociada.*

Proposición 3.7 *Una vez que una rama se ha cortado no es posible encontrar una mejor solución en los nodos descendientes (en el caso en el que se pueda seguir ramificando).*

Proposición 3.8 (Factibilidad II) *Una rama de los árboles BF tiene una solución factible, al menos, si el problema original es factible.*

Proposición 3.9 (Optimalidad) *La solución factible del problema original (2.5) con el menor valor de la función objetivo del conjunto de soluciones*

$$\begin{array}{l} (\delta, x, (\gamma^\omega)_{\omega\in\Omega}, (y^\omega)_{\omega\in\Omega}) = \\ \quad (\overline{\delta}, \overline{x}, (\overline{\gamma}^\omega)_{\omega\in\Omega}, (\overline{y}^\omega)_{\omega\in\Omega})_j \quad \in \quad \{0,1\}^{n_\delta} \times I\!R^{n_x} \times \{0,1\}^{n_\gamma|\Omega|} \times I\!R^{n_y|\Omega|} \end{array}$$

es la solución óptima del problema.

Finalmente, de todas estas proposiciones se deduce que la aproximación propuesta siempre encuentra una solución óptima o determina que el problema es infactible.

Sin embargo, su computación para ejemplos de grandes dimensiones no es trivial incluso con los ordenadores que existen hoy en día, ya que el número de ramas que se evalúa puede llegar a ser muy grande. Es decir, el cardinal del conjunto de soluciones factibles puede ser demasiado grande y puede existir un número alto de modelos mixtos enteros MIP^p, MIP^{TNF} y MIP^f para resolver.

En concreto, el algoritmo tardará más cuanto mayor sea el número de problemas MIP^{TNF} y MIP^f que tiene que resolver. Estos problemas no han sido rotos en clusters de modo que siguen siendo de unas dimensiones similares al problema original, salvo porque algunas variables δ han sido fijadas. Por lo tanto, se necesitará un gran esfuerzo para resolverlos.

La eficiencia de cortar ramas es más destacada para el caso de problema de grandes dimensiones. Este hecho se deduce de las Proposiciones 3.3 a 3.5 y especialmente de la Proposición 3.7.

3.3. Algoritmo

En esta sección proponemos el algoritmo completo BFC.

Recordamos la siguiente notación:

- $\mathcal{R}^p$: árbol BF para el cluster p, para $p = 1, \ldots, \hat{p}$.
- MIP^p: modelo entero mixto 0-1 (3.7) asociado al cluster para un nodo dado del árbol BF $\mathcal{R}^p$ en la TNF dada.
- z^p: valor de la solución del modelo mixto 0-1 MIP^p.
- $\underline{Z}$: cota inferior del valor de la solución del problema original (2.5) que será obtenida del mejor conjunto entero TNF descendiente de una familia dada. Este valor se calcula como $\underline{Z} = \sum_{p=1}^{\hat{p}} z^p$, excepto para los nodos raíz de los árboles BF, que es ese caso es $\underline{Z}_0 = \sum_{p=1}^{\hat{p}} z^p$.
- $\overline{Z}$: cota superior del valor de la solución del problema original (2.5).

Por convenio, para el modelo (3.8) infactible asociado a una TNF entera, consideramos $z^{TNF} = +\infty$; para el modelo (3.9) infactible, consideramos $z^f = +\infty$, y para el modelo (3.4) infactible, tomamos $z^p = +\infty$, para cada cluster o racimo, p, donde $p = 1, \ldots, \hat{p}$.

Aunque en la versión del algoritmo tomada de [12] la cota superior es inicializada a ∞ en el Paso 0, en nuestra propuesta hemos inicializado el valor de esta cota al obtenido en una solución cuasi-óptima del problema con un porcentaje ρ de tolerancia de optimalidad prefijado, $\overline{Z}(\rho)$.

El procedimiento BFC se puede representar mediante la siguiente secuencia de pasos:

Paso 0: Inicializar $\overline{Z} := \overline{Z}(\rho)$ e $i := 0$.

Paso 1: Resolver los problemas mixtos 0-1 bajo cada cluster, MIP^p (3.4), $\forall p = 1, \ldots, \hat{p}$, y calcular $\underline{Z}_0 = \sum_{p=1}^{\hat{p}} z_0^p$. Si algún problema es infactible, entonces parar (el problema original es infactible).

Si existe alguna variable δ (0-1) que no satisface (3.5), ir al Paso 2.

Si existe alguna variable continua x que no satisface (3.6), ir al Paso 6.

En otro caso, $\overline{Z} := \underline{Z}_0$ y parar (se ha encontrado la solución óptima del problema original).

Paso 2: Inicializar $i := 1$ e ir al Paso 4.

Paso 3: Actualizar $i := i + 1$. Si $i = n + 1$ entonces ir al Paso 8.

Paso 4: Ramificar $\delta_i := 0$, $\forall p = 1, \ldots, \hat{p}$.

Paso 5: Resolver MIP_i^p (3.7), $\forall p = 1, \ldots, \hat{p}$ y calcular $\underline{Z} = \sum_{p=1}^{\hat{p}} z_i^p$. Si algún problema es infactible, ir al Paso 7.

Si $\underline{Z} \geq \overline{Z}$ entonces ir al Paso 7.

Si existe alguna variable δ que no satisface (3.5) entonces ir al Paso 3.

Si todas las variables x satisfacen (3.6) entonces actualizar $\overline{Z} := \underline{Z}$ e ir al Paso 7.

Paso 6: Resolver MIP_i^{TNF} (3.8) para que las variables x satisfagan (3.6). Notar que el valor de la solución se denota por z^{TNF}. Si el problema es infactible e $i = 0$, parar (el problema original es infactible); si $i > 0$, ir al Paso 7.

Actualizar $\overline{Z} := \text{mín}\{z^{TNF}, \overline{Z}\}$.

Si $i = n$ entonces ir al Paso 7.

Resolver MIP_i^f (3.9). El valor de la solución en este caso se denota por z^f. Si el problema es infactible e $i = 0$, parar (el problema original es infactible); si $i > 0$, ir al Paso 7.

Si $z^{TNF} = z^f$ o $z^f \geq \overline{Z}$, entonces ir al Paso 7.

Si $z^f < \overline{Z}$ y todas las variables δ cumplen (3.5), actualizar $\overline{Z} := z^f$ e ir al Paso 7. En otro caso ir al Paso 3.

Paso 7: Cortar la rama. Si $\delta_i^p = 0$, $\forall p = 1, \ldots, \hat{p}$, entonces ir al Paso 10.

Paso 8: Nodo previo. Actualizar $i := i-1$. Si $i = 0$ entonces parar (se ha encontrado la solución óptima del problema original $\overline{Z}$).

Paso 9: Si $\delta_i^p = 1$, $\forall p = 1, \ldots, \hat{p}$, entonces ir al Paso 8.

Paso 10: Ramificar $\delta_i^p := 1$, $\forall p = 1, \ldots, \hat{p}$. Ir al Paso 5.

Capítulo 4

Experiencia computacional

4.1. Casos de estudio

Para evaluar el algoritmo propuesto utilizamos un conjunto de casos generados con la estructura del problema general (2.5). En concreto, los ejemplos están tomados de [31] y su estructura en representación compacta está basada en el problema (38) de [30].

La ocurrencia de cada escenario es equiprobable bajo cada escenario, es decir,

$$w^{\omega} = \frac{1}{|\Omega|},\ \forall \omega \in \Omega$$

donde Ω es el conjunto de escenarios.

El procedimiento utilizado para generar los distintos problemas es el siguiente. Se toman vectores aleatorios para los coeficientes de la función objetivo, c_1, c_2, q_1^{ω}, q_2^{ω}, usando la distribución uniforme sobre los intervalos [-2.5,-1.5], [-2.5,-1.5], [$-30+\frac{\omega}{|\Omega|}$,$-10+\frac{\omega}{|\Omega|}$] y [$-30+\frac{\omega}{|\Omega|}$,$-10+\frac{\omega}{|\Omega|}$], respectivamente. Los límites inferiores de las restricciones de primera y segunda etapa, b_1 y h_1^{ω}, se fijan a $\frac{1}{2}\cdot k$ y $\frac{1}{2}\cdot k+\frac{\omega}{|\Omega|}$, respectivamente. Y los límites superiores, b_2 y h_2^{ω}, se generan a partir de distribuciones uniformes sobre los intervalos $[k_1, k_1 + k\cdot(n_\delta + n_x)]$ y $[k_2+\frac{\omega}{|\Omega|}, k_2+\frac{\omega}{|\Omega|}+k\cdot(n_\delta+n_x+n_\gamma+n_y)]$, siendo $k \in [0,1]$, $k_1 \in [0,4]$ y $k_2 \in [0,3]$. Por último, la matriz de coeficientes de primera etapa, A, y las matrices de coeficientes de segunda etapa, T^{ω} y W^{ω}, se generan con distribuciones uniformes sobre [0,2], $[k'\cdot\frac{\omega}{|\Omega|}, k'\cdot\frac{\omega}{|\Omega|}+0.3]$ y $[k''\cdot\frac{\omega}{|\Omega|}, k''\cdot\frac{\omega}{|\Omega|}+8]$, respectivamente, donde $k' \in [-0.1,0]$ y $k'' \in [0,1.5]$.

En la Tabla 4.1 se recogen las dimensiones de los seis problemas estudiados, tanto en representación compacta como en representación en variables divididas. La notación utilizada es la siguiente: para la representación compacta, m^c es el número de restricciones, n_δ^c y n_x^c denotan el número de variables enteras y continuas de pri-

mera etapa, respectivamente, n^c_{el} representa el número de coeficientes no nulos en la matriz de restricciones, y $dens^c$ es la densidad (en %) de la matriz de restricciones. n_γ y n_y son el número de variables enteras y continuas de segunda etapa, respectivamente. De la misma forma se definen m^e, n^e_δ, n^e_x, n^e_{el} y $dens^e$ para la representación extendida o en variables divididas. Finalmente $|\Omega|$ denota el número de escenarios.

Tabla 4.1: Dimensión de los modelos

	Representación compacta							Representación extendida					
Caso	m^c	n^c_δ	n^c_x	n_γ	n_y	n^c_{el}	$dens^c$	m^e	n^e_δ	n^e_x	n^e_{el}	$dens^e$	$\|\Omega\|$
P1	330	30	15	320	64	18590	13.13	2080	960	480	31680	0.84	32
P2	148	10	10	128	128	3984	9.75	1408	320	320	17664	1.40	32
P3	288	5	10	280	420	70120	34.05	4410	350	700	80500	1.04	70
P4	1290	30	15	1280	256	73410	3.59	8320	3840	1920	142080	0.23	128
P5	1935	25	10	2560	1920	134925	1.54	8320	3200	1280	210560	0.28	128
P6	2010	20	20	2000	2000	120400	1.48	12000	4000	4000	216000	0.15	200

4.2. Caso ilustrativo

En esta sección presentamos un ejemplo de pequeñas dimensiones para ilustrar el procedimiento del algoritmo BFC. Para ello seleccionamos una variación del problema P3 e implementamos el esquema con el optimizador CPLEX.

El problema P3 presentado en la Tabla 4.1 tiene $|\Omega| = 70$ escenarios, pero cada uno de ellos está compuesto internamente por 10 escenarios. Para simplificar el ejemplo y así facilitar la exposición de los pasos que ha de realizar el algoritmo, tomamos este mismo problema pero en el que cada uno de los 70 escenarios está compuesto por un único escenario. Elegimos una descomposición en $\hat{p} = 7$ clusters o racimos de escenarios. En la Figura 4.1 se resumen los pasos del algoritmo.

Como hemos comentado, en este trabajo presentamos una modificación sobre el esquema BFC desarrollado por [12]. Nuestro algoritmo inicializa la cota superior inicial del valor de la función objetivo a la solución cuasi-óptima encontrada por el optimizador con un porcetaje de tolerancia de optimalidad. Sin embargo, para la ilustración de este ejemplo vamos a tomar la versión tradicional del algoritmo de [12] en la que la cota inicial es $\overline{Z} = \infty$. Elegimos esta inicialización ya que si calculamos la solución cuasi-óptima proporcionada por el optimizador obtendríamos la solución óptima debido a que estamos tratando con un problema de pequeñas dimensiones y no es ningún reto para ningún optimizador.

En el nodo inicial resolvemos la sucesión de problemas MIP^p (3.4), $\forall p = 1, ..., 7$, y calculamos la cota inferior inicial, $\underline{Z}_0 = \sum_{p=1}^{7} z^p = -58.8856$. Como las variables δ cumplen las condiciones de no anticipatividad (3.5), pero las x no cumplen (3.6), resolvemos el correspondiente modelo mixto 0-1 MIP^{TNF} (3.8) fijando las variables δ a la solución obtenida en los MIP^p y calculamos $z^{TNF} = -58.8493$. Observamos

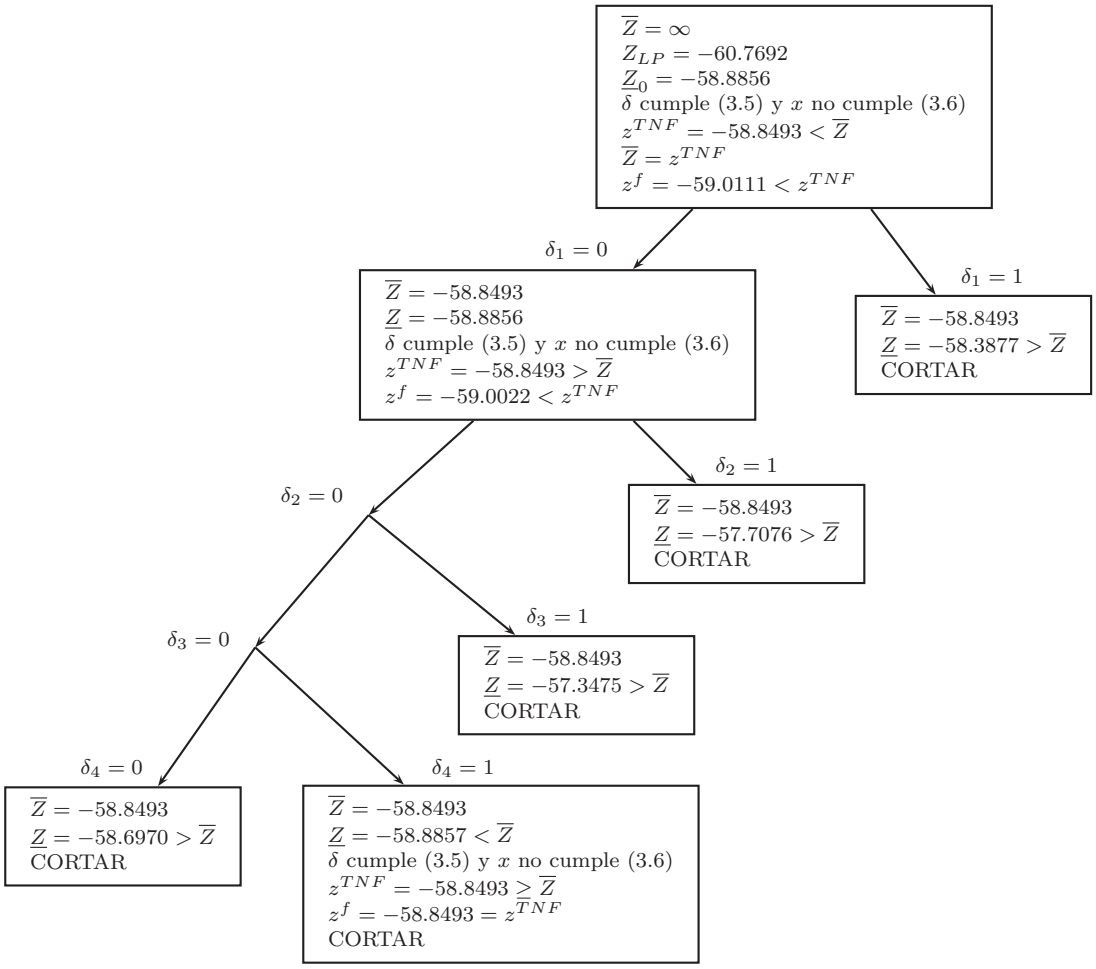

Figura 4.1: Ejemplo ilustrativo

que $z^{TNF} < \overline{Z}$, por lo que actualizamos la cota superior, $\overline{Z} = z^{TNF}$. Resolvemos el problema MIP^f (3.9), donde las variables δ son libres (es decir, pueden tomar valores en el intervalo [0,1]) y resulta $z^f = -59.0111 < z^{TNF}$, por lo que podríamos encontrar una solución menor a la cota superior en esta rama, de modo que no la cortamos y empezamos a ramificar.

Comenzamos ramificando $\delta_1^p = 0$, $\forall p = 1, ..., 7$. En este nodo resolvemos los problemas MIP^p (3.7), $\forall p$, correspondientes obteniendo una nueva cota inferior, $\underline{Z} = -58.8857$. Las variables de primera etapa de esta solución cumplen las condiciones (3.5) pero no (3.6), por lo que resolvemos el correspondiente MIP^{TNF}, $z^{TNF} = -58.8493$. Como esta solución no es inferior a la cota superior, no actualizamos el valor de esta última. En este punto aún no podemos cortar la rama hasta asegurarnos de que no podemos encontrar ninguna solución mejor en ella, por lo que resolvemos el MIP^f y obtenemos $z^f = -59.0022 < z^{TNF}$, es decir, podría haber una solución mejor a lo largo de la rama. Las δ de esta solución no cumplen las condiciones de no anticipatividad, pero $\delta_2^p = \delta_3^p = 0$, $\forall p$, por lo que fijamos estos dos valores, y pasamos al siguiente nodo.

Notar que la solución obtenida al resolver el problema MIP^f nos permite fijar los valores de δ_2 y δ_3 a cero, ya que si resolviésemos los problemas correspondientes en cada uno de estos nodos obtendríamos el mismo valor de la solución y de la función objetivo. De esta manera, las cotas no se van a actualizar, por lo que aprovechamos la solución para evitar resolver problemas MIP^p, $\forall p$, MIP^{TNF} y MIP^f.

Como δ_2 y δ_3 han sido fijadas, ramificamos $\delta_4^p = 0$, $\forall p$, y calculamos $\underline{Z} = -58.6970$ resolviendo los MIP^p. Como $\underline{Z} > \overline{Z}$, no podemos encontrar ninguna solución mejor por esta rama, ya que la cota inferior es más grande que la cota superior, así que cortamos la rama y ramificamos $\delta_4 = 1$. Calculamos la cota inferior correspondiente obteniendo $\underline{Z} = -58.8857$. Como este valor es inferior a la cota superior, no cortamos la rama. Esta solución cumple las condiciones de no anticipatividad (3.5) pero no (3.6), por lo que resolvemos el problema MIP^{TNF} y obtenemos $z^{TNF} = -58.8493 \geq \overline{Z}$. Resolvemos el problema MIP^f, $z^f = -58.8493$, que coincide con z^{TNF}, de modo que cortamos la rama, ya que no es posible encontrar un valor inferior de la función objetivo en ella.

Vamos al nodo anterior y ramificamos $\delta_3^p = 1$, $\forall p$. En esta ocasión la cota inferior resulta ser $\underline{Z} = -57.3475 > \overline{Z}$, por lo que también cortamos esta rama. Retrocedemos al nodo anterior ramificando $\delta_2^p = 1$, $\forall p$, y obtenemos la cota inferior $\underline{Z} = -57{,}7076$, que es mayor que la cota superior $\overline{Z} = -58.8493$, por lo que cortamos esta rama.

Por último, volvemos al nodo inicial para ramificar $\delta_1^p = 1$, $\forall p$. En este nodo, la resolución de los subproblemas MIP^p (3.7), $\forall p = 1, ..., 7$, nos da una cota inferior igual a $\underline{Z} = -58.3877 > \overline{Z}$, de modo que también debemos cortar esta última rama.

Una vez recorridas todas las ramas posibles del árbol BF y habiendo ramificado todas las variables enteras de primera etapa posibles, el algoritmo BFC se para. Debemos notar que gracias a las relaciones de desigualdad que se cumplen entre los valores de la función objetivo de los distintos problemas MIP^p, MIP^{TNF} y MIP^f, podemos cortar muchos caminos sin necesidad de evaluar todos los nodos del árbol.

Una vez terminado el proceso, la descomposición BFC ha llegado a la solución óptima $Z_{MIP} = \overline{Z} = -58.8493$. Esta solución óptima está dada por $\delta = (0, 0, 0, 1, 1)$.

4.3. Resultados numéricos

La finalidad práctica de este trabajo es comprobar la eficiencia del algoritmo BFC frente al uso de optimizadores como CPLEX o COIN-OR sobre el modelo completo (2.5). Otro de los objetivos consiste en la mejora en el tiempo de ejecución utilizando CPLEX en lugar de COIN-OR. Para ello, en esta sección se presentan los resultados más relevantes de la aplicación de la descomposición BFC bajo COIN-OR y bajo CPLEX sobre el conjunto de casos de estudio planteados en la Tabla

4.1.

La resolución de los problemas se ha realizado con un ordenador con procesador Intel®Core^{TM}i7-2630QM bajo el sistema operativo Windows, con una velocidad de CPU de 2,00 GHz y 6Gb de memoria RAM.

En la Tabla 4.2 podemos ver los resultados para la resolución de cada uno de los problemas descritos en representación compacta, usando tanto COIN-OR como CPLEX. Z_{LP} y Z_{MIP} son los valores óptimos de la función objetivo del problema lineal relajado y del problema mixto 0-1, respectivamente. En algunos casos en los que no se obtiene la solución óptima dentro del límite de tiempo establecido (3 horas), Z_{MIP} denota la solución incumbente del problema proporcionada por el optimizador CPLEX. T_{LP}^{COIN} y T_{MIP}^{COIN} recogen el tiempo (en segundos) necesario para llegar a la solución óptima correspondiente usando COIN-OR. De la misma manera T_{LP}^{CPLEX} y T_{MIP}^{CPLEX} recogen el tiempo (en segundos) necesario para llegar a la solución óptima del problema utilizando el optimizador CPLEX sobre el problema completo.

Tabla 4.2: Resultados para la representación compacta (COIN-OR y CPLEX)

			COIN-OR				CPLEX			
Caso	Z_{LP}	Z_{MIP}	T_{LP}^{COIN}	T_{MIP}^{COIN}	$\overline{Z}^{COIN}(\rho)$	$T_{\overline{Z}(\rho)}^{COIN}$	T_{LP}^{CPLEX}	T_{MIP}^{CPLEX}	$\overline{Z}^{CPLEX}(\rho)$	$T_{\overline{Z}(\rho)}^{CPLEX}$
P1	-62.1504	-57.0077	0.062	16650.1	-56.8833(5)	38.6	0.021	3	-57.0077(0.5)	3
P2	-100.423	-99.8996	0.024	1.833	-99.3327(1)	0.25	0.015	2	-99.6225(1)	0
P3	-61.4023	-59.4645*	0.547	-	-58.6387(10)	70	0.286	-	-59.46(0.1)	28
P4	-76.0569	-70.7906*	0.29	-	-68.5212(10)	24	0.109	-	-70.7906(1)	40
P5	-86.706	-84.2161*	0.632	-	-82.3986(5)	20	0.217	-	-84.2065(0.5)	79
P6	-69.3048	-66.0478*	0.692	-	-65.955(5)	125	0.28	-	-66.0315(0.5)	49

-: Se ha excedido el tiempo límite (3 horas = 10800 segundos)
*: Solución incumbente en el tiempo límite

La mayoría de los software de optimización tienen rutinas para encontrar una cota superior de los problemas a resolver. Estas cotas son soluciones cuasi-óptimas que se obtienen con una tolerancia de optimalidad ρ, y tanto COIN-OR como CPLEX las proporciona. En las dos últimas columnas del cuadro correspondiente a cada optimizador de la Tabla 4.2 recogemos las distintas soluciones cuasi-óptimas dadas por ambos optimizadores para cada uno de los problemas planteados, representadas por $\bar{Z}^{COIN}(\rho)$ y $\bar{Z}^{CPLEX}(\rho)$, donde la tolerancia de optimalidad ρ viene dada en porcentaje (%). Además, se incluyen los tiempos (en segundos) necesarios para obtener estas soluciones, $T_{\overline{Z}(\rho)}^{COIN}$ y $T_{\overline{Z}(\rho)}^{CPLEX}$, respectivamente.

La tolerancia de optimalidad no está elegida al azar. Hay que hacer un pequeño estudio para encontrar un equilibrio entre la solución cuasi-óptima conseguida y el tiempo requerido para ello. Hay ocasiones en las que para conseguir mejorar ligeramente la cota superior se necesita aumentar el tiempo de ejecución lo suficiente como para no merecer la pena.

En la Tabla 4.2 se observa claramente que los resultados obtenidos mediante CPLEX para resolver el modelo en representación compacta mejoran los obtenidos con COIN-OR. Para los problemas más grandes, P3, P4, P5 y P6, ningún optimizador es capaz de llegar a la solución óptima sin exceder el tiempo límite (3 horas); en el caso de CPLEX es debido a una exceso en el límite de la memoria del orde-

nador, mientras que con COIN-OR se trata de un problema de tiempo, es decir, el ordenador tiene capacidad para guardar el problema, pero el tiempo de resolución es muy elevado. En los casos más pequeños, P1 y P2, bajo CPLEX siempre se llega al óptimo en un tiempo razonable. Por el contrario, el tiempo requerido por COIN-OR es muy superior, llegando en el caso del problema P1 a obtener el óptimo excediendo el tiempo máximo de tres horas, mientras que con CPLEX tan solo son necesarios 3 segundos.

Por su parte, también encontramos indicios de superioridad de CPLEX sobre COIN-OR si comparamos las soluciones cuasi-óptimas proporcionadas por cada uno de los optimizadores. En general, utilizando COIN-OR se obtienen peores soluciones y se requiere más tiempo que usando CPLEX. Además, con este último optimizador la tolerancia de optimalidad ρ con la que se obtienen las soluciones es más baja, es decir, se llega a soluciones más cercanas a la óptima, requiriendo también menos tiempo para conseguirlas.

Los problemas lineales relajados son modelos muy pequeños y no suponen un reto para ningún optimizador en general. En este caso, vemos que tanto usando COIN-OR como CPLEX, se obtiene la solución lineal óptima en poco tiempo, aunque de nuevo CPLEX mejora el tiempo de COIN-OR.

Esta diferencia en tiempos de ejecución, se incrementa cuando implementamos el algoritmo BFC para cada uno de los ejemplos. En la Tabla 4.3 se puede ver esta afirmación, ya que presentamos los resultados del BFC usando los optimizadores COIN-OR y CPLEX. Las cotas superiores utilizadas en cada caso son las obtenidas por CPLEX que recogemos en la Tabla 4.2. La notación utilizada en esta tabla es la siguiente. Z_{MIP} denota la solución óptima del problema y T_{MIP}^{COIN} y T_{MIP}^{CPLEX} indican el tiempo (en segundos) que ha tardado el optimizador COIN-OR y CPLEX, respectivamente, en encontrar la solución. n^{TNF} y n^f denotan el número de problemas MIP^{TNF} y MIP^f que se resuelven, respectivamente, y también se recoge el número de nodos del árbol BF que se evalúan. Finalmente, $\hat{p}$ denota el número de clusters o racimos de escenarios en los que se ha dividido el modelo en cada caso.

En esta tabla presentamos los resultados de los casos P1 y P2. Escogemos estos dos ejemplos por ser los más sencillos, ya que el optimizador COIN-OR necesita un tiempo muy superior a CPLEX para obtener el mismo o peor resultado. Para casos de mayores dimensiones, las diferencias entre ambos son aún más notables.

En la Tabla 4.3 observamos que el problema P2 se resuelve muy rápido bajo ambos optimizadores, aunque CPLEX sigue mejorando los tiempos requeridos por COIN-OR para todas las elecciones del número de clusters, $\hat{p}$. Obviamente el número de nodos que examina y el número de problemas MIP^{TNF} y MIP^f que resuelve coincide para ambos optimizadores, ya que el algoritmo es el mismo en ambos casos.

Para el problema P1 vemos más diferencias entre los dos optimizadores. Usando CPLEX siempre se llega a la solución óptima, aunque la mejor elección del número

Tabla 4.3: Resultados BFC (con cota superior) bajo COIN-OR y CPLEX

			COIN-OR				CPLEX			
Caso	$\hat{p}$	Z_{MIP}	n^{TNF}	n^f	nodos	T_{MIP}^{COIN}	n^{TNF}	n^f	nodos	T_{MIP}^{CPLEX}
P1	2	-57.0077	2	2	2	-	10	11	55	87
	4	-57.0077	2	3	104	9279	2	3	104	41
	8	-57.0077	3	4	154	-	11	12	447	162
	16	-57.0077	2	3	250	-	26	27	817	559
	32	-57.0077	2	2	158	-	62	60	1466	3289
P2	2	-99.8996	0	1	4	1.747	0	1	4	0
	4	-99.8996	4	5	14	11.435	4	5	14	1
	8	-99.8996	4	5	14	9.986	4	5	14	1
	16	-99.8996	3	4	18	8.136	3	4	18	2
	32	-99.8996	3	4	18	8.418	3	4	18	3

-: Se ha excedido el tiempo límite (3 horas = 10800 segundos)

de racimos es $\hat{p} = 4$, ya que en este caso requiere menos tiempo y resuelve menos MIP^{TNF} y MIP^f para obtener el óptimo. Por otra parte, utilizando COIN-OR también es mejor esta elección de $\hat{p}$, porque es el único caso en el que llega a la solución óptima dentro del tiempo límite (3 horas). Para el resto de elecciones de $\hat{p}$ no consigue obtener la solución óptima, y además examina menos nodos y resuelve menos MIP^{TNF} y MIP^f que CPLEX, llegando este último al óptimo. En este problema la solución cuasi-óptima utilizada como cota superior inicial coincide con el óptimo del problema, y aunque ésta es una situación favorable para el algoritmo BFC, COIN-OR no logra llegar a la solución óptima dentro del tiempo límite en la mayoría de los casos.

El esquema BFC que hemos presentado en el Capítulo 3 siempre llega a la solución óptima, aunque en ocasiones el tiempo necesario para ello sea elevado. Con el fin de reducir el tiempo de resolución, hemos introducido una cota superior en el nodo inicial, como ya hemos descrito.

En la Tabla 4.4 recogemos los resultados del algoritmo aplicado sobre dos de los seis casos de estudio (P1 y P2). Presentamos los resultados utilizando el optimizador CPLEX. En la tabla, además de la solución óptima, Z_{MIP}, y el tiempo, T_{MIP}^{CPLEX}, (en segundos) necesario para determinarla, presentamos el número de problemas MIP^{TNF}, n^{TNF}; el número de problemas MIP^f, n^f, y el número de nodos del árbol BF que examina. Estos resultados son importantes para entender por qué hay diferencias en los tiempos de ejecución. Cuantos menos problemas MIP^{TNF} y MIP^f resuelva, menor será el tiempo requerido para llegar al óptimo.

En esta tabla podemos ver que para los problemas pequeños, como son en este caso P1 y P2, el algoritmo BFC llega a la solución óptima, que obviamente coincide con la obtenida al resolver el problema en representación compacta, aún cuando la cota superior no la hemos fijado a la solución cuasi-óptima. Sin embargo, no mejora el tiempo de ejecución con respecto al completo. En el caso del problema P2 el tiempo que tarda bajo CPLEX es similar resolviendo el compacto y utilizando la descomposición BFC, independientemente del número de racimos de escenarios se-

Tabla 4.4: Resultados BFC sin y con cota superior bajo CPLEX

			Sin cota superior				Con cota superior			
Caso	$\hat{p}$	Z_{MIP}	n^{TNF}	n^f	nodos	T_{MIP}^{CPLEX}	n^{TNF}	n^f	nodos	T_{MIP}^{CPLEX}
P1	2	-57.0077	10	11	55	87	10	11	55	87
	4	-57.0077	26	27	290	134	2	3	104	41
	8	-57.0077	50	51	877	380	11	12	447	162
	16	-57.0077	90	91	1315	1124	26	27	817	559
	32	-57.0077	149	144	2187	5985	62	60	1466	3289
P2	2	-99.8996	0	1	4	0	0	1	4	0
	4	-99.8996	4	5	14	1	4	5	14	1
	8	-99.8996	4	5	14	1	4	5	14	1
	16	-99.8996	3	4	18	2	3	4	18	2
	32	-99.8996	3	4	18	4	3	4	18	3

leccionado. Sin embargo, para esta última técnica se demora algo más ya que realiza alguna comprobación adicional para asegurarse de que efectivamente ha llegado al óptimo. En el problema P1 se observa que a medida que el número de clusters aumenta, el tiempo necesario para la resolución también crece. Sin embargo, podemos ver que la introducción de una cota superior inicial hace que la resolución sea más rápida. Para ambos problemas podríamos decir que el número de clusters óptimo es $\hat{p} = 4$, ya que es cuando el tiempo de ejecución es menor.

En esta tabla, además, se puede apreciar la importancia de cortar las ramas del árbol BF. Se observa que no necesitan recorrer todo el árbol para llegar a la solución óptima. El problema P1 tiene 2^{30} (del orden de 10^9) nodos y en el peor caso, con $\overline{Z} = +\infty$ tan solo examina 5985. El problema P2 presenta un árbol BF mucho más pequeño, con $2^{10} = 1024$ nodos y como mucho en 18 nodos encuentra el óptimo.

Como hemos visto, la descomposición BFC llega siempre a la solución, ya que se detiene sólo cuando ha recorrido todo el árbol BF y ha ramificado todas las variables enteras de primera etapa. Sin embargo, a veces tiene que recorrer muchos nodos de este árbol para obtenerla. En general, a medida que va recorriendo el árbol BF y encuentra soluciones factibles, actualiza la cota superior. Si en el primer paso inicializamos la cota superior a $\overline{Z} := +\infty$, hasta que encuentra una cota superior aceptable, en ocasiones tiene que evaluar muchos nodos y además resolver muchos problemas MIP^{TNF} y MIP^f, con lo que el tiempo aumenta. De este modo, la estrategia de utilizar las soluciones cuasi-óptimas de la Tabla 4.2 como cotas superiores iniciales del algoritmo BFC nos permite llegar a la solución óptima con más rapidez y eficiencia, ya que evitamos que resuelva problemas MIP^{TNF} y MIP^f.

Una vez examinados en detalle el algoritmo y los optimizadores para los casos de pequeñas dimensiones, nos proponemos evaluar la eficiencia del BFC en problemas de mayores dimensiones que en representación compacta no se pueden resolver, como son los modelos P3, P4, P5 y P6. Para ello, en las siguientes tablas recogemos los resultados de este algoritmo aplicado a estos cuatro casos de estudio bajo el optimizador CPLEX. Manteniendo la notación habitual, en estas tablas se detallan las soluciones óptimas (o incumbentes en caso de que en el tiempo límite no llegue

al óptimo) así como el tiempo (en segundos) requerido para obtenerlas, la cantidad de problemas MIP^{TNF} y MIP^{f} que ha de resolver y el número de nodos del árbol BF que examina.

En la Tabla 4.5 aparecen los resultados para el caso P5. En esta ocasión, el algoritmo BFC no logra llegar a la solución óptima dentro del tiempo máximo establecido (3 horas), por lo que hacemos una comparación de este esquema según la cota superior inicial escogida. Conseguimos mejores resultados utilizando como cota superior inicial la solución cuasi-óptima que cuando inicializamos esta cota a ∞. Por un lado, la solución incumbente a la que se llega está más cerca al óptimo, ya que es más pequeña, y por otro lado el número de nodos del árbol BF que examina es mayor con cota superior que sin cota superior inicial. Hay que destacar que la solución incumbente a la que se llega con cota superior inicial para 2 y 4 racimos de escenarios o clusters coincide con la solución incumbente obtenida al resolver el problema en representación compacta, que se recoge en la Tabla 4.2. Esto nos hace pensar que posiblemente la mejor elección de número de clusters, $\hat{p}$, sea 2 ó 4, a pesar de que en el mismo tiempo (3 horas) para otro valor de $\hat{p}$ el número de nodos examinados es mayor, como es el caso de $\hat{p} = 16$.

Tabla 4.5: Resultados BFC sin y con cota superior bajo CPLEX para el caso P5

		Sin cota superior				Con cota superior			
Caso	$\hat{p}$	Z_{MIP}	n^{TNF}	n^{f}	nodos	Z_{MIP}	n^{TNF}	n^{f}	nodos
P5	2	-80.8194*	6	7	20	-84.2161*	5	6	154
	4	-80.8194*	7	8	25	-84.2161*	24	25	601
	8	-80.8194*	7	8	28	-84.2065*	33	32	792
	16	-80.8191*	10	5	39	-84.2065*	30	20	1240
	32	-80.8191*	15	1	51	-84.2065*	27	3	844
	64	-80.8191*	15	1	51	-84.2065*	21	1	550
	128	-80.8191*	15	1	51	-84.2065*	4	1	332

*: Solución incumbente en el tiempo límite (3 horas = 10800 segundos)

En la Tabla 4.6 recogemos los resultados para los casos P3, P4 y P6 utilizando como cota superior inicial la que aparece en la Tabla 4.2 en cada caso. Podemos observar que mediante la descomposición BFC hemos obtenido la solución óptima de los problemas mientras que en su representación compacta no éramos capaces de llegar a ella. Aunque en uno de los casos, como es el problema P6, la solución incumbente obtenida en la representación compacta coincide con la solución óptima del problema.

Además, podemos apreciar la importancia de la elección del número de clusters o racimos, $\hat{p}$, al igual que ocurría en los casos P1 y P2 de pequeñas dimensiones. Vemos que la elección de $\hat{p}$ depende totalmente del problema a resolver, pero de alguna manera influye el número de escenarios que tiene cada uno de los racimos, ya que a la hora de calcular la cota inferior $\underline{Z}$ ha de resolver tantos problemas estocásticos mixtos 0-1 como clusters o racimos haya en la partición del árbol considerada, y cuanto mayores sean las dimensiones de éstos, más complicado resultará obtener esta cota. Pero por otro lado, cuanto más dividamos el número de escenarios, la cota

Tabla 4.6: Resultados BFC bajo CPLEX para problemas de altas dimensiones

Caso	$\hat{p}$	Z_{MIP}	n^{TNF}	n^f	nodos	T_{MIP}^{CPLEX}
P3	2	-59.4645	2	3	3	5266
	5	-59.4645	2	3	3	4490
	7	-59.4645	2	3	3	4539
	10	-59.4645	2	3	3	4592
	14	-59.4645	1	2	3	2659
	35	-59.4645	1	2	3	2653
	70	-59.4645	1	2	8	1831
P4	2	-70.7906*	1	2	7	-
	4	-70.7906	3	4	69	5472
	8	-70.7906	4	2	115	1152
	16	-70.7906	7	1	262	1343
	32	-70.7906	15	1	664	4884
	64	-70.7906*	7	1	681	-
	128	-70.7906*	3	1	503	-
P6	2	-66.0478	3	4	23	9436
	5	-66.0478	4	5	25	8225
	10	-66.0478	4	5	26	8979
	20	-66.0478	3	4	66	7092
	40	-66.0478	5	6	135	8116
	100	-66.0315*	10	2	243	-
	200	-66.0315*	5	1	292	-

-: Se ha excedido el tiempo límite (3 horas = 10800 segundos)
*: Solución incumbente en el tiempo límite

inferior conseguida en cada nodo será peor (más pequeña), por lo que no será muy útil para cortar la rama y hará que deba examinar más nodos. En concreto, para el problema P3 la elección óptima del número de racimos es $\hat{p} = 70$, para el problema P4 sería 8 ó 16, ya que para estas dos elecciones el tiempo de resolución es similar, aunque para $\hat{p} = 8$ el número de nodos examinados es aproximadamente la mitad que para $\hat{p} = 16$ y el número de problemas MIP^{TNF} y MIP^f es ligeramente menor (6 frente a 8). Para el problema P6 vemos que la elección óptima es dividir en 20 racimos, así los submodelos MIP^p modelizan la incertidumbre con 10 escenarios posibles.

Fijándonos en los resultados de los problemas de optimización de las Tablas 4.4 y 4.6, comprobamos que muchas de las soluciones cuasi-óptimas recogidas en la Tabla 4.2 eran realmente el óptimo del problema estocástico a resolver. Esto ha permitido que al tomar estas soluciones como cotas superiores iniciales mejoren muchos los tiempos de ejecución, así como el número de nodos que tiene que recorrer del árbol BF. En un primer momento podría parecer innecesario aplicar la metodología BFC para los casos en los que la cota superior inicial coincide con el óptimo; sin embargo, hay que destacar que si no se hubiese resuelto con este esquema, no se podría tomar como solución óptima ya que nunca habría sido demostrado.

Capítulo 5

Conclusiones

En este trabajo abordamos la resolución de problemas de optimización estocástica mixtos 0-1 de dos etapas. Repasamos las distintas representaciones de estos modelos y las distintas técnicas de resolución desarrolladas en la literatura. En concreto nos centramos en el desarrollo del esquema BFC (del inglés *Branch and Fix Coordination*) para modelos estocásticos bietapa.

Partimos del esquema de descomposición que aparece en [12] e introducimos una cota superior inicial. Esta cota superior es la solución cuasi-óptima que proporciona la mayoría de los optimizadores con una tolerancia de optimalidad ρ determinada.

En la implementación del algoritmo utilizamos dos software de optimización distintos: COIN-OR y CPLEX. En la experiencia computacional comprobamos que el segundo mejora en gran medida al primero. La principal razón de esta ventaja es que CPLEX incorpora las técnicas de preproceso y paralelización, que permiten reducir tiempo y aumentar por tanto la eficiencia en problemas de grandes dimensiones.

En este sentido, comprobamos la eficiencia del esquema BFC a la hora de resolver problemas de optimización de grandes dimensiones. Cuando el problema en representación compacta se puede resolver de forma directa, la implementación del esquema propuesto puede llegar a aumentar el tiempo de resolución, aunque también llegue al óptimo. En cambio, cuando el problema no se puede resolver por los medios convencionales en su representación compacta, la implementación del BFC sobre la representación en variables divididas sobre los racimos de escenarios o clusters se hace indispensable, ya que con él conseguimos llegar a la solución óptima, aunque en ocasiones el tiempo requerido sea muy alto.

Por otro lado, comprobamos en la experiencia computacional que la introducción de la cota superior inicial en el esquema BFC mejora la eficiencia del mismo, ya que gracias a ella examina menos nodos del árbol BF y resuelve menos problemas MIP^{TNF} y MIP^{f}.

Cuando las dimensiones son muy grandes, ni la introducción de cotas superiores consigue encontrar el óptimo dentro de un tiempo considerable. Este hecho nos proporciona una futura línea de investigación. En primer lugar, podríamos encontrar nuevas formas de cortar las ramas del árbol BF. Por un lado, podríamos estudiar nuevas relaciones de desigualdad entre los distintos valores de la función objetivo de los problemas resueltos durante el recorrido del árbol; y por otro lado, podríamos analizar cómo aprovechar las soluciones de estos problemas para evitar resolver tantos problemas MIP^{TNF} y MIP^{f}, que es dónde se requiere más tiempo. En segundo lugar, para mejorar el tiempo de ejecución podríamos bifurcar en las variables binarias de segunda etapa, de esta manera los subproblemas a resolver serán de menores dimensiones. Y en tercer lugar, viendo la superioridad de CPLEX sobre COIN-OR, podríamos incluir la idea del preproceso y de la paralelización directamente en nuestro algoritmo BFC.

Capítulo 6

Comentarios y trabajo futuro

Como hemos visto, el algoritmo BFC propuesto llega a la solución óptima para cualquier problema de optimización estocástica, ya que es un esquema exacto. Sin embargo, ante modelos de grandes dimensiones tiene dificultades para llegar al óptimo en un tiempo razonable, por lo que deducimos que este algoritmo es mejorable.

Este trabajo se ha presentado en el X Workshop de Banca y Finanzas Cuantitativas los días 4-6 de julio de 2012, celebrado en Bilbao. En él ha sido comentado por Aitziber Unzueta, y gracias a sus comentarios hemos visto varias líneas de investigación que nos permiten mejorar el algoritmo.

Los puntos fuertes de este esquema son las cotas, tanto inferior como superior. Por esta razón, las principales modificaciones planteadas están enfocadas sobre estas cotas.

En primer lugar vemos cómo se puede modificar la cota inferior para mejorar la eficiencia del algoritmo. Recordamos que definimos una cota inferior en el nodo inicial de la siguiente manera:

$$\underline{Z}_0 = \sum_{p=1}^{\hat{p}} z_0^p$$

donde z_0^p es la solución del problema MIP^p (3.4), $\forall p = 1, ..., \hat{p}$.

En este punto podemos incluir la relajación Lagrangiana para obtener una cota inferior mejor a ésta. Se introduce el método de la descomposición Lagrangiana como se plantea en [31]. A partir de las soluciones obtenidas al resolver los problemas mixtos 0-1 bajo cada escenario $\hat{p}$ (3.4) y fijado un número de iteraciones k, iteramos hasta determinar la solución de la descomposición Lagrangiana correspondiente, $z_{LD}^p(\mu^p)$, donde $\mu^p = (\mu_\delta^p, \mu_x^p)$ corresponde a los multiplicadores de Lagrange. Para obtener estos valores resolvemos la siguiente sucesión de problemas. Si $p = 2, 3, ..., \hat{p}$,

el problema a resolver es:

$$\begin{aligned} z^p_{LD}(\mu^p) = \text{mín} \quad & \left[w^p c_1 + \left(\mu^p_\delta - \mu^{(p-1)}_\delta\right)\right]\delta^p + \left[w^p c_2 + \left(\mu^p_x - \mu^{(p-1)}_x\right)\right]x^p + \\ & + \sum_{\omega\in\Omega^p} w^\omega [q^\omega_1 \gamma^\omega + q^\omega_2 y^\omega] \\ s.a. \quad & b_1 \leq A \begin{pmatrix} \delta^p \\ x^p \end{pmatrix} \leq b_2 \\ & h^\omega_1 \leq T^\omega \begin{pmatrix} \delta^p \\ x^p \end{pmatrix} + W^\omega \begin{pmatrix} \gamma^\omega \\ y^\omega \end{pmatrix} \leq h^\omega_2, \quad \forall\omega \in \Omega^p \\ & x^p \geq 0, \delta^p \in \{0,1\}, \\ & y^\omega \geq 0, \gamma^\omega \in \{0,1\}, \quad \forall\omega \in \Omega^p, \quad p = 2,3,...,\hat{p}. \end{aligned} \tag{6.1}$$

Y si $p = 1$ hay que resolver el siguiente modelo:

$$\begin{aligned} z^1_{LD}(\mu^1) = \text{mín} \quad & \left[w^1 c_1 + \left(\mu^1_\delta - \mu^{\hat{p}}_\delta\right)\right]\delta^1 + \left[w^1 c_2 + \left(\mu^1_x - \mu^{\hat{p}}_x\right)\right]x^1 + \\ & + \sum_{\omega\in\Omega^p} w^\omega [q^\omega_1 \gamma^\omega + q^\omega_2 y^\omega] \\ s.a. \quad & b_1 \leq A \begin{pmatrix} \delta^1 \\ x^1 \end{pmatrix} \leq b_2 \\ & h^\omega_1 \leq T^\omega \begin{pmatrix} \delta^1 \\ x^1 \end{pmatrix} + W^\omega \begin{pmatrix} \gamma^\omega \\ y^\omega \end{pmatrix} \leq h^\omega_2, \quad \forall\omega \in \Omega^1 \\ & x^p \geq 0, \delta^1 \in \{0,1\}, \\ & y^\omega \geq 0, \gamma^\omega \in \{0,1\}, \quad \forall\omega \in \Omega^1. \end{aligned} \tag{6.2}$$

Así obtenemos una nueva cota inferior,

$$z_{LD} = \sum_{p=1}^{\hat{p}} z^p_{LD}(\mu^p),$$

y la correspondiente solución (δ_{LD}, x_{LD}, γ^p_{LD}, y^p_{LD}).

Estos valores de las soluciones guardan las siguientes relaciones de desigualdad:

$$\underline{Z}_0 \leq z^1_{LD} \leq z^2_{LD} \leq ... \leq z_{LD} \leq z_{MIP}.$$

De manera que z_{LD} es una cota inferior mejor del problema original ya que es mayor que $\underline{Z}_0$, es decir, está más próxima al valor real de la solución del problema.

Con este planteamiento, podríamos reescribir el Paso 1 de nuestro algoritmo de la siguiente manera:

Paso 1: Resolver los problemas mixtos 0-1 bajo cada cluster, MIP^p (3.4), $\forall p = 1,\ldots,\hat{p}$, y calcular $\underline{Z}_0 = \sum_{p=1}^{\hat{p}} z^p_0$. Si algún problema es infactible, entonces parar (el problema original es infactible). Iterar k veces hasta determinar $\underline{Z}_0 =$

$z^k_{LD} = \sum_{p=1}^{\hat{p}} z^{(k)p}_{LD}(\mu^p)$, resolviendo los problemas (6.1) y (6.2), y obtener la correspondiente solución (δ_{LD}, x_{LD}, γ^p_{LD}, y^p_{LD}).

Si existe alguna variable δ_{LD} (0-1) que no satisface (3.5), ir al Paso 2.

Si existe alguna variable continua x_{LD} que no satisface (3.6), ir al Paso 6.

En otro caso, $\overline{Z} := z^k_{LD}$ y parar (se ha encontrado la solución óptima del problema original).

Mediante la utilización de la descomposición Lagrangiana en [31] vemos que se puede obtener un intervalo que contiene a la solución mejor que la solución incumbente a la que hemos llegado con el BFC. Es decir, en este trabajo, la mejor solución a la que llegamos tras tres horas de ejecución es -84.2161 eligiendo una partición en 2 ó 4 racimos de escenarios, mientras que el mejor intervalo obtenido con la descomposición Lagrangiana es [-84.9872, -84.1637]. Obviamente, este intervalo incluye una solución mejor que la incumbente a la que hemos llegado usando el BFC.

La segunda modificación que puede admitir el algoritmo es sobre la cota superior que se va actualizando a medida que recorremos el árbol BF. Para introducir este cambio definimos tres nuevos problemas a resolver a lo largo del esquema.

En cada nodo, una serie de variables enteras de primera etapa, δ, han sido fijadas. La siguientes variables δ aún están libres, es decir, sin ramificar ni fijar. Cuando ramificamos una de las variables, calculamos la cota inicial correspondiente en ese nodo. Para obtenerla, resolvemos los $\hat{p}$ problemas mixtos 0-1 bajo cada escenario, MIP^p (3.4). Si las variables enteras de primera etapa, δ, obtenidas en estas soluciones cumplen las condiciones de no anticipatividad (3.5),todas ellas toman el mismo valor sobre los distintos racimos de escenarios. Por su parte, las variables enteras de segunda etapa, γ, toman un valor determinado en cada uno de estos clusters. De este modo, una vez resueltos estos $\hat{p}$ problemas y obtenido sus soluciones, podemos plantear el siguiente problema con solución z^{TNF}_{LP}:

$$\begin{array}{lll} (LP^{TNF}_i): & LP^{TNF} & \\ & s.a. & \delta_j = \overline{\delta}_j, \quad \forall j \in \{1,...,i\} \quad \text{(fijadas por la rama)} \\ & & \delta_j = \overline{\delta}_j, \quad \forall j \in \{i+1,...,n_\delta\} \quad \text{(solución de } MIP^p_i\text{)} \\ & & \gamma_j = \overline{\gamma}_j, \quad \forall j \in \{1,2,...,n_\gamma\} \quad \text{(solución de } MIP^p_i\text{)} \end{array} \tag{6.3}$$

Si permitimos que las variables γ tomen el valor entero 0 ó 1 en vez de fijarlas al valor obtenido al resolver los problemas MIP^p (3.4), $\forall p = 1,..,\hat{p}$, resulta el siguiente problema de optimización:

$$\begin{array}{lll} (MIP^{TNF}_i): & MIP^{TNF} & \\ & s.a. & \delta_j = \overline{\delta}_j, \quad \forall j \in \{1,...,i\} \quad \text{(fijadas por la rama)} \\ & & \delta_j = \overline{\delta}_j, \quad \forall j \in \{i+1,...,n_\delta\} \quad \text{(solución de } MIP^p_i\text{)} \\ & & \gamma_j \in \{0,1\}, \quad \forall j \in \{1,2,...,n_\gamma\} \end{array} \tag{6.4}$$

cuya solución denotamos por z^{TNF}_{MIP}.

También podemos dejar que las variables δ que aún no han sido ramificadas ni

fijadas tomen el valor 0 ó 1, obteniendo así el siguiente problema a optimizar:

$$
\begin{array}{lll}
(MIP_i^f): \quad MIP^f & & \\
\quad s.a. & \underline{Z} = \sum_{p=1}^{\hat{p}} z_i^p \leq z_{MIP}^{TNF} & \\
& \delta_j = \bar{\delta}_j, & \forall j \in \{1, ..., i\} \quad \text{(fijadas por la rama)} \\
& \delta_j = \in \{0, 1\}, & \forall j \in \{i+1, ..., n_\delta\} \\
& \gamma_j \in \{0, 1\}, & \forall j \in \{1, 2, ..., n_\gamma\}
\end{array}
\tag{6.5}
$$

Vemos que este problema añade la restricción de que el valor de la solución del problema (6.4) tiene que ser mayor que la cota inferior calculada en el nodo en el que nos encontramos. La solución de este problema la denotamos como z_{MIP}^f.

Definidos estos nuevos problemas, los utilizamos para calcular nuevas cotas superiores en cada nodo. Para ello, habría que seguir el algoritmo planteado modificando el Paso 6 de la siguiente manera:

Paso 6: Resolver LP_i^{TNF} (6.3) para que las variables x satisfagan (3.6). Notar que el valor de la solución se denota por z_{LP}^{TNF}. Si el problema es infactible e $i = 0$, parar (el problema original es infactible); si $i > 0$, ir al Paso 7.

Actualizar $\overline{Z} := \text{mín}\{z_{LP}^{TNF}, \overline{Z}\}$.

Resolver MIP_i^{TNF} (6.4), cuya solución se denota por z_{MIP}^{TNF}. Si el problema es infactible e $i = 0$, parar (el problema original es infactible); si $i > 0$, ir al Paso 7.

Actualizar $\overline{Z} := \text{mín}\{z_{MIP}^{TNF}, \overline{Z}\}$.

Si $i = n$ entonces ir al Paso 7.

Resolver MIP_i^f (6.5). El valor de la solución en este caso se denota por z_{MIP}^f. Si el problema es infactible e $i = 0$, parar (el problema original es infactible); si $i > 0$, ir al Paso 7.

Si $z_{MIP}^{TNF} = z_{MIP}^f$ o $z_{MIP}^f \geq \overline{Z}$, entonces ir al Paso 7.

Si $z_{MIP}^f < \overline{Z}$ y todas las variables δ cumplen (3.5), actualizar $\overline{Z} := z_{MIP}^f$ e ir al Paso 7. En otro caso ir al Paso 3.

Una vez planteadas estas bases teóricas, sólo falta comprobar empíricamente que son más efectivas que el algoritmo BFC que hemos propuesto en este trabajo. Para ello, lo más adecuado sería aplicar este nuevo esquema al caso P5, que es el que más tiempo tarda en resolverse con el BFC. Esto queda como futuro trabajo, al igual que desarrollar nuevas técnicas que puedan mejorar la eficiencia del BFC

Apéndice A

Programación

En este apédice se presenta una breve descripción de los programas implementados en C++ empleados para realizar la experiencia computacional realizada en el Capítulo 4. Los programas resuelven los distintos casos de estudio utilizando el esquema de descomposición BFC (del inglés, *Branch and Fix Coordination*) sobre el problema en representación en variables dividas sobre los racimos de escenaios y también resuelven el DEM en representación compacta.

Los programas principales son los siguientes:

compactoCOIN.cpp, programa principal que optimiza el DEM en representación compacta utilizando el optimizador COIN-OR.

compactoCPLEX.cpp, programa principal que optimiza el DEM en representación compacta utilizando el optimizador CPLEX.

bfctsCOIN.cpp, programa principal que resuelve el DEM en representación en variables divididas sobre los racimos de escenarios o clusters empleando la descomposición BFC descrita en la Sección 3.3 utilizando el optimizador COIN-OR.

bfctsCPLEX.cpp, programa principal que resuelve el DEM en representación en variables divididas sobre los clusters empleando la descomposición BFC descrita en la Sección 3.3 utilizando el optimizador CPLEX.

cotaCOIN.cpp, programa principal que calcula, usando COIN-OR, la solución cuasi-óptima del problema DEM en representación compacta con una tolerancia de optimalidad ρ.

cotaCPLEX.cpp, programa principal que calcula la solución cuasi-óptima del problema DEM en representación compacta con una tolerancia de optimalidad ρ. Utiliza el optimizador CPLEX.

modelos.cpp, subrutina que genera los parámetros deterministas del modelo, así como los parámetros inciertos para cada escenario. Se utiliza en todos los programas principales.

constantes.h, subrutina asociada a los seis programas principales que recoge los parámetros de entrada, como son n_δ, n_x, n_γ, n_y, $|\Omega|$, $\hat{p}$ y m (número de restricciones).

nacbin.cpp, subrutina que comprueba si las variables 0-1 de primera etapa cumplen las restricciones de no anticipatividad y si las variables 0-1 de primera y segunda etapa son enteras.

naccon.cpp, subrutina que comprueba la no anticipatividad de las variables continuas de primera etapa.

param3.cpp, subrutina utilizada por los seis programas principales que genera los coeficientes de la función objetivo y de las restricciones para el modelo en representación compacta.

param4.cpp, subrutina que genera los coeficientes de la función objetivo y de las restricciones para el modelo en representación extendida sobre racimos de escenarios o clusters. Se emplea en los programas principales bfctsCOIN.cpp y bfctsCPLEX.cpp.

semilla.cpp, subrutina que genera números aleatorios de una distribución uniforme a partir de una semilla dada.

Bibliografía

[1] S. Ahmed, M. Tawarmalani, y N. V. Sahinidis. A multi-stage stochastic integer programming approach for capacity expansion under uncertainty. *Mathematical Programming*, 100:355–377, 2004.

[2] A. Alonso-Ayuso, L. Escudero, A. Garín, M. Ortuño and G. Pérez. An approach for strategic supply chain planning based on stochastic 0–1 programming. Journal of Global Optimization 2003a;26; 97–124.

[3] A. Alonso-Ayuso, L. Escudero, A. Garín, M. Ortuño and G. Pérez. On the product selection and plant dimensioning problem under uncertainty. Omega, Journal of Management 2005a;33; 307–318.

[4] A. Alonso-Ayuso, L. Escudero, M. Ortuño. BFC, a branch-and-fix coordination algorithmic framework for solving some types of stochastic pure and mixed 0-1 programs. European Journal of Operational Research 2003b;151; 503–519.

[5] A. Alonso-Ayuso, L. Escudero, M. Ortuño. Modeling production planning and scheduling under uncertainty. In: Applications of Stochastic Programming, pp. 217–252. S.W. Wallace and W.T. Ziemba (Eds.). MPS-SIAM-Series in Optimization 2005b.

[6] J. F. Benders. Partitioning procedures for solving mixed variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[7] J. R. Birge y F. V. Louveaux Introduction to Stochastic Programming. Ed. Springer Series in Operations Research.

[8] C. Carøe y R. Schultz. Dual decomposition in stochastic integer programming. Operations Research Letters 1999;24; 37–45.

[9] C. C. Carøe y J. Tind. L-shaped decomposition of two-stage stochastic programs with integer recourse. *Mathematical Programming*, 83:451–464, 1998.

[10] L. F. Escudero, M. A. Garín, M. Merino y G. Pérez. A two-stage stochastic integer programming approach as a misture of Branch-and-Fix coordination and Benders' decomposition schemes. Annals of Operations Research 152 (1) (2007), 395-327.

[11] L. F. Escudero, M. A. Garín, M. Merino y G. Pérez. On multistage stochastic integer programming for incorporating logical constrains in asset and liability management under uncertainty. Comutacional Management Science 6 (3) (2009), 307-327.

[12] L. F. Escudero, M. A. Garín, M. Merino y G. Pérez. An exact algorithm for solving large-scale two-stage stochastic mixed-integer problems: Some theoretical and experimental aspects. European Journal of Operational Research 204 (2010) 105-116.

[13] A. M. Geoffrion. Elements of Large Scale Mathematical Programming. Management Science, 16:652-691, 1970.

[14] R. Hemmecke y R. Schultz. Decomposition methods for two-stage stochastic integer programs. En: Online Optimization of Large Scale Systems, pp. 601–622. M. Grotschel, S.O. Krumke and J. Rambau (Eds.) Springer 2001.

[15] IBM ILOG. CPLEX v12.2. http://www.ilog.com/products/cplex; 2011.

[16] INFORMS. COIN-OR: COmputational INfrastucture for Operations research. www.coin-or.org, 2010.

[17] W. K. Klein Haneveld y M. H. van der Vlerk Kang. Stochastic integer programming: General models and algorithms. *Annals of Operations Research*, 85:39–57, 1999.

[18] W. Klein Haneveld y M. Van der Vlerk Kang. Optimizing electricity distribution using integer recourse models. En: Stochastic Optimization: Algorithms and Applications, pp. 137–154. S. Uryasev and P.M. Pardalos (Eds.) Kluwer AcademicPublishers 2001.

[19] M. I. Kusy y W. T. Ziemba. A bank asset and liability management model. Operational Research, 34:356-376, 1986.

[20] G. Laporte y F.V. Louveaux. An integer l-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research*, 50:415–423, 2002.

[21] M. Nowak, R. Schultz y W. Westphalen. Optimization of simultaneous power production and trading by stochastic integer programming 2002. Technical report, Stochastic Programming E-Print Series, http://dochost.rz.huberlin.de/speps.

[22] L. Ntaimo y S. Sen. The million variable 'march' for stochastic combinatorial optimization. *Journal of Global Optimization*, 32:385–400, 2005.

[23] R. T. Rockafellar y R. J-B. Wets. Scenario and policy aggregation in optimization under uncertaninty. Mathematics of Operations Research, 16:119-147, 1991.

[24] W. Romisch y R. Schultz. Multi-stage stochastic integer programs: An introduction. En: Online Optimization of Large Scale Systems, pp. 581–600. M. Grotschel, S.O. Krumke and J. Rambau (Eds.). Springer 2001.

[25] R. Schultz. Stochastic programming with integer variables. Mathematical Programming Series B 2003;97; 285–309.

[26] R. Schultz, L. Stougie, y M.H. van der Vlerk. Two-stage stochastic integer programming: a survey. *Statistika Neerlandica*, 50:404–416, 1996.

[27] S. S. Sen y H. D. Sherali. Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming. *Mathematical Programming. Series A*, 105:203–223, 2006.

[28] S. S. Sen y J. L. Higle. The c^3 theorem and the d^2 algorithm for large scale stochastic mixed-integer programming: set convexification. *Mathematical Programming*, 104:1–20, 2005.

[29] H. D. Sherali y B. M. P. Fraticelli. A modification of benders' decomposition algorithm for discrete subproblems: An approach for stochastic programs with integer recourse. *Journal of Global Optimization*, 22:319–342, 2002.

[30] H. D. Sherali y X. Zhu. On solvig discrete two-stage stochastic programs having mixed-integer first- and second-stage varaibles. Mathematical Programming, Serie B 108, 597-616 (2006).

[31] A. Unzueta. Efficient methodologies for the treatment of large-scale stochastic optimization problems. PhD. Thesis. Universidad del País Vasco, UPV/EHU, 2012.

[32] R. J-B Wets. Stochastic programs with fixed recourse: The equivalent deterministic program. SIAM Review, 16:309-339, 1974.

[33] R. J-B Wets. On the relation between stochastic and deterministic optimization. Lecture Notes in Economonics and Matematical Systems, Serie 10, 350-361, 1974.

[34] S. A. Zenios Financial Optimization. Cambridge University Press.