

Máster Interuniversitario en Estadística e Investigación Operativa

Título: OPTIMIZACIÓN DE MODELOS ESTOCÁSTICOS DE MERCADO ELÉCTRICO MÚLTIPLE MEDIANTE MÉTODOS DUALES

Autor: ALDASORO MARCELLAN, UNAI

Director: HEREDIA CERVERA, FRANCISCO JAVIER

Departamento: DEPARTAMENT D'ESTADÍSTICA I INVESTIGACIÓ OPERATIVA

Universidad: UNIVERSITAT POLITÈCNICA DE CATALUNYA

Convocatoria: FEBRERO DE 2011



Memoria TFM

Master en Estadística e Investigación Operativa

Optimización de modelos estocásticos de mercado eléctrico múltiple mediante métodos duales

AUTOR: ALDASORO MARCELLAN, UNAI

DIRECTOR: HEREDIA CERVERA, FRANCISCO JAVIER

*Departament D'Estadística i Investigació Opertiva,
Facultat de Matemàtiques i Estadística,
Universitat Politècnica de Catalunya*

Barcelona, febrero de 2011

El presente trabajo forma parte del *Proyecto de Investigación DPI2008-02153* del *Ministerio de Ciencia e Innovación* del *Gobierno de España*.

El presente trabajo ha sido seleccionado en la 4^o convocatoria de las *Ayudas CERMET* de la *Facultat de Matemàtiques i Estadística* de la *Universitat Politècnica de Catalunya*.

gnom.upc.edu/heredia

RESUMEN

El presente trabajo plantea la resolución computacional de un modelo de optimización de la oferta de generación eléctrica para compañías eléctricas que participan en el mercado eléctrico liberalizado *MIBEL*. Dicho mercado se circunscribe a España y Portugal y se compone de una serie de subastas energéticas consecutivas donde el operador de mercado realiza para cada una de ellas la casación entre la oferta y demanda. Así, el objetivo de la compañía generadora será maximizar los beneficios obtenidos en la participación del conjunto de mercados teniendo en cuenta el cumplimiento de las obligaciones contractuales ya establecidas.

El modelo matemático propuesto para su caracterización corresponde a un modelo de programación estocástica multietapa cuyo equivalente determinista es un problema de optimización cuadrática con variable binaria. Con el objetivo de aprovechar la estructura del problema se procede a plantear la dualización de un grupo de restricciones que producen que el problema original pueda ser dividido en subproblemas.

Para su resolución se deberá estudiar la idoneidad de diversos métodos duales (subgradiente, Bundle Methods, ACCPM) y seleccionar el más conveniente para el caso abordado. La decisión finalmente adoptada ha consistido en elegir como método de resolución el algoritmo *Proximal Bundle Method* descrito en [18] y adaptado satisfactoriamente a problemas de coordinación de la generación hidro-térmica [17].

El análisis de método *Proximal Bundle Method* corresponderá a su comprensión e interpretación gráfica, a la resolución de un ejemplo de pequeña escala de manera analítica y a su resolución computacional. El objetivo de la fase de resolución será valorar el proceso iterativo y la convergencia del *Proximal Bundle Method* aplicado al problema multimercado de oferta óptima y la comparación de resultados respecto a otro método dual como el método del subgradiente.

La implementación computacional se realizará mediante el lenguaje C++, específicamente se utilizará el metalenguaje *Concert Technology* creado por IBM para el enlace entre el código C++ y el solver *CPLEX*. Se comprueba que dicho lenguaje tiene como ventajas principales su simplicidad estructural y el compacto código que produce. No obstante la implementación del *Proximal Bundle Method* manifiesta una serie de limitaciones prácticas de *Concert Technology* en cuanto al almacenado y actualización de problemas de optimización. Se propone como línea de futuro el análisis de lenguajes alternativos.

En todo caso, los resultados obtenidos desprenden que el *Proximal Bundle Method* se adapta satisfactoriamente al problema multimercado de oferta óptima, además se concluye que en la aplicación numérica considerada un tamaño de *Bundle* ilimitado produce los mejores resultados.

Además en trabajo propone una serie de líneas de investigación futuras en las que destacan la paralelización de la resolución de los subproblemas, y la definición del subproblema asociado a cada térmica como un problema de caminos mínimos, en la línea de lo sugerido en [30].

Palabras clave

Investigación operativa, programación matemática, programación estocástica, optimización dual, mercados eléctricos.

gnom.upc.edu/heredia

ABSTRACT

This project describes the computational resolution of an optimization model regarding the electricity supply of a generation company participating in the deregulated electricity market called MIBEL. This market is limited to Spain and Portugal and is composed of a series of consecutive energy auction markets where the operator agent matches the offer and demand in each time interval. In this context, the generation company's goal corresponds to maximize the benefits of its participation in the market series while taking the contractual obligations into account.

The mathematical model created to characterize the market mechanisms correspond to a multistage stochastic programming model whose deterministic equivalent can be presented as a quadratic optimization problem with binary variables. In order to exploit the internal structure of the problem, a dualization of a group of constraints is proposed so that the primary problem could be divided in subproblems.

Concerning the resolution process, the suitability of different dual methods (subgradient, Bundle Methods, ACCPM) will be analysed in order to choose the method that is better adapted to the addressed case. After this process, the final decision has corresponded to selection of the Proximal Bundle Method as solving method, following the guidelines established in [18]. This method has been successfully adapted to solve hydro-thermal coordination problems in [17].

The process of analyzing and graphical understanding of the Proximal Bundle Method has been carried out by comprehending the theoretical basis of the algorithm, by an analytical resolution of a small scale example and by a computational resolution of a real scale example. The aim of the resolution phase is to assess the iterative process and the convergence of the Proximal Bundle Method applied to the multimarket optimal bidding problem and its comparison with the results obtained by other dual methods such as subgradient method.

The computational implementation has been performed using the C++ language, specifically IBM's Concert Technology metalanguage has been used as a link between the C++ code and the CPLEX solver. The decision has been based on its structural simplicity and the compact code that it produces. However, the implementation of the Proximal Bundle Method has shown a series of practical limitations, mainly regarding the administration of the iterative optimization problems, their storage and updating. In this context a research concerning alternative implementation languages is advised.

The results obtained following the steps described in the previous lines have shown that the Proximal Bundle Method can be successfully adapted to the multimarket optimal bidding problem. Moreover, it can be concluded that the numerical implementation considering an unlimited size Bundle produces the best results found.

This project also suggests a series of future research directions, emphasizing the parallelization of the subproblem solving process and the definition of the subproblem associated with each power plant as a shortest path network flow problem, following the idea presented in [30].

Keywords

Operations research, mathematical programming, stochastic programming, dual optimization, electricity markets.

gnom.upc.edu/hereada

ÍNDICE DE CONTENIDO

1.	INTRODUCCIÓN	1
1.1.	Contexto académico.....	1
1.2.	Motivación personal	1
1.3.	Formación previa	1
1.4.	Contexto de investigación: GNOM	2
1.5.	Entidades participantes y financiación.....	2
1.6.	Situación personal durante el desarrollo del trabajo.....	3
2.	OBJETIVOS.....	4
3.	DESCRIPCIÓN DEL PROBLEMA	6
3.1.	Contexto.....	6
3.2.	Estructura del mercado	6
3.3.	Hipótesis consideradas en el modelo matemático:	8
4.	MODELO DE OPTIMIZACIÓN	9
4.1.	Caracterización de la incertidumbre	9
4.2.	Modelo matemático asociado al problema	10
4.3.	Evolución del modelo multimercado de oferta óptima.....	15
5.	MÉTODO DE RESOLUCIÓN: <i>ACCPM</i>	16
5.1.	Interés.....	16
5.2.	Aplicaciones existentes de <i>ACCPM</i>	16
5.3.	Analytic Center Cutting Plane Method.....	17
5.4.	Valoración del <i>ACCPM</i> como método de resolución	19
6.	MÉTODO DE RESOLUCIÓN: <i>PROXIMAL BUNDLE METHOD</i>	21
6.1.	Algoritmo <i>Proximal Bundle Method</i>	21
6.2.	Ejemplo de resolución mediante el algoritmo <i>PB</i>	25
7.	ADAPTACIÓN DEL PROBLEMA MULTIMERCADO DE OFERTA ÓPTIMA AL ALGORITMO <i>PB</i>	41
7.1.	Formulación de la función dual Lagrangiana	41
7.2.	Elementos del <i>Bundle Method</i> . Subproblema <i>Proximal Bundle</i>	43
8.	DATOS PARA LA RESOLUCIÓN	45
8.1.	Origen de los datos	45
8.2.	Dimensiones del problema.....	45
9.	IMPLEMENTACIÓN.....	47
9.1.	Lenguaje de programación seleccionado	47
9.2.	Herramientas C++.....	48
9.3.	Problemas de optimización implementados	49
10.	RESULTADOS OBTENIDOS.....	57
10.1.	Resolución del problema original mediante el solver <i>CPLEX</i>	57
10.2.	Resolución de la relajación lineal del problema original	58
10.3.	Resolución del dual lagrangiano mediante el método del subgradiente	59
10.4.	Resolución del dual lagrangiano dividido en subproblemas mediante el método el subgradiente	60
10.5.	Resolución del dual lagrangiano mediante el <i>Proximal Bundle Method</i>	61
11.	PLAN DE TRABAJO	70
11.1.	Planteamiento inicial	70
11.2.	Desarrollo real	70
12.	CONCLUSIONES.....	71
12.1.	Futuras líneas de investigación	73
13.	BIBLIOGRAFIA.....	75

ÍNDICE DE FIGURAS

Figura 1 Esquema de la secuencia del mercado	7
Figura 2 Árbol de escenarios	9
Figura 3 Algoritmo PB, inicialización: subgradiente asociado a \bar{x}	26
Figura 4 Algoritmo PB, iteración 1: función cuadrática asociada a y	28
Figura 5 Algoritmo PB, iteración 1: subgradiente asociado a \bar{x}	29
Figura 6 Algoritmo PB, iteración 2: función cuadrática asociada a y	32
Figura 7 Algoritmo PB, iteración 2: subgradiente asociado a \bar{x}	33
Figura 8 Algoritmo PB, iteración 3: función cuadrática asociada a y	35
Figura 9 Algoritmo PB, iteración 3: subgradiente asociado a \bar{x}	37
Figura 10 Algoritmo PB, ejemplo con $t^1 = 7$, iteración 1.	40
Figura 11 Algoritmo PB, ejemplo con $t^1 = 7$, iteración 2.	40
Figura 12 Esquema estructural de <i>Concert Technology</i>	48
Figura 13 Esquema estructural de <i>Callabe Library</i>	49
Figura 14 Tamaño de <i>Bundle</i> = 5. Valores del dual lagrangiano	62
Figura 15 Tamaño de <i>Bundle</i> = 5. Longitudes de paso	62
Figura 16 Tamaño de <i>Bundle</i> = 5. Cota inferior	62
Figura 17 Tamaño de <i>Bundle</i> = 10. Valores del dual lagrangiano	63
Figura 18 Tamaño de <i>Bundle</i> = 10. Longitudes de paso	63
Figura 19 Tamaño de <i>Bundle</i> = 10. Cota inferior	63
Figura 20 Tamaño de <i>Bundle</i> = 20. Valores del dual lagrangiano	64
Figura 21 Tamaño de <i>Bundle</i> = 20. Longitudes de paso	64
Figura 22 Tamaño de <i>Bundle</i> = 20. Cota inferior	64
Figura 23 Tamaño de <i>Bundle</i> = 30. Valores del dual lagrangiano	65
Figura 24 Tamaño de <i>Bundle</i> = 30. Longitudes de paso	65
Figura 25 Tamaño de <i>Bundle</i> = 30. Cota inferior	65
Figura 26 Tamaño de <i>Bundle</i> = Ilimitado. Valores del dual lagrangiano	66
Figura 27 Tamaño de <i>Bundle</i> = Ilimitado. Longitudes de paso	66
Figura 28 Tamaño de <i>Bundle</i> = Ilimitado. Cota inferior	66
Figura 29 Comparación de los resultados iterativos en el PBM	68
Figura 30 Comparación de los resultados iterativos en el PBM. Zoom	69

ÍNDICE DE TABLAS

Tabla 1. Ejemplo <i>PB</i> con $t^1 = 2$, resumen de resultados analíticos.	37
Tabla 2. Ejemplo <i>PB</i> con $t^1 = 2$, resumen de resultados computacionales. Parte 1. .	38
Tabla 3. Ejemplo <i>PB</i> con $t^1 = 2$, resumen de resultados computacionales. Parte 2. .	38
Tabla 4. Ejemplo <i>PB</i> con $t^1 = 7$, resumen de resultados computacionales. Parte 1. .	39
Tabla 5. Ejemplo <i>PB</i> con $t^1 = 7$, resumen de resultados computacionales. Parte 2. .	39
Tabla 6. Comparación de resultados entre resoluciones realizadas con Proximal Bundle y con el método del subgradiente	67

DOCUMENTACIÓN ANEXA

ANEXO A. ÍNDICE DE ACRÓNIMOS	i
------------------------------------	---

1. INTRODUCCIÓN

1.1. Contexto académico

Este Trabajo de Fin de Master (en adelante TFM) se encuadra dentro del plan de estudios del *Master Interuniversitario en Estadística e Investigación Operativa* coordinada conjuntamente por la *Universitat Politècnica de Catalunya* y la *Universitat de Barcelona*, más concretamente se integra en la especialidad *Investigación Operativa* de dicho master.

Su realización se sitúa tras el periodo docente y su objetivo principal es aglutinar parte del conocimiento adquirido durante el master en un proyecto personal de mayor densidad y profundidad que los realizados durante el periodo docente. El valor académico del TFM corresponde a 30 créditos ECTS (*European Credit and Accumulation System*), lo cual equivaldría a 750 horas de dedicación según los estándares habituales.

La tutoría del TFM ha corrido a cargo del profesor Francisco Javier Heredia Cervera, miembro del *Departament D'Estadística i Investigació Operativa* de la *Universitat Politècnica de Catalunya*, y a su vez miembro del *Group on Numerical Optimization and Modeling* de dicho departamento. Este proyecto se sitúa dentro de las líneas del proyecto de investigación en mercados eléctricos *Short- and Medium-Term Multimarket Optimal Electricity Generation Planning with Risk and Environmental Constraints*. (DPI2008-02153) del Ministerio de Ciencia e Innovación del Gobierno Español apoyando a otra serie de trabajos que se están realizando actualmente dentro del mismo proyecto

1.2. Motivación personal

A la hora de seleccionar el tema del Trabajo de Fin de Master mi principal criterio fue el poder familiarizarme con los diferentes pasos de la resolución de un problema de optimización. Es decir, partiendo de un modelo barajar diferentes métodos de resolución, elegir el más adecuado, valorar la herramienta informática y lenguaje más adecuados para su implementación, realizar el proceso de implementación y valorar los resultados.

Además, el segundo de los objetivos personales ha sido el trabajar en un contexto relacionado con la tecnología y la industria, es decir, enlazar mi formación previa de Ingeniería Industrial con las técnicas de optimización abordadas durante el *Master en Estadística e Investigación Operativa*.

La línea de investigación sobre energía que desarrolla el GNOM abarca estos dos aspectos y es por tanto el contexto elegido para realizar el TFM.

1.3. Formación previa

Durante el Master en Estadística e Investigación Operativa he seguido un proceso de familiarización con el proyecto en cuestión cursando la

asignatura de especialización “*Mercats Elèctrics Liberalitzats*” (se puede consultar la guía docente de la asignatura en la siguiente URL: <https://fme-intranet.upc.edu/tmp/consgd/2010/34436-e-1.pdf>) donde se presentaban las características principales de los mercados eléctricos en España, se abordaban ejemplos de modelización a corto, medio y largo plazo de la planificación de la generación eléctrica y finalmente se procedía a la resolución mediante el lenguaje AMPL [10] de problemas de pequeña escala.

Tras ello realicé el trabajo fin de asignatura del curso “*Mètodes heurístics en programació matemàtica*” (se puede consultar la guía docente en la siguiente URL: <https://fme-intranet.upc.edu/tmp/consgd/2010/34436-e-1.pdf>) abordando mediante diferentes métodos heurísticos el problema de planificación de la generación energética suponiendo un escenario determinista y de mercado único a corto plazo.

Considero que esta formación junto con los estudios de Ingeniería Industrial me han permitido tener una visión cercana del contexto tecnológico y de la dificultad real de la resolución del problema en cuestión.

1.4. Contexto de investigación: GNOM

GNOM (Group on Numerical Optimization and Modeling) es un grupo de investigación de la UPC formado por PDI y alumnos de post-grado y doctorado de los departamentos de *Estadística e Investigación Operativa* y *Matemática Aplicada 1* de la *Universitat Politècnica de Catalunya*. Es grupo de investigación consolidado reconocido y financiado por la Generalitat de Catalunya (SGR-2009-1122). El grupo trabaja tanto en métodos de optimización numérica como en modelos de programación matemática, con especial interés en las aplicaciones a problemas de [energía](#) y [protección de datos estadísticos](#). Puede encontrarse más información sobre GNOM en <http://gnom.upc.edu>

El trabajo a desarrollar en este TFM formará parte del Proyecto de Investigación Fundamental no Orientada “*Short- and Medium-Term Multimarket Optimal Electricity Generation Planning with Risk and Environmental Constraints*”, financiado por el Ministerio de Ciencia e Innovación (Proyecto DPI2008-02153). En concreto, desarrollaría parte de la tarea descrita en la memoria técnica como:

ALG_LAG : To study the suitability of the dual decomposition methods (Lagrangian Relaxation, Cutting Plane, ACCPM) to solve the Stochastic Integrated Multimarket Optimal Bid model.

Es decir, valorar la incorporación de métodos duales de descomposición a la resolución del modelo eléctrico multimercado de oferta óptima.

1.5. Entidades participantes y financiación

El proyecto “*Short- and Medium-Term Multimarket Optimal Electricity Generation Planning with Risk and Environmental Constraints*” mencionado anteriormente cuenta con la participación como Ente Promotor Observador de

la compañía *Gas Natural – Unión Fenosa*. Puede encontrarse más información sobre el proyecto en la url <http://gnom.upc.edu/projects/energy/dpi2008-02153/>

Además el proyecto ha sido seleccionado en la 4^o convocatoria de las *Ayudas CERMET* de la *Facultat de Matemàtiques i Estadística* de la *Universitat Politècnica de Catalunya* que tiene como objetivo promocionar “la realización de trabajos de fin de master que impliquen transferencia de conocimientos”.

1.6. Situación personal durante el desarrollo del trabajo

El presente proyecto arranca formalmente a principios Junio de 2010 con una dedicación a tiempo completo. En este contexto, y siempre dependiendo del desarrollo final del proyecto, se barajo la posibilidad de presentar el Trabajo de Fin de Master en la convocatoria de Octubre de 2010.

Sin embargo a partir de Septiembre de 2010 comienzo a impartir docencia en la *Escuela Técnica Superior de Ingeniería de Bilbao* perteneciente a la *Euskal Herriko Unibertsitatea / Universidad del País Vasco*. Este hecho junto con las dificultades encontradas en las primeras fases del proyecto hacen que planificación del *Trabajo de Fin de Master* se prolongue en el tiempo con tal de poder alcanzar de los objetivos marcados.

El hecho de no residir en Barcelona a provocado además la dificultad de acceso a los centros computacionales disponibles en el *Departament D'Estadística i Investigació Opertiva* de la *Universitat Politècnica de Catalunya*, por lo que las resoluciones se han desarrollado en un ordenador personal de las siguientes características: procesador *Intel Celeron* de un núcleo a 1,60 Hz y 1,99 GB de memoria RAM.

Consecuentemente la valoración de los resultados numéricos se tendrá que centrar en la convergencia de los diferentes métodos aplicados y no tanto en el tiempo computacional, ya que no han sido testados en un entorno que permita explotar sus características estructurales.

2. OBJETIVOS

En este trabajo se abordan los diferentes aspectos relativos a la resolución de un problema de optimización, que de manera genérica podríamos denominar:

- Compresión e interpretación del modelo matemático asociado a un contexto real.
- Análisis y valoración de posibles métodos de resolución.
- Selección del método de resolución.
- Consideración de los diferentes lenguajes de programación y elección del más adecuado.
- Implementación del método.
- Resolución y análisis de resultados.

Específicamente, el presente proyecto pretende abordar los siguientes objetivos:

Objetivo principal:

- Construir la base de un método de resolución que tenga como resultado final la obtención una cota inferior ajustada del problema estocástico mixto descrito en el artículo "Optimal Day-Ahead Bidding in the MIBEL's Multimarket Energy Production System" [05] de manera computacionalmente eficiente. Con ello se pretende mejorar el rendimiento algorítmico de la técnica de resolución adoptada hasta el momento, que corresponde a implementar el modelo mediante el lenguaje de programación AMPL [10] y resolver el problema mediante el *solver* comercial CPLEX [09].
- Dicha base consistirá en seleccionar, analizar e implementar un método de resolución que se adapte de manera robusta al problema abordado. Por lo tanto el presente Trabajo de Fin de Master pretende: analizar detenidamente el funcionamiento del algoritmo seleccionado, realizar pruebas de pequeña escala para verificar su correcto funcionamiento, extender dicho algoritmo al modelo eléctrico multimercado de oferta óptima, analizar el efecto de los parámetros seleccionables por el usuario en el algoritmo, analizar de los resultados obtenidos.
- Otro aspecto clave este trabajo es proponer vías futuras de cara a mejorar el rendimiento computacional de la implementación.

Métodos de obtención de la cota inferior:

- Valorar la adaptabilidad del método Analytic Center Cutting Plane Method (ACCPM, [01], [02], [03] y [04]) para resolver problemas enteros, introduciendo este método en una estrategia de exploración Branch & Bound.

- Valorar la adaptabilidad de métodos duales para la resolución del modelo multimercado, en concreto se analizará el comportamiento del *Bundle Method* [18].

Implementación:

- Obtener la cota correspondiente al problema multimercado utilizando como lenguaje de programación C++ y CPLEX [09] como algoritmo de resolución de subproblemas.
- Valorar el lenguaje comunicativo más adecuado entre C++ y CPLEX, pudiendo ser: *Callable Library*, es decir, utilizar el lenguaje de programación clásico de C++ y llamar desde aquí a las librerías CPLEX correspondientes; o por el contrario *Concert Technology*, un metalenguaje creado por la empresa ILOG para facilitar el manejo de ambas herramientas.

Resolución:

- Tanto el *ACCPM* como el *Bundle Method* exigen que el usuario determine el valor de una serie de parámetros de resolución. La convergencia del algoritmo y la calidad del resultado final dependen en gran medida de estos valores. Consecuentemente uno de los objetivos será determinar el efecto de cada parámetro en la resolución y decidir sus valores o rangos de valores óptimos.

Resultados:

- Valorar la calidad de los resultados y concluir si el tiempo computacional dedicado a la obtención de la cota inferior puede ser mejorado mediante herramientas adicionales en futuras investigaciones.

3. DESCRIPCIÓN DEL PROBLEMA

Una vez definidos los objetivos de este proyecto pasamos a describir el problema concreto que se desea resolver, es decir, el modelo eléctrico multimercado de oferta óptima.

3.1. Contexto

El proceso de integración de los sistemas eléctricos de España y Portugal culmina con la creación de una serie de mercados eléctricos liberalizados comunes denominado Mercado Ibérico de la Electricidad (MIBEL, más información en <http://www.mercadoibericoenergia.org/presentacion.html>). Se trata de un grupo de mercados en los que las compañías generadoras ofertan energía y las compañías distribuidoras realizan una demanda de energía. El precio del mercado en cada intervalo horario viene definido por la casación que realiza el operador de mercado. Esta serie de mercados secuenciados incluye el Day-Ahead Market (DAM), el Reserve Market (RM) y un conjunto de seis Intraday Markets (IM).

En este contexto las compañías generadoras que participan en el mercado eléctrico pueden aumentar sus beneficios optimizando su participación en esta secuencia de mercados. Analizando concretamente la perspectiva a corto plazo (previsión de demanda y su distribución horaria) el objetivo de la compañía generadora será maximizar los beneficios esperados de su participación en el DAM, el RM y los IM. Además la compañía tendrá que tener en cuenta sus Contratos Bilaterales (Bilateral Contracts, BC) así como los Contratos de Futuros (Futures Contracts, FC) ya acordados y no optimizables.

El artículo "Optimal Day-Ahead Bidding in the MIBEL's Multimarket Energy Production System" [05] construye un modelo matemático que proporciona a la compañía generadora la estrategia óptima de participación en el DAM teniendo en cuenta la secuencia mercados y sus BC y FC. Esta formulación será tomada como referencia, para el desarrollo del modelo del mercado del proyecto aquí descrito.

3.2. Estructura del mercado

A continuación analizamos de manera más detallada las características de los diferentes mercados secuenciales descritos anteriormente.

DAM (Day Ahead Market): Se trata del mercado principal donde se negocia la mayor parte de las transacciones. Consiste en 24 subastas simultáneas, una por cada intervalo horario del día siguiente. Por lo tanto se trata de un mercado a un día vista. La casación de la oferta y la demanda lo realiza el operador del mercado teniendo en cuenta el cumplimiento de los BC y FC.

RM (Reserve Market): Se produce inmediatamente después del proceso de casación del DAM. Es un mercado de servicios complementarios donde las compañías envían pujas para aumentar o

disminuir la energía casada de las unidades durante el DAM. La particularidad de este mercado reside en que si una puja se casa en el RM entonces la unidad debe permitir cambiar el nivel de generación en un intervalo dado en la operación a tiempo real.

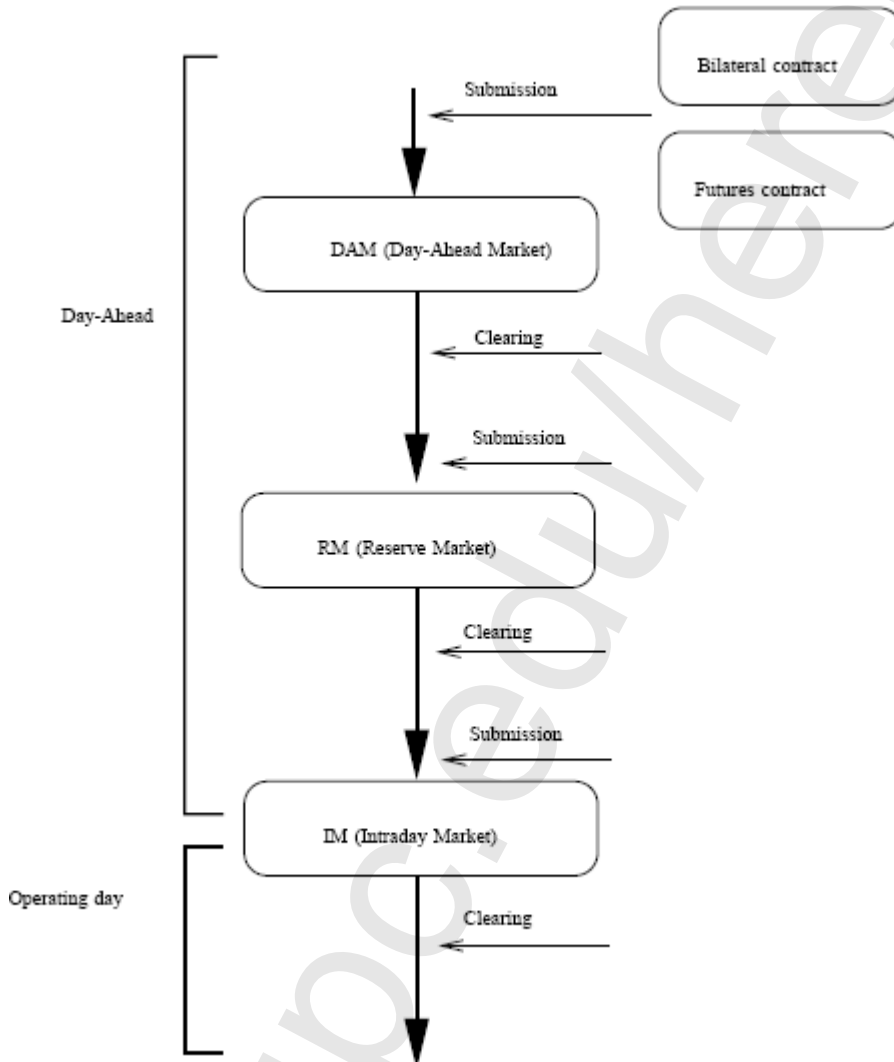


Figura 1 Esquema de la secuencia del mercado

IM (Intraday Market): Tiene lugar antes y durante el día de consumo. Se compone de 6 mercados consecutivos con 24 subastas cada uno. En este mercado las compañías generadoras pueden tanto vender como comprarse energía eléctrica entre sí. El funcionamiento es exactamente igual al DAM y tiene como objetivo que las compañías generadoras puedan cambiar la planificación resultante. Cabe destacar que en cada intervalo horario una unidad solo puede enviar ofertas de compra o de venta, no ambas, no obstante este rol puede cambiar de un intervalo horario a otro. Una unidad únicamente puede participar en este mercado si sus pujas han sido casadas en el DAM o si está produciendo energía para saldar los BC.

3.3. Hipótesis consideradas en el modelo matemático:

A la hora de modelizar el mercado eléctrico el modelo multimercado de oferta óptima asume una serie de hipótesis que habrá que tener en cuenta:

- Las unidades que participan en el RM pujarán siempre la capacidad AGC (Automatic Generation Control) característica fija de cada unidad. Por lo tanto sólo se optimizará la participación o no de cada unidad.
- Se considera únicamente la primera sesión del IM.
- Se supone que todas las pujas realizadas en los RM e IM serán casadas.

4. MODELO DE OPTIMIZACIÓN

Una vez descrito el contexto de aplicación y las hipótesis realizadas se detallará el modelo de optimización adoptado para la caracterización de mercado eléctrico.

Tal y como se verá a continuación el problema abordado es un problema de optimización estocástica multietapa mixto entero con función objetivo cuadrática:

4.1. Caracterización de la incertidumbre

Cada mercado tiene asociado un vector de precios de casación de igual dimensión al número de intervalos considerado. Los tres vectores de precios de mercado pueden ser caracterizados como variables aleatorias y pueden usarse para construir el árbol de escenarios. La probabilidad de cada escenario es igual al producto de probabilidades de cada vector de precios.

Cabe destacar que sólo hay un escenario de tercera etapa (RM) para cada escenario de segunda etapa (DAM) debido a las hipótesis realizadas sobre el mercado RM. En el RM la variable de decisión es pujar o no pujar, la cantidad pujada es un dato operacional independiente del precio RM. Por lo tanto no hay necesidad de representar el precio RM en escenarios ya que lo representamos mediante su valor esperado.

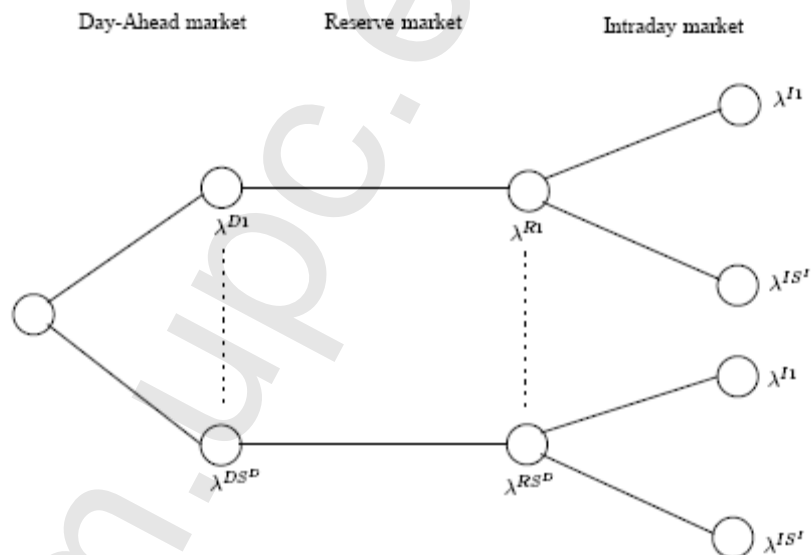


Figura 2 Árbol de escenarios

Primera etapa (elaboración de la oferta al DAM)

Las decisiones de primera etapa corresponden a la planificación de los encendidos y apagados de las unidades y la generación energética de cada unidad de cara a cubrir los BC y los FC. De estos valores (variables de primera etapa) se deriva la oferta óptima a las 24 sesiones del DAM.

Segunda etapa (casación del DAM)

El operador de mercado casa el precio de las 24 sesiones del DAM y se obtiene la energía casada de la oferta que cada unidad de generación ha sometido al DAM.

Tercera etapa (casación del RM)

Se decide qué unidades participan en el mercado de reserva y en qué intervalos. El operador de mercado casa el precio RM de cada intervalo (24 en total)

Cuarta etapa (casación del IM)

Se decide qué unidades participan en el IM y su rol como compradores o vendedores en los diferentes intervalos. El operador de mercado casa los precios de venta y de compra de cada intervalo (48 en total)

4.2. Modelo matemático asociado al problema

Para facilitar la comprensión del modelo matemático se presentan en primera instancia los elementos que tomarán parte en él:

Conjuntos e índices

T : Conjunto de intervalos.

t : Elemento del conjunto T .

I : Conjunto de unidades.

i : Elemento del conjunto I .

S : Conjunto de escenarios.

s : Elemento del conjunto S .

F : Conjunto de contratos de tipo FC.

j : Elemento del conjunto F .

U_j : Conjunto de unidades cuya generación es asignable (parcial o totalmente) a la demanda del contrato j de tipo FC.

B : Conjunto de contratos de tipo BC.

bc : Elemento del conjunto B .

Constantes

P^s : Probabilidad del escenario s .

$\lambda_t^{D,s}$: Precio de mercado de DAM en el intervalo t bajo el escenario s .

$\lambda_t^{R,s}$: Precio de mercado de RM en el intervalo t bajo el escenario s .

$\lambda_t^{I,s}$: Precio de mercado de IM en el intervalo t bajo el escenario s .

c_i^b : Coeficiente de coste de funcionamiento de la unidad i .

c_i^l : Coeficiente de coste lineal de generación de la unidad i .

c_i^q : Coeficiente de coste cuadrático de generación de la unidad i .

c_i^{on} : Coeficiente de coste de encendido de la unidad i .

c_i^{off} : Coeficiente de coste de apagado de la unidad i .

g_i : Capacidad de generación ACG de la unidad i .

\bar{P}_i : Máxima capacidad de generación de la unidad i en funcionamiento.

\underline{P}_i : Máxima capacidad de generación de la unidad i en funcionamiento.

t_i^{on} : Número de intervalos consecutivos en los que la unidad i debe permanecer activa tras el encendido.

t_i^{off} : Número de intervalos consecutivos en los que la unidad i debe permanecer inactiva tras el apagado.

M^l : Límite superior de w_{it}^s e y_{it}^s .

L_j^{FC} : Generación energética acordada en el FC j .

L_j^{BC} : Generación energética necesaria en el intervalo t para cubrir los BC.

G_i : Número de periodos iniciales durante los cuales la unidad i debe permanecer encendida.

H_i : Número de periodos iniciales durante los cuales la unidad i debe permanecer apagada.

Variables

Variables de primera etapa

$f_{ij} \geq 0$: Contribución de la unidad i en el intervalo t al contrato FC j . Esta energía tiene que ser ofrecida a precio nulo (*instrumental price*).

$b_{it} \geq 0$: Contribución de la unidad i en el intervalo t para el cumplimiento de los contratos bilaterales.

$q_{it} \geq 0$: Cantidad energética ofertada a precio nulo (*instrumental price*).

$c_{it}^u \geq 0$: Coste de encendido de la unidad i en el intervalo t .

$c_{it}^d \geq 0$: Coste de apagado de la unidad i en el intervalo t .

$u_{it} \in \{0,1\}$: Estado de la unidad de generación, 1 si está en funcionamiento, 0 si está en parada.

Variables de segunda etapa y posteriores

$p_{it}^{M,s} \geq 0$: Cantidad energética de la unidad i casada en el mercado DAM en el intervalo t bajo el escenario s .

$y_{it}^s \geq 0$: Cantidad energética comprada por la unidad i en el intervalo t bajo el escenario s .

w_{it}^s : Cantidad energética vendida o comprada por la unidad i en el intervalo t bajo el escenario s .

$p_{it}^s \geq 0$: Generación total de la unidad i en el intervalo t bajo el escenario s .

$r_{it}^s \in \{0,1\}$: Valdrá 1 si la unidad i participa en el RM en el intervalo t bajo el escenario s . Valdrá 0 en otro caso.

Función objetivo

La función objetivo corresponde a una función cuadrática que representa la maximización de los beneficios obtenidos de la participación en la secuencia de mercados.

Los ingresos provienen de la generación casada en cada mercado multiplicado por el precio de mercado correspondiente, mientras que los gastos están compuestos por gastos fijos de encendido, apagado y funcionamiento, más un coste cuadrático que caracteriza el gasto de generación de la cantidad energética ofertada en cada mercado.

$$\max_{p,q,f,b} \sum_{t \in T} \sum_{i \in I} \left\{ -c_{it}^u - c_{it}^d - c_i^b u_{it} + \sum_{s \in S} P^s \left[\lambda_t^{D,s} p_{it}^{M,s} + \lambda_t^{R,s} r_{it}^s g_i + \lambda_t^{L,s} w_{it}^s - \left(c_i^l p_{it}^s + c_i^q (p_{it}^s)^2 \right) \right] \right\} \quad (01)$$

En esta expresión no se han incluido los ingresos derivados de los contratos bilaterales y de futuros, cuya expresión es:

$$\sum_{t \in T} \left(\sum_{bc \in BC} \lambda_{bc}^{BC} L_{bc,t}^{BC} + \sum_{j \in F} (\lambda_j^{FC} - \bar{\lambda}_t^{D,s}) L_j^{FC} \right)$$

Donde $\bar{\lambda}_t^{D,s}$ corresponde al valor promedio de los precios de mercado en el periodo t . La anterior expresión no depende de las variables de decisión del problema, y por lo tanto puede excluirse de la función objetivo a maximizar.

Restricciones

La participación en los diferentes mercados está sometida a un total de 29 familias de restricciones. Entre ellas encontramos restricciones referidas a:

- Obligado cumplimiento de los BC y FC.
- Restricciones de oferta máxima y mínima por unidad en el DAM.
- Decisión de participar o no participar en el RM por unidad.
- Elección de compra o venta de energía en el IM por unidad.
- Restricciones de generación total.
- Restricciones de apagados y encendidos de las unidades.
- Restricciones de no anticipatividad de escenarios (propios de modelos estocásticos multietapa).

La expresión matemática de dichas restricciones corresponde a:

Restricciones que enlazan el funcionamiento de las unidades en conjunto

El cumplimiento de los Bilateral Contracts y los Future Contracts obliga a que se cumplan las siguientes condiciones:

$$\sum_{i \in U_j} f_{ij} = L_j^{FC} \quad j \in F, t \in T, i \in I \quad (02)$$

$$\sum_{i \in I} b_{it} = \sum_{bc \in BC} L_{bc,t}^{BC} \quad t \in T, i \in I \quad (03)$$

Restricciones individuales

Para cada unidad $i \in I$ se tienen que cumplir las siguientes restricciones: (desde (04) hasta (26))

$$f_{ij} \geq 0 \quad j \in F, t \in T \quad (04)$$

$$0 \leq b_{it} \leq \bar{P}_i \quad t \in T \quad (05)$$

El Day-Ahead Market implica las siguientes restricciones:

$$p_{it}^{M,s} \leq \bar{P}_i u_{it} - b_{it} \quad t \in T, s \in S \quad (06)$$

$$p_{it}^{M,s} \geq q_{it} \quad t \in T, s \in S \quad (07)$$

$$q_{it} \geq \underline{P}_i u_{it} - b_{it} \quad t \in T \quad (08)$$

$$q_{it} \geq \sum_{j \in F} f_{ij} \quad t \in T \quad (09)$$

$$q_{it} \geq 0 \quad t \in T \quad (10)$$

La decisión de participar o no en el Reserve Market se modelizan mediante las siguientes ecuaciones:

$$p_{it}^s - p_{i,(t-1)}^s \leq (1 - r_{it}^s) R_i \quad t \in T, s \in S \quad (11)$$

$$p_{it}^s - p_{i,(t-1)}^s \geq (1 - r_{it}^s) (-R_i) \quad t \in T, s \in S \quad (12)$$

A su vez existen restricciones de participación entre mercados:

$$p_{it}^s = b_{it} + p_{it}^{M,s} + w_{it}^s \quad t \in T, s \in S \quad (13)$$

$$\underline{P}_i u_{it} + g_i r_{it}^s \leq p_{it}^s \leq \bar{P}_i u_{it} - g_i r_{it}^s \quad t \in T, s \in S \quad (14)$$

$$r_{it}^s \leq u_{it} \quad t \in T, s \in S \quad (15)$$

Las decisiones de primera etapa referidas a la conmutación de las unidades deben cumplir que:

$$c_{it}^u \geq c_i^{on} [u_{it} - u_{i,(t-1)}] \quad t \in T \setminus \{1\} \quad (16)$$

$$c_{it}^d \geq c_i^{off} [u_{i,(t-1)} - u_{it}] \quad t \in T \setminus \{1\} \quad (17)$$

$$c_{it}^u, c_{it}^d \geq 0 \quad t \in T \quad (18)$$

$$\sum_{j=1}^{G_i} (1 - u_{ij}) = 0 \quad (19)$$

$$\sum_{j=1}^{H_i} u_{ij} = 0 \quad (20)$$

$$\sum_{n=t}^{t+t_i^{on}-1} u_{in} \geq t_i^{on} [u_{it} - u_{i,(t-1)}] \quad t = G_i + 1, \dots, |T| - t_i^{on} + 1 \quad (21)$$

$$\sum_{n=t}^{t+t_i^{off}-1} (1 - u_{in}) \geq t_i^{off} [u_{i,(t-1)} - u_{it}] \quad t = H_i + 1, \dots, |T| - t_i^{off} + 1 \quad (22)$$

$$\sum_{n=t}^{|T|} (u_{in} - [u_{it} - u_{i,(t-1)}]) \geq 0 \quad t = |T| - t_i^{on} + 2, \dots, |T| \quad (23)$$

$$\sum_{n=t}^{|T|} (1 - u_{in} - [u_{it} - u_{i,(t-1)}]) \geq 0 \quad t = |T| - t_i^{off} + 2, \dots, |T| \quad (24)$$

Teniendo en cuenta que el problema es estocástico multietapa debemos añadir restricciones de no anticipatividad de escenarios

$$(DAM): p_{it}^s = p_{it}^{\hat{s}} \quad \forall s, \hat{s} : (\lambda^{D_s}, \lambda^{D_{\hat{s}}}) \quad \forall t \in T \quad (25)$$

$$(RM): r_{it}^s = r_{it}^{\hat{s}} \quad \forall s, \hat{s} : ((\lambda^{D_s}, \lambda^{R_s}) = (\lambda^{D_{\hat{s}}}, \lambda^{R_{\hat{s}}})) \quad \forall t \in T \quad (26)$$

Analizando la formulación del modelo podemos observar que cada unidad tiene asociado el grupo de restricciones (04) a (26). Así, únicamente las restricciones (02) y (03) ligan el comportamiento de las centrales entre sí. Tal y como se explicará en capítulos posteriores; esta propiedad permitirá, tras haber realizado las transformaciones pertinentes, la división del problema original en subproblemas.

De cara a compactar la notación, se define un conjunto que contiene las restricciones (04) a (26) asociadas a la unidad i :

$$\tau_i = \{(M04) \dots (M26)\} \quad (27)$$

4.3. Evolución del modelo multimercado de oferta óptima

El modelo mostrado en las líneas precedentes corresponde al modelo final adoptado para la caracterización de la secuencia de mercados del MIBEL. El presente proyecto tomó como base el modelo inicial creado por los miembros de *GNOM*, el cual fue implementado en las primeras fases del proyecto.

No obstante y gracias a la aportación de diferentes personas y a la ayuda de los resultados obtenidos con *AMPL* y *C++*, se pudieron corregir algunos errores de modelización y se pudo simplificar el modelo prescindiendo un gran grupo de variables binarias.

Si bien este proceso ralentizó en parte el desarrollo continuo del *Trabajo de Fin de Master* el resultado final arroja un modelo más riguroso y sencillo que el considerado inicialmente. Consecuentemente el presente trabajo se sustenta sobre una base más firme.

Otro de los aspectos positivos de este proceso ha sido constatar que las implementaciones realizadas en este trabajo han sido de la ayuda al *GNOM* a la hora de detectar algunos errores de modelización.

5. MÉTODO DE RESOLUCIÓN: ACCPM

5.1. Interés

Tal y como se ha mencionado anteriormente el problema analizado ha sido resuelto mediante el software comercial CPLEX sobre AMPL [10] y en un trabajo que se está realizando en paralelo dentro del mismo proyecto de investigación en la Universidad del País Vasco, por el método de *Perspective Cuts*. Por tanto el interés de este trabajo es ver si el ACCPM resuelve satisfactoriamente el problema y si mejora el rendimiento ofrecido por los dos métodos alternativos mencionados.

Desde un punto de vista teórico la aplicación del ACCPM a nuestro problema abarca una gran cantidad de aspectos de la teoría de optimización. Así el ACCPM tiene una estructura de resolución típica de los métodos de descomposición de la función original (como por ejemplo el método de Benders, muy útil para problemas estocásticos lineales) y realiza una exploración de la región factible semejante a los métodos de punto interior (ya que el centro analítico no coincidirá en general con un vértice). Además el problema abordado enriquece aun más el contexto de optimización ya que se tendrá que adaptar el método a un árbol de exploración de programación entera.

5.2. Aplicaciones existentes de ACCPM

Consultando la literatura sobre resoluciones de problemas mediante ACCPM se observa que es un método muy eficiente para la resolución de problemas de gran escala. Los trabajos [11] y [12] estudian la resolución de problemas de flujos multiartículo no lineales de hasta 5000 arcos y 10000 artículos. En este contexto [13] concluye que comparativamente con otros métodos el ACCPM no es siempre el más rápido pero presenta un buen comportamiento de manera constante y es de lejos el más estable.

La descomposición del problema original que realiza el ACCPM crea una estructura muy adecuada para resolver problemas estocásticos. El trabajo [14] aborda la manera de resolver este tipo de problemas de manera general, mientras que [15] presenta una aplicación particular del ámbito de la planificación de portafolio donde se consigue resolver un problema de un millón de escenarios.

A día de hoy no existe una amplia literatura referida a la aplicación de ACCPM a programación entera. No obstante encontramos el trabajo [16] donde se aborda la resolución de un problema entero reformulado mediante relajación Lagrangiana. Los autores concluyen que el ACCPM es una potente alternativa al método de planos de corte de Kelly y a la optimización subgradiente cuando se tiene que resolver iterativamente un problema no diferenciable (por ejemplo en una exploración Branch & Bound).

Se espera que este método permita resolver de manera relativamente rápida y robusta el problema objetivo incluso con una gran cantidad de escenarios.

5.3. Analytic Center Cutting Plane Method

El ACCPM es un método de resolución de problemas convexos mediante cortes sucesivos. Es posible aplicar el ACCPM a cualquier problema que tenga la siguiente forma:

$$\min \left\{ f(x) = f_0(x) + \sum_{i=1}^p f_i(x) \mid x \in X = X_0 \cap_{i=1}^r X_i \right\}$$

Hipótesis 1: Los conjuntos $X_t \subset R^n$, $t = 0, \dots, r$ son convexos. Las funciones $f_j : R^n \rightarrow R$, $j = 1, \dots, p$ son convexas.

Hipótesis 2: La función f_0 es lineal y el conjunto X_0 es un poliedro acotado definido por las inecuaciones lineales $\langle B, x \rangle \leq b$

Hipótesis 3: El oráculo describe las funciones $f_j(x)$, $j = 1, \dots, p$ y los conjuntos X_t , $t = 1, \dots, r$.

La idea principal es resolver iterativamente un problema de programación lineal cuya solución es asociada a un corte de la región factible actual. De esta manera la caracterización de la región factible será cada vez más completa hasta que la solución del problema de programación lineal corresponda a la solución del problema original.

En este contexto se identifican tres elementos importantes en la resolución mediante el método ACCPM: el cálculo del centro analítico, la generación de los cortes y la caracterización de la región factible.

Cálculo del centro analítico

El centro analítico de un poliedro formado por el conjunto de puntos que cumplen $\{x \mid A^T x \leq c\}$ corresponde a la única solución (si existe) del siguiente problema de optimización:

$$\min \left\{ - \sum_{i=1}^K \log s_i \mid s = c - A^T x > 0 \right\}$$

Si realizamos la siguiente hipótesis:

Hipótesis 4: El poliedro $\{x \mid A^T x \leq c\}$ está acotado y es un conjunto no vacío.

El centro analítico es también la única solución del sistema de ecuaciones no lineales correspondientes a las condiciones de optimalidad de primer orden:

$$\begin{aligned} A^T x + s &= c \\ A\lambda &= 0 \end{aligned}$$

$$Ys = e$$

Donde Y es la matriz diagonal cuya diagonal principal es el vector dual y . Así, podemos calcular el centro analítico del polítopo dual $\{A\lambda = 0, \lambda \geq 0\}$ como la solución única de:

$$\min \left\{ \langle c, \lambda \rangle - \sum_{i=1}^K \log \lambda_i \mid A\lambda = 0, \lambda > 0 \right\}$$

Oráculo

Se llama oráculo al problema de optimización definido por el usuario con el objetivo de crear un corte en la región factible. La formulación del problema queda en manos del criterio del usuario, teniendo siempre en cuenta una serie de consideraciones:

El dato de entrada del oráculo será el centro analítico de la región factible actual. Mientras que los datos de salida corresponderá a la información local del problema en el punto de análisis: valor de la función objetivo $f(\tilde{x})$, el subgradiente $\partial f(\tilde{x})$ y la Hessiana $\nabla^2 f(\tilde{x})$.

Para cada punto $\tilde{x} \in X_0$ el oráculo dará como salida una de las siguientes informaciones:

Corte de factibilidad:

Para $t \in \{1, \dots, r\}$ tal que $\tilde{x} \notin X_t$, el oráculo devuelve el vector $(\gamma_0, \gamma) \in R \times R^n$ y el corte de factibilidad:

$$\langle \gamma, x - \tilde{x} \rangle + \gamma_0 \leq 0 \text{ para todo } x \in X_t$$

Corte de optimalidad:

Si el punto es factible $\tilde{x} \in X$, el oráculo devuelve p valores de la función objetivo $f_j(\tilde{x})$ y p subgradientes $\gamma_j \in \partial f_j(\tilde{x})$ que definen la siguiente inecuación:

$$f_j(x) \geq f_j(\tilde{x}) + \langle \gamma_j, x - \tilde{x} \rangle \text{ para todo } x \in X_t, j = 1, \dots, p$$

Región factible

Sea (x^1, \dots, x^K) una secuencia de centros analíticos, donde K está dividida en dos conjuntos $I_K \cup J_K$ donde:

$$I_K = \{k \mid x^k \text{ es infactible (corte de factibilidad)}\}$$

$$J_K = \{k \mid x^k \text{ es factible (corte de optimalidad)}\}$$

Si $J_K \neq \emptyset$ se puede definir una cota superior del problema:

$$\bar{\theta}_K = \min\{f(x^k) \mid k \in J_K\}$$

Teniendo en cuenta las inecuaciones válidas generadas hasta el momento podemos caracterizar la región factible L_k como el conjunto de puntos que cumplan:

$$z_i \geq f_j(x^k) + \langle \gamma_j^k, x - x^k \rangle \text{ para todo } j = 1, \dots, p \text{ y } k \in J_K$$

$$0 \geq \gamma_0^k + \langle \gamma^k, x - x^k \rangle \text{ para todo } k \in J_K$$

$$\bar{\theta}_K \geq z_1 + \dots + z_p$$

$$b \geq \langle B, x \rangle$$

Finalmente podemos resumir el método de resolución de manera esquemática de la siguiente manera:

Mientras $\bar{z} - \underline{z} > \varepsilon$

- 1.- Calcular \tilde{x} , centro analítico de L_k
- 2.- Calcular \underline{z} , cota inferior del valor óptimo
- 3.- Ejecutar el oráculo en el punto \tilde{x} . Se obtiene:
 - a) Un corte de factibilidad
 - b) o bien un corte de optimalidad y \bar{z} cota superior del valor óptimo.
- 4.- Añadir los cortes a la definición de la región factible L_k

Fin mientras

5.4. Valoración del ACCPM como método de resolución

El planteamiento inicial correspondía a resolver el modelo multimercado mediante el método ACCPM. Esta idea nace de la lectura de diversos artículos científicos en los que se pone de manifiesto el buen rendimiento de dicho método para resolver problemas de gran escala, especialmente en el contexto estocástico, tal y como se ha señalado anteriormente.

Se procedió al análisis en profundidad del método y a la comprensión de los diferentes aspectos del mismo. Se comprobó que las aplicaciones prácticas descritas en la literatura científica correspondían en su casi totalidad a problemas de programación lineal. Faltaba por valorar por tanto si el rendimiento del método mantenía su calidad en problemas de programación lineal mixta, y en este caso además con función objetivo cuadrática.

Tras ello se trazó un esquema general de resolución, en donde el método ACCPM estaría integrado dentro de una búsqueda de tipo *Branch & Bound*. Tal y como se ha visto en el capítulo descriptivo de este método, el método ACCPM no es un algoritmo estático válido para todo problema, sino que el subproblema interno del algoritmo debe ser adaptado a la aplicación en

concreto. La adecuada selección de la tipología del subproblema es de gran importancia y requiere de un profundo análisis y la adopción y testado de las posibilidades consideradas.

Al analizar el encaje del problema abordado y el método ACCPM se observan varios aspectos que invitan a pensar que el método no se adaptaría satisfactoriamente al problema abordado:

El elevado número de variables enteras y binarias hace que el árbol de exploración Brach & Bound sea de grandes dimensiones. Además resulta difícil ajustar cotas inferiores al problema mediante la relajación del problema original, ya que el valor óptimo estará en general muy alejado de la solución obtenida y además la recuperación de factibilidad de manera aceptablemente buena resultaría difícil y exigiría la programación de métodos heurísticos. Esto hace pensar que el número de nodos a explorar será grande y por tanto el número de iteraciones será importante.

Además el gran número de variables a relajar hacen intuir que el cálculo del centro analítico de la caracterización del problema relajado diste de manera significativa del centro analítico real. Por tanto la generación de cortes sucesivos puede no ser muy ajustada.

El cálculo del centro analítico de la región considerada es la parte más costosa algorítmicamente del método. En la literatura consultada este aspecto se compensaba mediante oráculos bien definidos que aportaban los diferentes cortes de manera rápida. En este caso, y considerando los dos aspectos mencionados en los párrafos anteriores, se espera que el número de iteraciones sea alto y que la caracterización del problema sea pobre.

Tras identificar estos aspectos se procede a contrastar estas ideas con otros expertos en el área que hayan trabajado anteriormente con el ACCPM de manera extensa de cara a tener una segunda opinión respecto a la valoración realizada. Confirmando las ideas planteadas en el análisis realizado, valoran que no hay evidencia empírica de que el método ACCPM es más eficiente que los métodos duales de tipo *Bundle* a la hora de resolver problemas enteros. No obstante la complejidad algorítmica del ACCPM hace que su implementación computacional se significativamente más costosa que la correspondiente a los métodos de tipo *Bundle*.

Consecuentemente se decide considerar los métodos *Bundle* como algoritmo de resolución.

6. MÉTODO DE RESOLUCIÓN: *PROXIMAL BUNDLE METHOD*

Tras descartar la utilización del ACCPM se procede a valorar otro método de la familia de métodos de resolución duales. En concreto se decide analizar las diferentes estrategias de resolución del dual Lagrangiano que se hayan adoptado en el ámbito de la optimización de la generación eléctrica.

En la literatura científica referida a la planificación de la generación energética podemos encontrar diferentes métodos de optimización no diferenciable propuestos para solucionar el dual Lagrangiano. La correcta elección del método es una decisión crítica tanto en términos de computación de la cota inferior como de calidad de la solución primal obtenida. Las primeras aproximación se basaron en métodos de subgradiente [20], [21], los cuales requieren poco esfuerzo computacional por iteración pero en cambio presentan una mala convergencia.

Más recientemente, se han adoptado estrategias de relajación lagrangiana basados en restricciones dinámicas [22] y métodos de corte de punto interior [23], y métodos Bundle en todas su variantes: agregado, desagregado [24][25] y los de complejidad reducida [26]. Otra manera de solventar la no diferenciable de la función Lagrangiana corresponde al uso de estrategias de Lagrangianos Aumentados (Augmented Lagrangian, AL) [27][28]. No obstante en ese el AL no permite la separación por unidades de generación, cosa que sí sucede en la solución finalmente adoptada. Por último señalar que los modelos Bundle generalizados son aproximaciones de los AL [29] [18].

Tras valorar las diferentes opciones se opta por la adopción de un método *Bundle*, ya que en [17] se demuestra que este método proporciona muy buenos resultados, que además se alcanzan tras un reducido número de iteraciones, para el problema de optimización de la generación eléctrica en un contexto unimercado y de escenario determinista. Si bien la escala del problema a abordar es significativamente mayor al descrito en el artículo, se espera que el rendimiento algorítmico mantenga su eficacia.

El algoritmo utilizado por A.Borghetti et al. en [17] corresponde al algoritmo Bundle proximal descrito por C. Lemaréchal y J.B. Hiriart-Urruty en su libro "Convex Analysis and Minimization Algorithms" [18]. En el siguiente punto se describen detalladamente los diferentes aspectos de dicho algoritmo.

6.1. Algoritmo *Proximal Bundle Method*

Los métodos *Bundle* se caracterizan por guardar en grupos o paquetes la principal información referida a las últimas iteraciones, es decir, el valor de la función objetivo, el valor de los subgradientes, el valor de las variables duales etc. Los algoritmos de tipo *Bundle* utilizan esta información para construir una serie de cortes que aproximarán la función objetivo mediante un poliedro; la

particularidad de estos cortes estriba en que al coincidir con el subgradiente del punto analizado son tangentes a la función objetivo.

Esta familia de algoritmos difiere internamente en el método de obtención de cortes así como la información almacenada y su actualización iterativa. En concreto, el algoritmo de *Proximal Bundle (PB)*, descrito en el texto de Urruty y Lemarechal (capítulo XV, sección 3 de [18]), guarda en cada iteración el valor del subgradiente asociado a las variables obtenidas, así como el error producido en ese punto al linealizar la función objetivo mediante el poliedro. La particularidad de dicho algoritmo dentro de la familia de algoritmos de tipo *Bundle* es que penaliza la obtención de valores de variables significativamente alejados respecto a los relativos a la mejor solución conocida.

Para poder interpretar los pasos del algoritmo definimos a continuación los elementos que lo componen:

PROBLEMA PRINCIPAL:

Se trata de un problema convexo de minimización que puede estar sujeto a una serie de restricciones lineales de igualdad o desigualdad. El objetivo último del algoritmo *PB* será obtener la solución óptima de dicho problema. Siendo:

μ : Variable del problema principal.

$\psi(\mu) = \min_{\mu} f(\mu)$	(28)
sometido a: μ libre	

SUBPROBLEMA PROXIMAL BUNDLE:

La obtención de nuevas variables duales en cada iteración corresponde a la resolución de un subproblema. Como vemos se trata de un problema de programación lineal con función cuadrática y restricciones lineales. Siendo:

$\overline{\mu}^k$: Valor de la variable del problema principal en la mejor solución conocida hasta la iteración k .

μ^k : Valor de la variable del problema original a obtener en la iteración k .

$s(\mu)$: Valor del subgradiente en el punto μ .

t^k : Valor de la longitud de paso del subproblema en la iteración k .

β : Conjunto de cortes del *Bundle*.

e^j : Error de linealización del elemento j del *Bundle*.

r : Variable auxiliar.

ϕ^k : Función poliédrica de aproximación a la función $f(\mu)$ en la iteración k .

$$\min_{\mu^k, r^k} \left\{ r + \frac{1}{2 \cdot t^k} \|\mu^k - \bar{\mu}^k\|^2 \right\} \quad (29)$$

sometido a: $r \geq \psi(\bar{\mu}^k) - e^j + s(\mu^j) \cdot (\mu^k - \bar{\mu}^k) \quad j \in \beta$

El valor r posicionará la parábola de tal manera que está será tangente respecto al poliedro actual φ^k en el punto μ^k , es decir $\varphi^k(\mu^k) = r$.

La mecánica de la relación entre ambos problemas y su procedimiento iterativo vienen descritos en el esquema general del algoritmo PB:

ALGORITMO:

\hat{s} : Valor del subgradiente de la restricción activa en la última resolución de (29).

\hat{e} : Error de linealización de la restricción activa en la última resolución de (29).

PASO 0: inicialización

Paso 0.01: Seleccionar punto inicial μ_1

Paso 0.02: Seleccionar tolerancia de parada $\underline{\delta} \geq 0$

Paso 0.03: Seleccionar tamaño máximo del Bundle $\bar{\beta}$

Paso 0.04: Seleccionar coeficiente de descenso $m \in]0,1[$

Paso 0.05: Inicializar conjunto de descenso $K = \emptyset$

Paso 0.06: Inicializar contador de iteraciones $k = 1$

Paso 0.07: Inicializar tamaño del Bundle $\beta = 1$

Paso 0.08: Calcular $\psi(\mu_1)$

Paso 0.09: Calcular $s_1 = s(\mu_1)$

Paso 0.10: Calcular $e_1 = 0$

Por lo tanto el Bundle inicial es: (s_1, e_1)

Si conocemos el valor de μ^k se deduce que el valor de r será:

$$\mu^k \mapsto \varphi(\mu^k) := \psi(\mu^1) + s(\mu^1) \cdot (\mu^k - \mu_1) \quad (30)$$

Paso 0.11: Fijar una cota superior ajustada del problema UB

Paso 0.12: Seleccionar la longitud de paso inicial $\alpha^1 > 0$

Se sigue el criterio propuesto por Lemaréchal en la Nota 3.1.5 de la página 305 de [18].

$$t^1 = \frac{2(\psi(\mu_1) - UB)}{\|s_1\|^2} \quad (31)$$

PASO 1: Computación principal y test de parada**Paso 1.01:** Elegir una longitud de paso $t^k > 0$ **Paso 1.02:** Resolver el problema de optimización (29)Su única solución será: $\mu^{k+1} = \overline{\mu^k} - t^k \cdot \hat{s}^k$ donde $\hat{s}^k \in \partial \varphi^k(\mu^{k+1})$ **Paso 1.03:** Fijar $\hat{e}^k := \psi(\overline{\mu^k}) - \psi^k(\mu^{k+1}) - t^k \langle \hat{s}^k, \hat{s}^k \rangle$ (32)**Paso 1.04:** Fijar $\delta^k := \psi(\overline{\mu^k}) - \varphi^k(\mu^{k+1}) - \frac{1}{2} t^k \|\hat{s}^k\|^2$ (33)**Paso 1.05:** Si $\delta^k \leq \underline{\delta}$ STOP**PASO 2: Test de descenso****Paso 2.01:** Calcular $\psi(\mu^{k+1})$ y $s(\mu^{k+1})$ **Paso 2.02:** Si no se cumple la condición de descenso $\psi(\mu^{k+1}) \leq \psi(\overline{\mu^k}) - m\delta^k$ declarar el paso como “nulo” e ir al PASO 4.**PASO 3: Paso de descenso****Paso 3.01:** Fijar $\overline{\mu^{k+1}} := \mu^{k+1}$ **Paso 3.02:** Añadir k al conjunto K.**Paso 3.03:** Para $j \in \beta$ cambiar e^j por

$$e^j + \psi(\overline{\mu^{k+1}}) - \psi(\overline{\mu^k}) - \langle s^j, \overline{\mu^{k+1}} - \overline{\mu^k} \rangle$$

cambiar \hat{e}^k de manera similar

$$\hat{e}^k + \psi(\overline{\mu^{k+1}}) - \psi(\overline{\mu^k}) - \langle \hat{s}^k, \overline{\mu^{k+1}} - \overline{\mu^k} \rangle$$

PASO 4: Gestión del tamaño del *Bundle***Paso 4.01:** Si $\beta = \overline{\beta}$ entonces:Eliminar al menos 2 elementos del *Bundle* e insertar el elemento (\hat{s}^k, \hat{e}^k) **Paso 4.02:** Renombrar $(s^j, e^j)_{j \in \beta}$ el nuevo *Bundle* obtenido (se cumple $\beta < \overline{\beta}$)**PASO 5: Adición el elemento adicional al *Bundle*****Paso 5.01:** Añadir (s^{j+1}, e^{j+1}) al *Bundle*, donde $e^{j+1} = 0$ en caso de paso de descenso

$$e^{j+1} = \psi(\overline{\mu^k}) - \left[\psi(\mu^{k+1}) + \langle s^{j+1}, \overline{\mu^k} - \mu^{k+1} \rangle \right] \quad (34)$$

Paso 5.02: Reemplazar j por j+1 y definir el modelo:

$$\mu^k \mapsto \varphi(\mu^k) = \psi(\overline{\mu^{k+1}}) + \max_{j \in \beta} \left[-e^j + \langle s^j, \mu^k - \overline{\mu^{k+1}} \rangle \right] \quad (35)$$

Paso 5.03: Reemplazar k por k+1 e ir al PASO 1

El algoritmo *PB* se basa en una serie de propiedades y relaciones geométricas de los elementos descritos. Si bien su definición formal ayuda a conocer la naturaleza matemática de cada elemento, la mejor forma de analizar los diferentes pasos es mediante la resolución de un pequeño problema y el graficar las diferentes soluciones obtenidas.

6.2. Ejemplo de resolución mediante el algoritmo *PB*

Para ilustrar los pasos del algoritmo *PB* definimos el problema de optimización correspondiente a minimizar la siguiente función objetivo lineal sin restricciones:

$$f(x) = 3 - x + \frac{x^2}{6} \quad (36)$$

El resultado óptimo puede obtenerse fácilmente a través de las condiciones de optimalidad de primer y segundo orden:

$$f'(x) = -1 + \frac{x}{3} \rightarrow f'(x) = 0 = -1 + \frac{x}{3} \rightarrow x^* = 3 \quad (37)$$

$$f''(x) = \frac{1}{3} > 0 \text{ mínimo}$$

$$f(x^*) = 3 - 3 + \frac{3^2}{6} = \frac{3}{2}$$

Por lo tanto se observa que el valor óptimo de la función objetivo es igual $\frac{3}{2}$ y se obtiene cuando el valor de la variable es igual a 3. Demostraremos en las siguientes líneas que el algoritmo *PB* converge efectivamente a dichos valores.

ALGORITMO PB

En las siguientes líneas se realizarán tres iteraciones completas del algoritmo PB. Para su correcta interpretación conviene tener en cuenta los siguientes aspectos:

- La longitud de paso se multiplicará por un factor $\frac{1}{2}$ en cada iteración independientemente del tipo de paso realizado.
- Notación: subíndices indica el número de elemento mientras que los supraíndices indican la iteración a la que corresponde dicho valor.

INICIALIZACIÓN:

$$\bar{x} = 0 \quad e_1 = 0 \quad \underline{\delta} = 10^{-4} \quad \bar{l} = 3 \quad m = \frac{1}{2}$$

$$f(\bar{x}) = 3 - 0 + \frac{0^2}{6} = -3 \quad s(\bar{x}) = z_1 = -1 + \frac{0}{3} = -1 \quad t^1 = 2$$

$$B^0 = \begin{pmatrix} z & e \\ -1 & 0 \end{pmatrix} 1$$

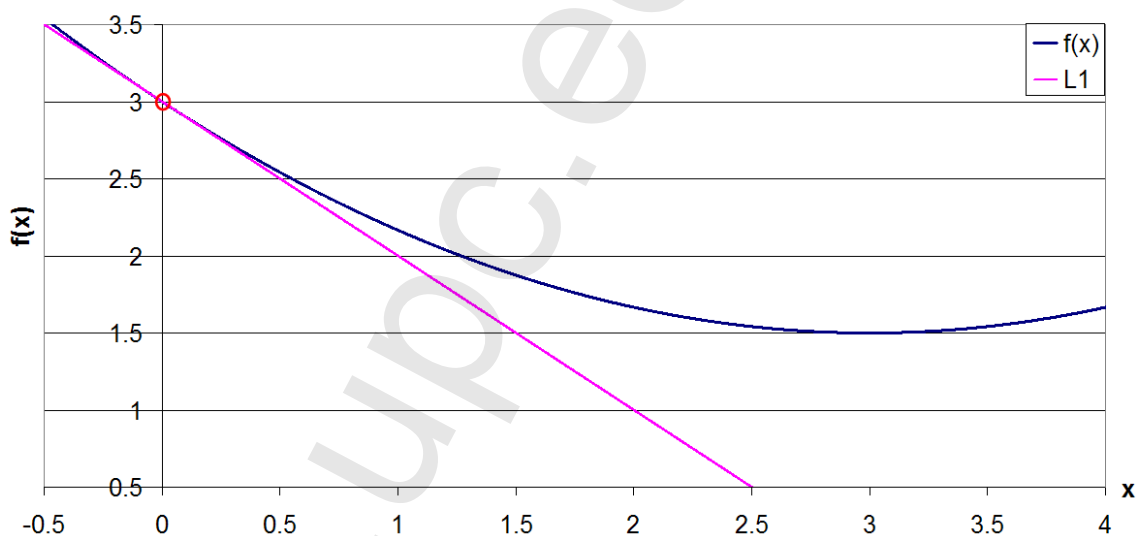


Figura 3 Algoritmo PB, inicialización: subgradiente asociado a \bar{x}

ITERACIÓN 1:

Paso1

Siendo la longitud de paso $t^1 = 2$ el subproblema de optimización PB tendrá la siguiente forma:

$$\min r + \frac{1}{2 \cdot t^1} \|y - \bar{x}\|^2 = \min r + \frac{y^2}{4} \quad \rightarrow \quad \Psi(r, y) := r + \frac{y^2}{4}$$

$$s.t \quad r \geq f(\bar{x}) - e_1 + s_1 \cdot (y - \bar{x})' = 3 - y \quad \rightarrow \quad \Gamma_1(r, y) := r + y - 3 \geq 0$$

Planteamos las condiciones de optimalidad necesarias de primer orden (condiciones de Karush-Kuhn-Tucker), que en este caso serán necesarias y suficientes de óptimo global al ser el problema convexo:

$$\begin{aligned} 1) \quad & \nabla \Psi + \sum_{j=1}^{|J|} \rho_j \nabla \Gamma_j = 0 \\ 2) \quad & \rho_j \Gamma_j = 0 \quad \forall j \in J \\ 3) \quad & \rho_j \leq 0 \quad \forall j \in J \end{aligned}$$

Escribiendo es sistema de ecuaciones podemos calcular la solución del subproblema de Lemaréchal:

$$1) \quad \nabla \Psi = \begin{bmatrix} 1 \\ y/2 \end{bmatrix} \quad \nabla \Gamma_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$1 + \rho_1 = 0 \quad \rightarrow \quad \rho_1 = -1$$

$$\frac{y}{2} + \rho_1 = 0 \quad \rightarrow \quad y = 2$$

$$2) \quad \rho_1 \cdot (r + y - 3) = 0 \quad r + y - 3 = 0 \quad \rightarrow \quad r = 1$$

$$3) \quad \rho_1 = -1 \leq 0$$

La restricción activa corresponde al elemento 1 por lo tanto definimos el subgradiente activo de esta solución como $\hat{s}^1 \equiv s_1$. Con la información obtenida hasta el momento podemos calcular el valor de la función poliédrica y el valor del error de linealización en el punto de análisis:

$$\varphi^1(y) := f(\bar{x}) + \hat{s}^1 \cdot (y - x_1)' = 3 + (-1) \cdot (2 - 0) = 1$$

$$\hat{e}^1 = f(\bar{x}) - \varphi^1(y) - t^1 \langle \hat{s}^1, \hat{s}^1 \rangle = 3 - 1 - 2 \cdot 1 = 0$$

Realizando el test de parada del algoritmo observamos que aún no se cumple dicha condición:

$$\delta^1 = f(\bar{x}) - \varphi^1(y) - t^1 \frac{\|\hat{s}^1\|^2}{2} = 3 - 1 - \frac{2}{2} \cdot 1 = 1 > \underline{\delta}$$

Paso2

Calculamos el valor de la función objetivo analizada en el nuevo punto obtenido y realizamos el test de descenso:

$$f(y) = 3 - 2 + \frac{2^2}{6} = \frac{5}{3}$$

condición de descenso: $f(y) \leq f(\bar{x}) - m \cdot \delta^1$

$$\frac{5}{3} \leq 3 - \frac{1}{2} \cdot 1 = \frac{5}{2} \quad \text{SE CUMPLE} \rightarrow \text{Paso de descenso}$$

Mediante los resultados obtenidos podemos definir la ecuación de la función cuadrática como:

$$q(y) = \Psi(r^*, y^*) - \frac{y^2}{4} = r^* + \frac{y^{*2}}{4} - \frac{y^2}{4} = 1 + \frac{4}{4} - \frac{y^2}{4} = 2 - \frac{y^2}{4}$$

Así mismo podemos definir la ecuación del subgradiente asociado al punto de análisis inicial:

$$l(y) = s(\bar{x}) \cdot y + f(\bar{x}) = -y + 3$$

Además, podemos plantear el cálculo de las coordenadas del punto de intersección entre ambas curvas:

$$q(y) = 2 - \frac{y^2}{4} = l(y) = -y + 3 \rightarrow -\frac{y^2}{4} + y - 1 = 0 \rightarrow \text{Solución única } y = 2$$

El valor **del punto de corte entre la parábola y el subgradiente activo corresponde a** $q(2) = l(2) = 1$, que coincide exactamente con el **valor de la variable auxiliar** r en esta iteración.

Este último aspecto no es un caso particular, sino que es una propiedad geométrica de las soluciones encontradas mediante el *Proximal Bundle Method*.

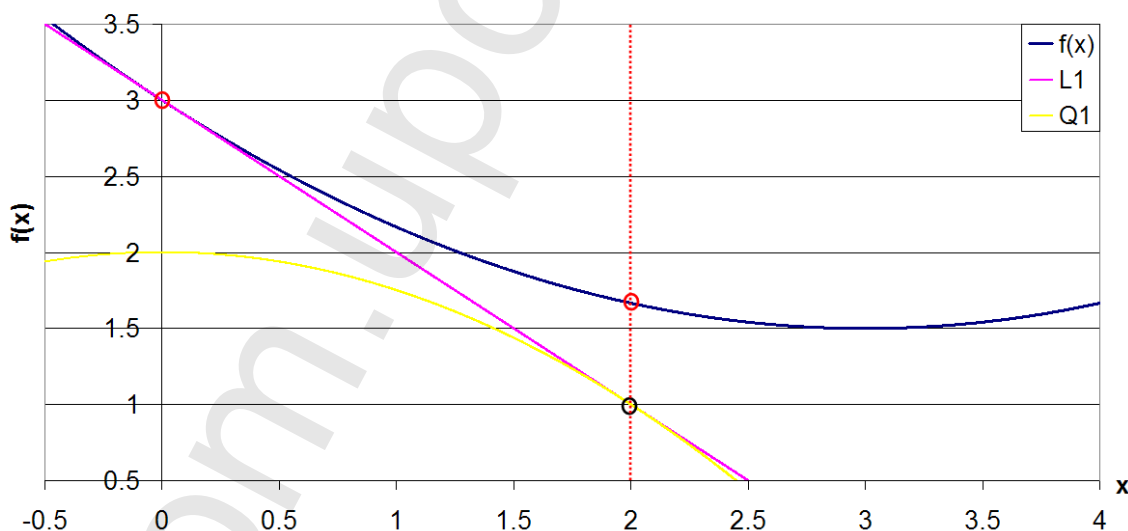


Figura 4 Algoritmo PB, iteración 1: función cuadrática asociada a y

Paso3

Actualizamos los errores de actualización trasladándolos al nuevo punto:

$$e_j \leftarrow e_j + f(y) - f(\bar{x}) - \langle s_j, y - \bar{x} \rangle \quad \forall j \in J$$

$$\hat{e}^1 \leftarrow \hat{e}^1 + f(y) - f(\bar{x}) - \langle \hat{s}^1, y - \bar{x} \rangle$$

$$e_1 = 0 + \frac{5}{3} - 3 - (-1) \cdot (2 - 0) = \frac{2}{3}$$

$$\hat{e}^1 = 0 + \frac{5}{3} - 3 - (-1) \cdot (2 - 0) = \frac{2}{3}$$

Definimos el nuevo punto obtenido en esta iteración como mejor valor conocido:

$$\bar{x} \leftarrow y$$

Paso4

Vemos que el *Bundle* no está lleno y podemos seguir añadiendo elementos:

$$l = 1 < \bar{l} = 3$$

Paso5

El paso realizado ha sido de descenso y por lo tanto hemos cambiado el punto de análisis, en consecuencia el error de linealización de nuevo punto respecto al punto de análisis es lógicamente nulo: $e_2 = 0$

Calculamos el subgradiente en el punto obtenido y añadimos la pareja (s_2, e_2) al *Bundle*:

$$s_2 = s(y) = s(2) = -1 + \frac{2}{3} = -\frac{1}{3}$$

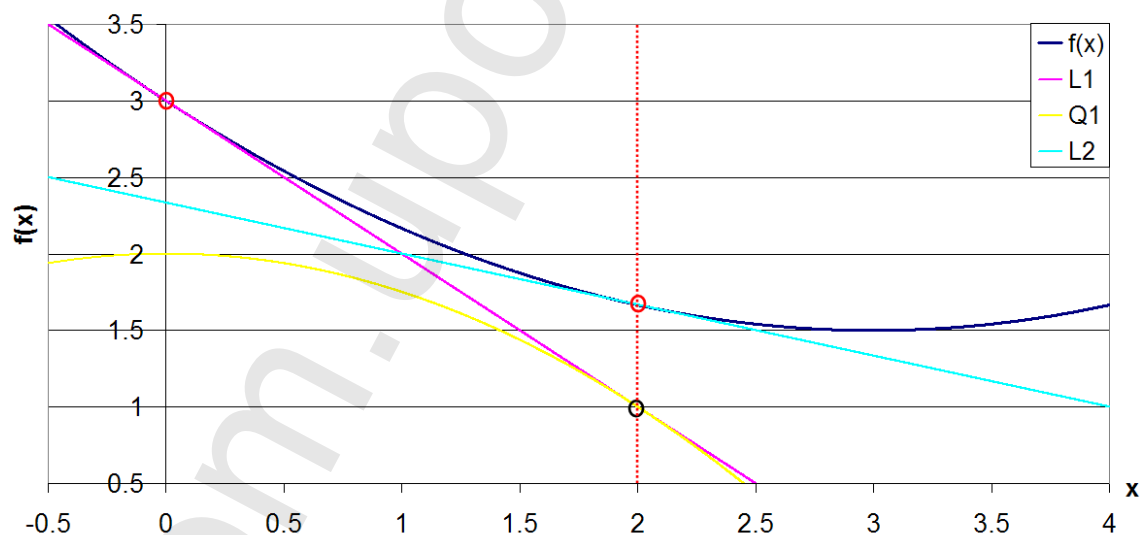


Figura 5 Algoritmo PB, iteración 1: subgradiente asociado a \bar{x}

Si ordenamos la información almacenada en el *Bundle* en una matriz obtenemos lo siguiente:

$$B^1 = \begin{pmatrix} -1 & 2/3 \\ -1/3 & 0 \end{pmatrix}$$

ITERACIÓN 2:

Paso1

Actualizamos la longitud de paso y redefinimos el subproblema de Lemaréchal:

$$t^2 = t^1/2 = 1$$

$$\min r + \frac{1}{2 \cdot t^2} \|y - \bar{x}\|^2 = \min r + \frac{(y-2)^2}{2} \rightarrow \Psi(r, y) := r + \frac{y^2}{2} - 2y + 2$$

$$s.t \quad r \geq f(\bar{x}) - e_1 + s_1 \cdot (y - \bar{x})' = \frac{5}{3} - \frac{2}{3} + (-1)(y-2)' = 3 - y$$

$$r \geq f(\bar{x}) - e_2 + s_2 \cdot (y - \bar{x})' = \frac{5}{3} - 0 + \left(-\frac{1}{3}\right)(y-2)' = -\frac{y}{3} + \frac{7}{3}$$

$$\Gamma_1(r, y) := r + y - 3 \geq 0 \quad \Gamma_2(r, y) := r + \frac{y}{3} - \frac{7}{3} \geq 0$$

Observamos que la restricción asociada al primer elemento del Bundle no ha cambiado, esto es lógico ya que el hecho de cambiar el punto de análisis no modifica las propiedades del subgradientes conocidos hasta el momento.

Volvemos a escribir las condiciones de Karush-Kuhn-Tucker y obtenemos:

$$1) \nabla \Psi = \begin{bmatrix} 1 \\ y-2 \end{bmatrix} \quad \nabla \Gamma_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \nabla \Gamma_2 = \begin{bmatrix} 1 \\ 1/3 \end{bmatrix}$$

$$1 + \rho_1 + \rho_2 = 0$$

$$y - 2 + \rho_1 + \frac{\rho_2}{3} = 0$$

$$2) \rho_1 \cdot (r + y - 3) = 0$$

$$\rho_2 \cdot \left(r + \frac{y}{3} - \frac{7}{3}\right) = 0$$

$$3) \rho_1 \leq 0$$

$$\rho_2 \leq 0$$

Para poder proceder a la resolución del sistema de ecuaciones consideramos la siguiente hipótesis: la restricción 1 es inactiva y la restricción 2

activa. Tras obtener los resultados tendremos que verificar que efectivamente la hipótesis se cumple:

$$\rho_1 = 0 \text{ (restricción inactiva)} \rightarrow \rho_2 = -1$$

$$y - 2 + \rho_1 + \rho_2 \frac{1}{3} = 0 \rightarrow y - 2 + 0 - \frac{1}{3} = 0 \rightarrow y = \frac{7}{3}$$

$$\rho_2 \cdot \left(r + \frac{y}{3} - \frac{7}{3} \right) = 0 \rightarrow r + \frac{7}{9} - \frac{7}{3} = 0 \rightarrow r = \frac{14}{9}$$

Verificamos si efectivamente la restricción 1 es inactiva:

$$r + y - 3 = \frac{14}{9} + \frac{7}{3} - 3 = \frac{8}{9} \geq 0 \quad \text{CUMPLE}$$

Por lo tanto hemos obtenido el resultado óptimo del subproblema *Proximal Bundle*, el cual indica que el subgradiente activo corresponde a la restricción 2: $\hat{s}^2 \equiv s_2$. Con la información obtenida hasta el momento podemos calcular el valor de la función poliédrica y el valor del error de linealización en el punto de análisis:

$$\varphi^2(y) := f(\bar{x}) + \hat{s}^2 \cdot (y - \bar{x})' = \frac{5}{3} + \left(-\frac{1}{3}\right) \cdot \left(\frac{7}{3} - 2\right) = \frac{14}{9}$$

$$\hat{e}^2 = f(\bar{x}) - \varphi^2(y) - t^2 \langle \hat{s}^2, \hat{s}^2 \rangle = \frac{5}{3} - \frac{14}{9} - 1 \cdot \left(-\frac{1}{3}\right)^2 = 0$$

El test de parada del algoritmo muestra que aun no se cumple la condición:

$$\delta^2 = f(\bar{x}) - \varphi^2(y) - \frac{t^2}{2} \|\hat{s}^2\|^2 = \frac{5}{3} - \frac{14}{9} - \frac{1}{2} \cdot \left(-\frac{1}{3}\right)^2 = \frac{1}{18} > \underline{\delta}$$

Paso2

Calculamos el valor de la función objetivo analizado en el nuevo punto obtenido y realizamos el test de descenso:

$$f(y) = f\left(\frac{7}{3}\right) = 3 - \frac{7}{3} + \frac{\left(\frac{7}{3}\right)^2}{6} = \frac{85}{54}$$

condición de descenso: $f(y) \leq f(\bar{x}) - m \cdot \delta^2$

$$\frac{85}{54} \leq \frac{5}{3} - \frac{1}{2} \cdot \frac{1}{18} = \frac{59}{36} \quad \text{SE CUMPLE} \rightarrow \text{Paso de descenso}$$

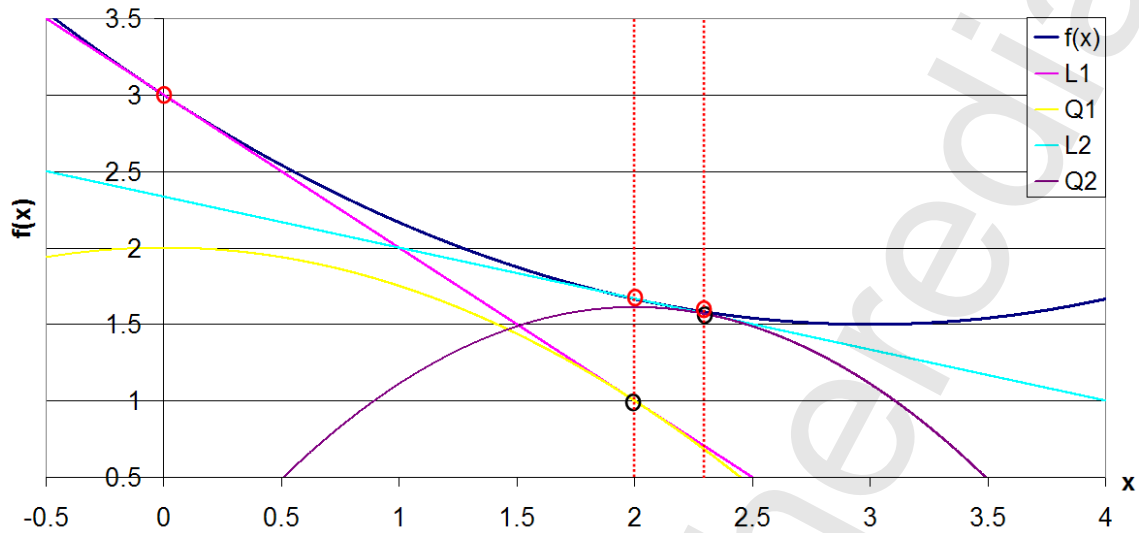


Figura 6 Algoritmo PB, iteración 2: función cuadrática asociada a y

Paso3

Actualizamos los errores de actualización trasladándolos al nuevo punto:

$$e_j \leftarrow e_j + f(y) - f(\bar{x}) - \langle s_j, y - \bar{x} \rangle \quad \forall j \in J$$

$$\hat{e}^2 \leftarrow \hat{e}^2 + f(y) - f(\bar{x}) - \langle \hat{s}^2, y - \bar{x} \rangle$$

$$e_1 = \frac{2}{3} + \frac{85}{54} - \frac{5}{3} - (-1) \cdot \left(\frac{7}{3} - 2\right) = \frac{49}{54}$$

$$e_2 = 0 + \frac{85}{54} - \frac{5}{3} - \left(-\frac{1}{3}\right) \cdot \left(\frac{7}{3} - 2\right) = \frac{1}{54}$$

$$\hat{e}^2 = 0 + \frac{85}{54} - \frac{5}{3} - \left(-\frac{1}{3}\right) \cdot \left(\frac{7}{3} - 2\right) = \frac{1}{54}$$

El nuevo punto de análisis (mejor valor conocido) pasa a ser el valor encontrado en esta segunda iteración:

$$\bar{x} \leftarrow y$$

Paso4

Vemos que el *Bundle* no está lleno y podemos seguir añadiendo elementos:

$$l = 2 < \bar{l} = 3$$

Paso5

El paso realizado ha sido de descenso y por lo tanto el error de linealización asociado es nulo: $e_3 = 0$

Calculamos el subgradiente en el punto obtenido y añadimos la pareja (s_3, e_3) al *Bundle*:

$$s_3 = s(y) = s\left(\frac{7}{3}\right) = -1 + \frac{7}{9} = -\frac{2}{9}$$

La matriz *Bundle* contiene ahora la información almacenada en la inicialización y las dos iteraciones realizadas. Obsérvese que se ha alcanzado su tamaño máximo y por tanto para poder añadir más información habrá que retirar la menos útil.

$$B^2 = \begin{pmatrix} -1 & 49/54 \\ -1/3 & 1/54 \\ -2/9 & 0 \end{pmatrix}$$

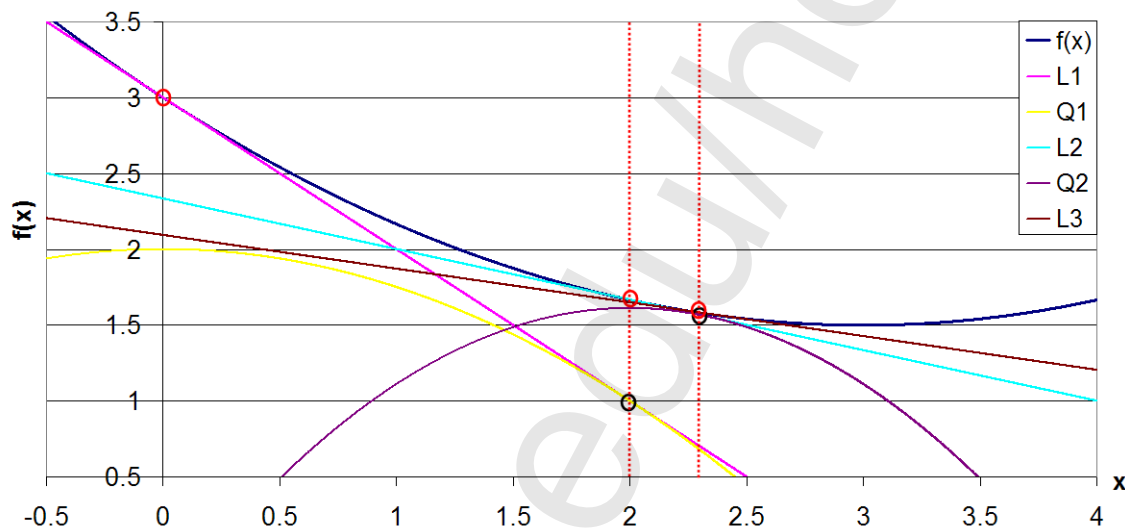


Figura 7 Algoritmo PB, iteración 2: subgradiente asociado a \bar{x}

ITERACIÓN 3:

Paso1

Actualizamos la longitud de paso y redefinimos el subproblema *Proximal Bundle*:

$$t^3 = t^2/2 = 1/2$$

$$\min r + \frac{1}{\frac{1}{2} \cdot 1} \|y - \bar{x}\|^2 = \min r + (y - \frac{7}{3})^2 \rightarrow \Psi(r, y) := r + (y - \frac{7}{3})^2$$

$$s.t. \quad r \geq 3 - y \quad \text{NO CAMBIA} \rightarrow \Gamma_1(r, y) := r + y - 3 \geq 0$$

$$r \geq -\frac{y}{3} + \frac{7}{3} \quad \text{NO CAMBIA} \rightarrow \Gamma_2(r, y) := r + \frac{y}{3} - \frac{7}{3} \geq 0$$

$$r \geq f(\bar{x}) - e_3 + s_3 \cdot (y - \bar{x}) = \frac{85}{54} - 0 + \left(-\frac{2}{9}\right)\left(y - \frac{7}{3}\right) = -\frac{2y}{9} + \frac{113}{54}$$

$$\Gamma_3(r, y) := r + \frac{2y}{9} - \frac{113}{54} \geq 0$$

Volvemos a escribir las condiciones de Karush-Kuhn-Tucker y obtenemos:

$$1) \nabla \Psi = \begin{bmatrix} 1 \\ 2y - 14/3 \end{bmatrix}; \nabla \Gamma_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}; \nabla \Gamma_2 = \begin{bmatrix} 1 \\ 1/3 \end{bmatrix}; \nabla \Gamma_3 = \begin{bmatrix} 1 \\ 2/9 \end{bmatrix}$$

$$1 + \rho_1 + \rho_2 + \rho_3 = 0$$

$$2y - 14/3 + \rho_1 + \rho_2/3 + 2\rho_3/9 = 0$$

$$2) \quad \begin{aligned} \rho_1 \cdot (r + y - 3) &= 0 \\ \rho_2 \cdot \left(r + \frac{y}{3} - \frac{7}{3} \right) &= 0 \\ \rho_3 \cdot \left(r + \frac{2y}{9} - \frac{113}{54} \right) &= 0 \end{aligned}$$

$$3) \quad \begin{aligned} \rho_1 &\leq 0 \\ \rho_2 &\leq 0 \\ \rho_3 &\leq 0 \end{aligned}$$

Para poder proceder a la resolución del sistema de ecuaciones consideramos la siguiente hipótesis: la nueva restricción es activa, es decir, las restricciones 1 y 2 son inactivas y la restricción 3 es activa. Bajo esta hipótesis:

$$\rho_1 = 0, \rho_2 = 0 \text{ restricciones inactivas} \rightarrow \rho_3 = -1$$

$$2y - 14/3 + \rho_1 + \rho_2/3 + 2\rho_3/9 = 0 \rightarrow 2y - 14/3 + 0 + 0/3 - 2/9 = 0 \rightarrow y = 66/27$$

$$\rho_3 \cdot \left(r + \frac{2y}{9} - \frac{113}{54} \right) = 0 \rightarrow r + \frac{2y}{9} - \frac{113}{54} = 0 \rightarrow r = 251/162$$

Verificamos si efectivamente las restricciones 1 y 2 son inactivas:

$$r + y - 3 = \frac{251}{162} + \frac{66}{27} - 3 = \frac{161}{162} \geq 0 \quad \text{CUMPLE}$$

$$r + \frac{y}{3} - \frac{7}{3} = \frac{251}{162} + \frac{66}{81} - \frac{7}{3} = \frac{5}{162} \geq 0 \quad \text{CUMPLE}$$

Por lo tanto hemos obtenido el resultado óptimo del subproblema *Proximal Bundle*, el cual indica que el subgradiente activo corresponde a la restricción 3: $\hat{s}^3 \equiv s_3$. Con la información obtenida hasta el momento podemos calcular el valor de la función poliédrica y el valor del error de linealización en el punto de análisis:

$$\varphi^3(y) := f(\bar{x}) + \hat{s}^3 \cdot (y - \bar{x})' = 85/54 + (-2/9) \cdot (66/27 - 7/3) = 251/162$$

$$\hat{e}^3 = f(\bar{x}) - \varphi^3(y) - t^3 \langle \hat{s}^3, \hat{s}^3 \rangle = 85/54 - 251/162 - 1/2 \cdot (-2/9)^2 = 0$$

El criterio de parada del algoritmo nos indica que debemos seguir iterando:

$$\delta^3 = f(\bar{x}) - \varphi^3(y) - t^3/2 \|\hat{s}^3\|^2 = 85/54 - 251/162 - 1/4 \cdot (-2/9)^2 = 6/486 > \underline{\delta}$$

Paso2

Calculamos el valor de la función objetivo analizada en el nuevo punto obtenido y realizamos el test de descenso:

$$f(y) = f(66/27) = 3 - 66/81 + \frac{(66/27)^2}{6} = 377/243$$

condición de descenso: $f(y) \leq f(\bar{x}) - m \cdot \delta^3$

$$377/243 \leq 85/54 - 1/2 \cdot 6/486 = 762/486 \quad \text{SE CUMPLE} \rightarrow \text{Paso de descenso}$$

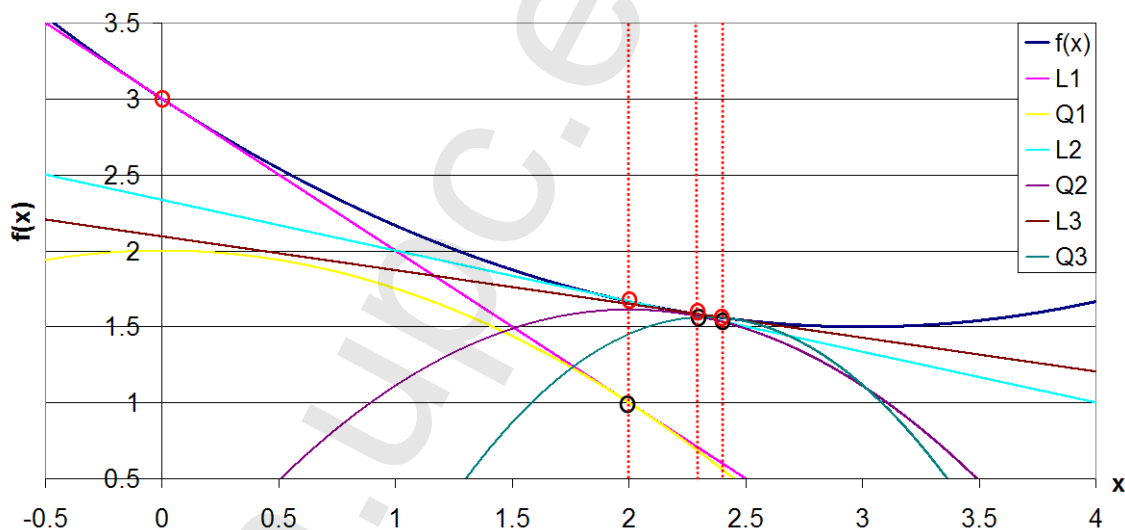


Figura 8 Algoritmo PB, iteración 3: función cuadrática asociada a y

Paso3

Actualizamos los errores de actualización trasladándolos al nuevo punto:

$$e_j \leftarrow e_j + f(y) - f(\bar{x}) - \langle s_j, y - \bar{x} \rangle \quad \forall j \in J$$

$$\hat{e}^3 \leftarrow \hat{e}^3 + f(y) - f(\bar{x}) - \langle \hat{s}^3, y - \bar{x} \rangle$$

$$\begin{aligned}
 e_1 &= 49/54 + 377/243 - 85/54 - (-1) \cdot (66/27 - 7/3) = 242/243 \\
 e_2 &= 1/54 + 377/243 - 85/54 - (-1/3) \cdot (66/27 - 7/3) = 8/243 \\
 e_3 &= 0 + 377/243 - 85/54 - (-2/9) \cdot (66/27 - 7/3) = 1/486 \\
 \hat{e}^3 &= 0 + 377/243 - 85/54 - (-2/9) \cdot (66/27 - 7/3) = 1/486
 \end{aligned}$$

El nuevo punto de análisis (mejor valor conocido) pasa a ser el valor encontrado en esta segunda iteración:

$$\bar{x} \leftarrow y$$

Paso4

$$l = 3 = \bar{l}$$

Vemos que el Bundle está lleno, por lo tanto tenemos que eliminar un elemento. El criterio seleccionado ha sido el de mayor error de linealización por tanto se procede a eliminar el elemento 1. Tras ello renombramos el Bundle:

$$B^3 = \begin{pmatrix} -1/3 & 8/243 \\ -2/9 & 1/486 \end{pmatrix}$$

Paso5

El paso realizado ha sido de descenso y por lo tanto el error de linealización asociado es nulo: $e_3 = 0$

Calculamos el subgradiente en el punto obtenido y añadimos la pareja (s_3, e_3) al *Bundle*:

$$s_3 = s(y) = s(66/27) = -1 + 66/81 = -5/27$$

La matriz *Bundle* contiene ahora la información almacenada en las tres últimas iteraciones:

$$B^3 = \begin{pmatrix} -1/3 & 8/243 \\ -2/9 & 1/486 \\ -5/27 & 0 \end{pmatrix}$$

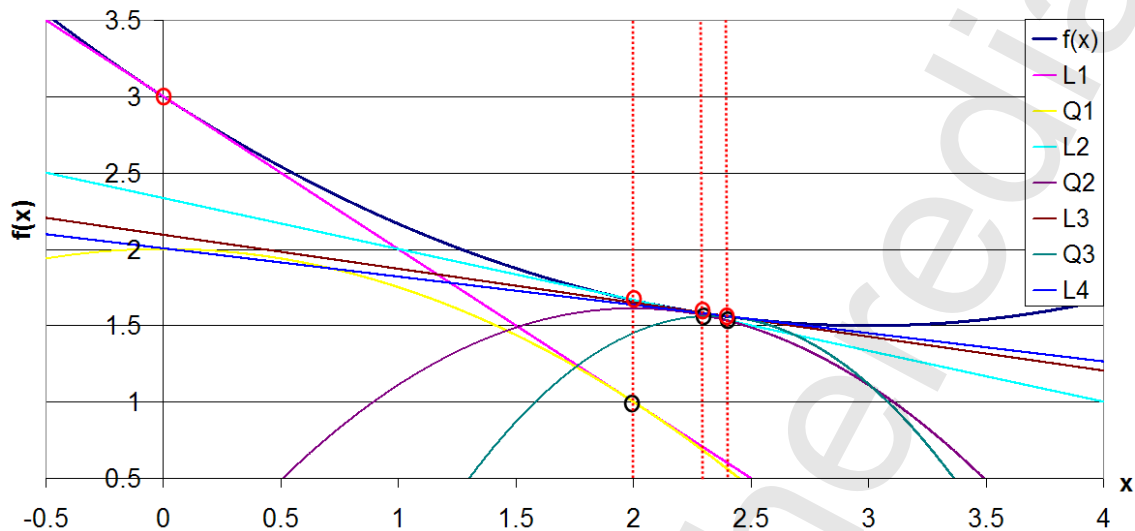


Figura 9 Algoritmo PB, iteración 3: subgradiente asociado a \bar{x}

Las tres iteraciones realizadas muestran la mecánica general del *Bundle Method aproximado*. Recopilamos los resultados obtenidos y recordamos cual es el resultado óptimo final:

$$x^* = 3 \quad f(x^*) = 3 - 3 + \frac{3^2}{6} = \frac{3}{2} = 1,5$$

k	t_k	y	r	$f(y)$	$s(y)$	\bar{x}	$f(\bar{x})$	$s(\bar{x})$	δ
0	-	-	-	-	-	0	3	-1	-
1	2	2	1	1,6	-0,3	2	1,6	-0,3	1
2	1	2,3	1,5	1,574	-0,2	2,3	1,574	-0,2	0,05
3	0,5	2,4	1,549	1,551	-0,185	2,4	1,551	-0,185	0,0123

Tabla 1. Ejemplo PB con $t^1 = 2$, resumen de resultados analíticos.

De la tabla superior se desprende que las tres iteraciones han supuesto pasos de descenso y por lo tanto en cada una de ellas se ha mejorado la mejor solución conocida hasta entonces. Sin embargo sigue sin cumplirse la condición de parada del algoritmo y por lo tanto es necesario seguir iterando. Para ello no servimos de ayuda computacional, tal y como se describe en el siguiente apartado.

IMPLEMENTACIÓN COMPUTACIONAL DEL EJEMPLO

La implementación computacional de este ejemplo persigue dos objetivos simultáneamente. Por una parte se pretende finalizar el proceso iterativo completo del Bundle Method aproximado para visualizar la evolución de los distintos valores así como su convergencia. Un segundo aspecto corresponde a poder implementar un código de resolución de problemas de una variable mediante el Bundle Method y testar su validez con los resultados obtenidos en las iteraciones manuales. Esto nos permitirá poder extender el código a problemas de mayor escala partiendo de una base sólida.

Dado que los aspectos propiamente computacionales de las implementaciones realizadas vienen descritos en un capítulo específico nos centraremos únicamente en los resultados obtenidos.

Se ha procedido a implementar el ejemplo presentado anteriormente utilizando los mismos valores en los parámetros seleccionables por el usuario. Para su resolución el solver *CPLEX* emplea 0.0625 segundos y presenta los siguientes resultados: (para más detalles sobre el equipo utilizado consultar el apartado 9. IMPLEMENTACIÓN)

k	t_k	y	r	$f(y)$	$s(y)$
0	-	-	-	-	-
1	2	2	1	1.6667	-0.3333
2	1	2.3333	1.5556	1.5741	-0.2222
3	0.5	2.4444	1.5494	1.5514	-0.1852
4	0.25	2.4907	1.5429	1.5432	-0.1698
5	0.125	2.5120	1.5396	1.5397	-0.1627
6	6.25E-02	2.5221	1.5380	1.5381	-0.1593
7	3.13E-02	2.5271	1.5373	1.5373	-0.1576
8	1.56E-02	2.5296	1.5369	1.5369	-0.1568
9	7.81E-03	2.5308	1.5367	1.5367	-0.1564

Tabla 2. Ejemplo *PB* con $t^1 = 2$, resumen de resultados computacionales. Parte 1.

k	<i>Paso</i>	\bar{x}	$f(\bar{x})$	$s(\bar{x})$	δ
0	-	0	3	-1	-
1	descenso	2	1.6667	-0.3333	1
2	descenso	2.3333	1.5741	-0.2222	5.56E-02
3	descenso	2.4444	1.5514	-0.1852	1.23E-02
4	descenso	2.4907	1.5432	-0.1698	4.29E-03
5	descenso	2.5120	1.5397	-0.1627	1.80E-03
6	descenso	2.5221	1.5381	-0.1593	8.27E-04
7	descenso	2.5271	1.5373	-0.1576	3.96E-04
8	descenso	2.5296	1.5369	-0.1568	1.94E-04
9	descenso	2.5308	1.5367	-0.1564	9.56E-05

Tabla 3. Ejemplo *PB* con $t^1 = 2$, resumen de resultados computacionales. Parte 2.

Vemos que efectivamente los resultados obtenidos en las tres primeras iteraciones coinciden con los correspondientes en las iteraciones manuales. El proceso iterativo se desarrolla sobre un total de 9 iteraciones hasta alcanzar el criterio de parada sin que el resultado obtenido sea óptimo.

El mejor resultado conocido (1,5367) se aleja en un 2,5% de la solución óptima (1,5), considerando que se trata de un ejemplo de pequeña escala podemos concluir que el criterio de parada no es lo suficientemente exigente. Esto nos lleva a formular la primera propuesta de modificación del algoritmo *PB*

en nuestra implementación. Considerando que el término δ es conceptualmente difícil de interpretar y de seleccionar de manera rigurosa para cada ejemplo, proponemos un criterio de parada basado en la distancia relativa entre la mejor solución conocida y una cota superior ajustada.

Otra cuestión que se aprecia en la tabla de resultados es que las 9 iteraciones son de mejora, o dicho de otro modo de “paso de descenso”. Esto en general no ocurrirá sistemáticamente y menos aun en problemas de gran escala. La tipología de los pasos realizados depende en gran medida de la longitud de paso seleccionada ya que una longitud de paso demasiado pequeña produce que los nuevos puntos encontrados sean muy próximos al punto de análisis, mientras que una longitud de paso excesivamente grande produce el efecto contrario.

Para ilustrar esta situación ejecutamos el código implementado con el único cambio de aumentar la longitud de paso inicial a $t^1 = 7$. CPLEX resuelve el problema en 0.109375 segundos y presenta los siguientes resultados:

k	t_k	y	r	$f(y)$	$s(y)$
0	-	-	-	-	-
1	7	7	-4	4.1667	1.3333
2	3.5	3.4998	-0.4998	1.5416	0.1666
3	1.75	3.2083	1.4931	1.5072	0.0694
4	0.875	3.1475	1.5030	1.5036	0.0492
5	0.4375	3.1260	1.5026	1.5027	0.0420
6	0.2188	3.1168	1.5023	1.5023	0.0389
7	0.1094	3.1126	1.5021	1.5021	0.0375

Tabla 4. Ejemplo *PB* con $t^1 = 7$, resumen de resultados computacionales. Parte 1.

k	<i>Paso</i>	\bar{x}	$f(\bar{x})$	$s(\bar{x})$	δ
0	-	0	3	-1	-
1	Nulo	0	3	-1	3,5
2	Descenso	3.4998	1.5416	0.1666	1.7498
3	Descenso	3.2083	1.5072	0.0694	0.0243
4	Descenso	3.1475	1.5036	0.0492	2.11E-03
5	Descenso	3.1260	1.5027	0.0420	5.29E-04
6	Descenso	3.1168	1.5023	0.0389	1.93E-04
7	Descenso	3.1126	1.5021	0.0375	8.29E-05

Tabla 5. Ejemplo *PB* con $t^1 = 7$, resumen de resultados computacionales. Parte 2.

Vemos que si bien la primera iteración presenta un paso nulo, el número total de iteraciones se reduce a 7. Además el resultado final es significativamente mejor, ya que nos encontramos únicamente a una distancia del 0,14% respecto al resultado óptimo. Por lo tanto vemos que si bien el criterio de parada δ es el mismo el resultado final varía de manera importante.

Este aspecto reafirma la idea expresada anteriormente de elegir otro criterio de parada que indique de manera más precisa la calidad del resultado obtenido.

Gráficamente podemos observar que en la primera iteración el resultado obtenido se aleja de manera significativa del valor óptimo de la variable. Sin embargo en la segunda iteración el algoritmo encuentra un punto muy próximo al óptimo.

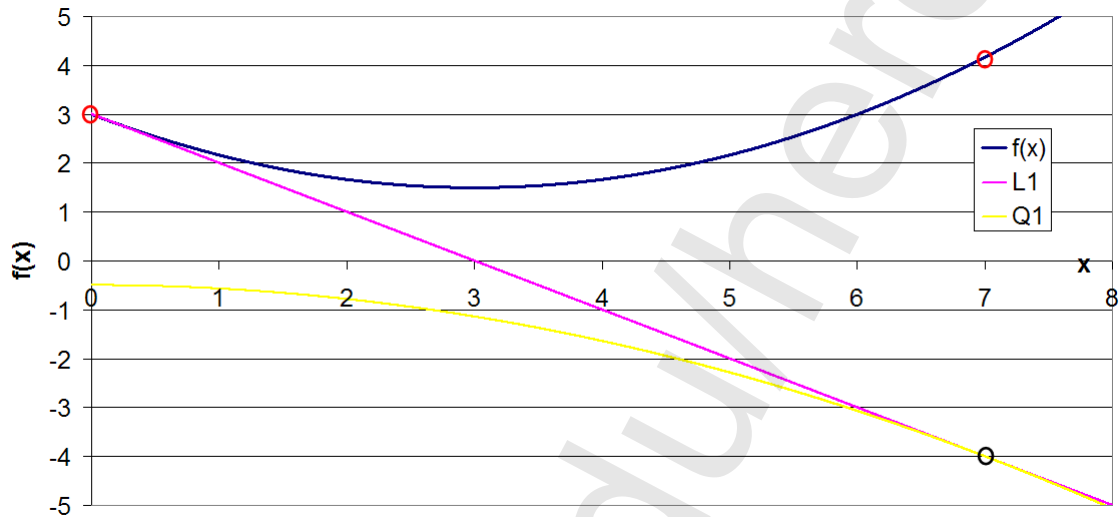


Figura 10 Algoritmo PB, ejemplo con $t^1 = 7$, iteración 1.

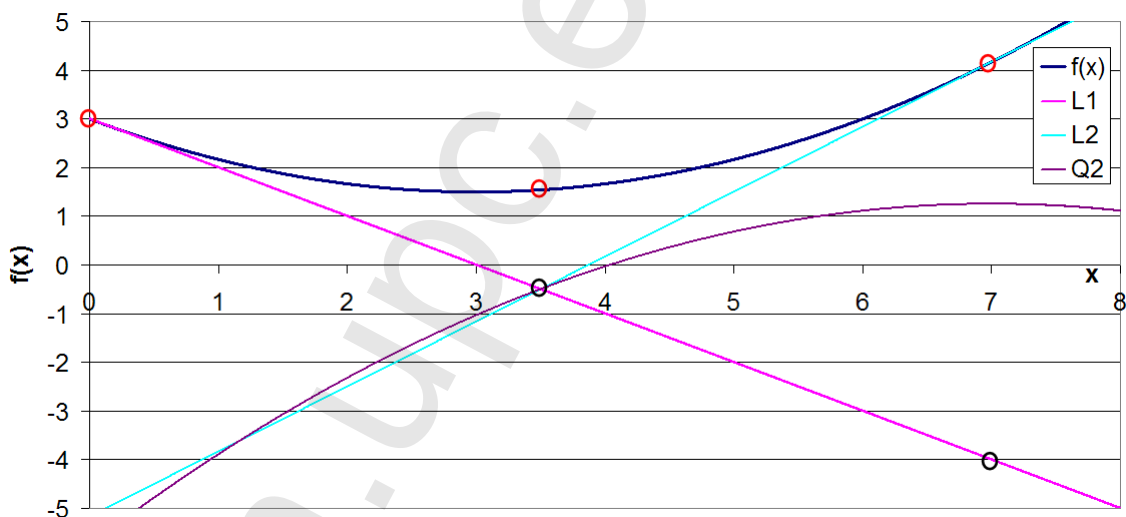


Figura 11 Algoritmo PB, ejemplo con $t^1 = 7$, iteración 2.

7. ADAPTACIÓN DEL PROBLEMA MULTIMERCADO DE OFERTA ÓPTIMA AL ALGORITMO PB

En los capítulos anteriores hemos procedido a analizar las características del algoritmo y hemos realizado una serie de iteraciones manuales en un ejemplo de pequeña escala. Tras ello se ha implementado el algoritmo PB obteniendo los resultados previstos en las iteraciones realizadas anteriormente.

En el presente capítulo se abordarán las transformaciones necesarias del problema multimercado de oferta óptima a fin de que enlace con el algoritmo PB de la forma algorítmicamente más conveniente. Es decir, se tratará de estructurar el problema de optimización original en otro que proporcione una cota inferior ajusta de dicho problema.

Tal y como veremos a continuación se planteará la relajación Lagrangiana correspondiente a la relajación de las restricciones (02) y (03). Estas restricciones ligam el comportamiento de las diferentes unidades entre sí, por tanto el problema resultante será divisible en subproblemas.

A continuación se detallarán dichos pasos de manera más exhaustiva:

7.1. Formulación de la función dual Lagrangiana

- 1) Formulamos el problema original de forma compacta. Constará de la función objetivo de maximización, las restricciones de enlace (02), (03) y del grupo de restricciones individuales de cada unidad térmica:

Problema original:

$$\begin{aligned} & \max_{p,q,f,b} \sum_{t \in T} \sum_{i \in I} \left\{ -c_{it}^u - c_{it}^d - c_i^b u_{it} + \right. \\ & \quad \left. + \sum_{s \in S} P^s \left[\lambda_i^{D,s} p_{it}^{M,s} + \lambda_i^{R,s} r_{it}^s g_i + \lambda_i^{I,s} w_{it}^s - \left(c_i^l p_{it}^s + c_i^q (p_{it}^s)^2 \right) \right] \right\} \\ & = \max_{p,q,f,b} C(p,q,f,b) \end{aligned} \tag{01}$$

sometido a: (02), (03), $\tau_i, i \in I$

- 2) Transformamos el problema original en un problema equivalente de minimización con el objetivo de encontrarnos con el caso estándar de la literatura de la optimización matemática.

Problema original transformado a problema de minimización:

$$\begin{aligned}
& \min_{p,q,f,b} \sum_{t \in T} \sum_{i \in I} \left\{ c_{it}^u + c_{it}^d + c_i^b u_{it} + \right. \\
& \quad \left. + \sum_{s \in S} P^s \left[-\lambda_t^{D,s} p_{it}^{M,s} - \lambda_t^{R,s} r_{it}^s g_i - \lambda_t^{I,s} w_{it}^s + \left(c_i^l p_{it}^s + c_i^q (p_{it}^s)^2 \right) \right] \right\} \\
& = \min_{p,q,f,b} -C(p,q,f,b) \tag{38}
\end{aligned}$$

sometido a: (02), (03), $\tau_i, i \in I$

- 3) Realizamos la relajación Lagrangiana de las restricciones (02) y (03). El objetivo es aprovechar la estructura del problema para poder separarla en subproblemas y poder tratarlas de manera independiente y potencialmente en paralelo, mejorando así el rendimiento computacional.

Se relajan las restricciones (02) y (03) siendo los respectivos multiplicadores $\mu_{t,j}^F$ y μ_t^B .

Relajación lagrangiana:

$$\begin{aligned}
\phi(\mu^F, \mu^B) = \min_{p,q,f,b} -C(p,q,f,b) + \sum_{t \in T} \sum_{j \in F} \mu_{t,j}^F \left(L_j^{FC} - \sum_{i \in U_j} f_{ij} \right) \\
+ \sum_{t \in T} \mu_t^B \left(\sum_{bc \in BC} L_{bc,t}^{BC} - \sum_{i \in I} b_{it} \right) \tag{39}
\end{aligned}$$

sometido a: $\tau_i, i \in I$

- 4) Para proceder a la división en subproblemas realizamos un cambio de notación, para ello definimos la matriz de elementos binarios J . En dicha matriz un elemento J_{ij} valdrá 1 si la unidad i puede generar energía para satisfacer la FC j . Por lo tanto la componente de relajación asociada a la restricción (02) vendrá definida como:

$$\sum_{t \in T} \sum_{j \in F} \mu_{t,j}^F \left(L_j^{FC} - \sum_{i \in I} f_{ij} \cdot J_{ij} \right) \tag{40}$$

Ahora podemos describir la expresión (39) como:

$$\begin{aligned}
\phi(\mu^F, \mu^B) = \min_{p,q,f,b} \sum_{t \in T} \sum_{i \in I} \left\{ c_{it}^u + c_{it}^d + c_i^b u_{it} - \sum_{j \in F} \mu_{t,j}^F f_{ij} J_{ij} - \mu_t^B b_{it} \right. \\
+ \sum_{s \in S} P^s \left[-\lambda_t^{D,s} p_{it}^{M,s} - \lambda_t^{R,s} r_{it}^s g_i - \lambda_t^{I,s} w_{it}^s + \left(c_i^l p_{it}^s + c_i^q (p_{it}^s)^2 \right) \right] \left. \right\} + \\
+ \sum_{t \in T} \sum_{j \in F} \mu_{t,j}^F L_j^{FC} + \sum_{t \in T} \sum_{bc \in BC} \mu_t^B L_{bc,t}^{BC} \tag{41}
\end{aligned}$$

- 5) Con la nueva notación la expresión (41) es una función separable en I subproblemas independientes:

División en subproblemas:

$$\phi(\mu^F, \mu^B) = \sum_{i \in I} \phi_i(\mu^F, \mu^B) + \sum_{t \in T} \left(\sum_{j \in F} \mu_{t,j}^F L_j^{FC} + \sum_{bc \in BC} \mu_t^B L_{bc,t}^{BC} \right) \quad (42)$$

Donde:

$$\begin{aligned} \phi_i(\mu^F, \mu^B) = \min_{p,q,f,b} \sum_{t \in T} \left\{ c_{it}^u + c_{it}^d + c_{it}^b u_{it} - \sum_{j \in F} \mu_{t,j}^F f_{ij} J_{ij} - \mu_t^B b_{it} \right. \\ \left. + \sum_{s \in S} P^S \left[-\lambda_t^{D,s} p_{it}^{M,s} - \lambda_t^{R,s} r_{it}^s g_i - \lambda_t^{L,s} w_{it}^s + \left(c_{it}^l p_{it}^s + c_{it}^q (p_{it}^s)^2 \right) \right] \right\} \quad (43) \end{aligned}$$

sometido a: τ_i

$$\sum_{t \in T} \left(\sum_{j \in F} \mu_{t,j}^F L_j^{FC} + \sum_{bc \in BC} \mu_t^B L_{bc,t}^{BC} \right) = cte \quad (44)$$

- 6) Los valores óptimos $\mu_{t,j}^{F*}$ y μ_t^{B*} de los multiplicadores de Lagrange se obtienen solucionando en dual de la función Lagrangiana:

$$L^* = \max \phi(\mu^F, \mu^B) \quad \text{sometido a: } \mu^F, \mu^B \text{ libres} \quad (45)$$

La solución del problema (45) es una cota inferior del problema original (01). Podemos obtener una oferta de solución del problema primal resolviendo (40) donde $\mu^F = \mu^{F*}$ y $\mu^B = \mu^{B*}$. En general dicha oferta de mercado no satisfará las restricciones dualizadas (02) y (03), para resolver esta situación se deben aplicar una serie de métodos heurísticos de recuperación de factibilidad. Este aspecto excede de los objetivos de este trabajo y por tanto será una de las cuestiones que se plantearán como líneas futuras de investigación.

7.2. Elementos del *Bundle Method*. Subproblema *Proximal Bundle*

Si tomamos el problema de optimización (45) como referencia nos faltaría definir la función subgradiente para poder aplicar el algoritmo PB como método de resolución.

La función $\phi(\mu^F, \mu^B)$ es no diferenciable. Sin embargo es convexa y por tanto podemos calcular sus subgradientes respecto a los multiplicadores de Lagrange:

$$z_{t,j}^F(\mu^F) = \partial\phi(\mu^F) = L_j^{FC} - \sum_{i \in U_j} f_{ij} \quad j \in F, t \in T \quad (46)$$

$$z_t^B(\mu^B) = \partial\phi(\mu^B) = \sum_{bc \in BC} L_{bc,t}^{BC} - \sum_{i \in I} b_{it} \quad t \in T \quad (47)$$

Conexión entre la función dual lagrangiana y el subproblema Proximal Bundle:

El problema de optimización (45) corresponderá a nuestro problema principal en el algoritmo de resolución. Sin embargo cabe destacar que el algoritmo PB prevé una función objetivo de minimización. Consecuentemente en la transferencia de información entre el problema principal y el subproblema deberemos tener en cuenta el cambio de signo en los subgradientes y en el valor de la función objetivo asociado a la solución analizada.

Es decir, si definimos el subproblema *Proximal Bundle* correspondiente al problema multimercado de oferta óptima de la siguiente manera:

(Para simplificar las expresiones de este apartado definimos la siguiente notación)

$$\mu = [\mu^F, \mu^B] \quad (48)$$

$$z(\mu) = [z(\mu^F), z(\mu^B)] \quad (49)$$

SUBPROBLEMA PROXIMAL BUNDLE:

$$\min_{\mu^k, \gamma^k} \left\{ \gamma^k + \frac{1}{2 \cdot \alpha^k} \|\mu_k - \bar{\mu}_k\|^2 \right\} \quad (50)$$

$$\text{sometido a: } \gamma^k \geq \tilde{\phi}(\bar{\mu}_k) - e^j + \tilde{z}(\mu^j) \cdot (\mu^k - \bar{\mu}_k)' \quad j \in \beta$$

La correspondencia entre ambos problemas de optimización será:

$$\phi(\mu) = -\tilde{\phi}(\mu) \quad (51)$$

$$z(\mu) = -\tilde{z}(\mu) \quad (52)$$

8. DATOS PARA LA RESOLUCIÓN

8.1. Origen de los datos

Es lógico pensar que para poder testar la implementación del algoritmo PB referido al problema multimercado de oferta óptima y sacar conclusiones, los datos de partida deben ser lo más cercanos posibles a los de una situación real.

En este contexto los datos de partida se han tomado del grupo *GNOM* basándose en los datos existentes en investigaciones en curso. Dentro de los diferentes grupos de datos disponibles se ha decidido seleccionar uno de tamaño medio, ya que el proceso de implementación requerirá la ejecución de diferentes pruebas por lo que se precisa un tamaño de problema relativamente manejable.

8.2. Dimensiones del problema

Los datos de partida seleccionados definen un problema de optimización que consta de:

Número de unidades térmicas: $|I| = 10$

Número de intervalos temporales: $|T| = 24$

Número de escenarios: $|S| = 25$

Número de contratos de tipo FC: $|F| = 3$

Número de contratos de tipo BC: $|B| = 3$

El vector de número de periodos iniciales durante los cuales las unidades deben permanecer encendidas es nulo $G = 0$

El vector de número de periodos iniciales durante los cuales las unidades deben permanecer apagadas es nulo $H = 0$

Podemos por lo tanto determinar el número total de restricciones del problema original sumando el número de restricciones de cada grupo:

(02), (04) $\rightarrow |F||T||I|$

(03), (05), (08), (09), (10), (18) $\rightarrow |T||I|$

(06), (07), (11), (12), (13), (14), (15) $\rightarrow |T||I||S|$

(16), (17) $\rightarrow |T - 1||I|$

$$(19), (20) \rightarrow |I|$$

$$(25), (26) \rightarrow |T||S|$$

$$(21)+(23) \rightarrow \left| T - \sum_{i \in I} G_i \right| |I|$$

$$(22)+(24) \rightarrow \left| T - \sum_{i \in I} H_i \right| |I|$$

$$N_{restricciones} = 2 \cdot |F||T||I| + 6 \cdot |T||I| + 7 \cdot |T||I||S| + 2 \cdot |T-1||I| + 2 \cdot |I| + \\ + 2 \cdot |T||S| + \left| T - \sum_{i \in I} G_i \right| |I| + \left| T - \sum_{i \in I} H_i \right| |I| = 47040$$

Así mismo podemos determinar el número de variables lineales y binarias del problema abortado:

$$C_u, C_d, b, q \rightarrow |T||I|$$

$$pM, p, w \rightarrow |T||I||S|$$

$$f \rightarrow |T||I||F|$$

$$N_{lineales} = 4 \cdot |T||I| + 3 \cdot |T||I||S| + |T||I||F| = 19680$$

$$u \rightarrow |T||I|$$

$$r \rightarrow |T||I||S|$$

$$N_{binarias} = |T||I| + |T||I||S| = 6240$$

Resumiendo, el problema a resolver será un problema de mediana escala ya que consta de 47040 restricciones y 25920 variables. El 24% de dichas variables son binarias por lo tanto podemos esperar que el árbol de exploración *Branch&Bound* sea relativamente extenso.

9. IMPLEMENTACIÓN

9.1. Lenguaje de programación seleccionado

A la hora de abordar la implementación, la primera decisión corresponde al lenguaje de programación a utilizar. Esta decisión viene acotada por el *solver* seleccionado para resolver los problemas lineales, ya que el lenguaje tendrá que ser interpretable por él. En este caso aprovechando las licencias de uso del grupo *GNOM* se utilizará el *solver* comercial *CPLEX*, de robustez y precisión altamente contrastadas.

CPLEX puede interpretar modelos matemáticos creados mediante lenguajes de programación generales como *C++*, *Java* o *.NET* así como mediante lenguajes propios del entorno de la optimización matemática como *AMPL*. Por tanto se tendrán que valorar las ventajas e inconvenientes que ofrece cada plataforma de cara a poder seleccionar la más adecuada.

El primer candidato analizado corresponde a *AMPL* por varias razones: La primera es que durante el periodo docente del master ha sido el lenguaje propuesto mayoritariamente por el profesorado, por tanto la parte de aprendizaje básico del lenguaje está realizado. Además, una gran parte de los modelos realizado por el *grupo GNOM* se ha construido mediante este código y por tanto su validez y adaptación a problemas estocásticos de mercado eléctrico está más que contrastada. Finalmente habría que añadir los código escritos mediante *AMPL* son generalmente compactos y fácilmente interpretables por terceras personas.

A pesar de estas ventajas evidentes se ha preferido utilizar un lenguaje de programación general, en concreto *C++*. Las razones prácticas son las siguientes:

- La versión gratuita de *AMPL* sólo permite ejecutar problemas de pequeña escala y por lo tanto el problema abordado no es ejecutable. Esta situación exigiría disponer de una licencia de *AMPL*. Si bien esta situación no es problemática en el entorno del grupo *GNOM* dificulta su ejecución en otros equipos informáticos sin licencia.
- Los errores de programación o de compilado son significativamente más fáciles de localizar y subsanar en el lenguaje *C++* gracias a las advertencias mostradas.
- Permite importar y exportar datos o soluciones de manera relativamente sencilla a otros lenguajes y bases de datos. La universalidad del lenguaje permite la conexión con numerosos programas informático.s

Además, habría que añadir dos aspectos de nivel más personal que han inclinado la balanza hacia el uso de *C++*.

- Gran parte de los proyectos posteriores al presente Trabajo de Fin de Master los realizaré junto con investigadores del departamento de *Matemática Aplicada, Estadística e Investigación Operativa* de la *Euskal*

Herriko Unibertsitatea / Universidad del País Vasco (EHU-UPV). En dicho grupo de trabajo el lenguaje estándar de programación es C++.

- Personalmente considero que AMPL es un lenguaje muy útil para abordar problemas de optimización matemática pero su integración como módulo de optimización dentro de otras aplicaciones puede ser complicado, al contrario de lo que ocurre con C++.

En resumen, considero que el lenguaje de programación C++ es no solo un lenguaje versátil que se adapta al problema abordado sino que a nivel personal me supondrá una formación muy enriquecedora.

9.2. Herramientas C++

Una vez seleccionado C++ como lenguaje de programación pasamos a decidir la tipología de la conexión entre C++ y CPLEX. En este contexto se nos presenta dos metodologías de trabajo:

- **Concert Technology:** Se trata de metalenguaje creado por IBM para la interconexión de elementos optimización programados en C++ y el solver CPLEX. Si trabajamos mediante esta opción los elementos programados en C++ serán divididos en dos objetos: IloModel (que abarca todas las especificaciones del modelo matemático) y IloCplex (que contiene toda la información referida a la configuración de CPLEX). De esta manera el usuario irá definiendo objetos en C++ que a continuación insertando en uno u otro bloque. La interconexión entre los dos objetos principales será automática una vez lanzada la resolución. El usuario puede además demandar información sobre el estado actual de cada uno de estos objetos. El siguiente esquema resume los aspectos descritos:

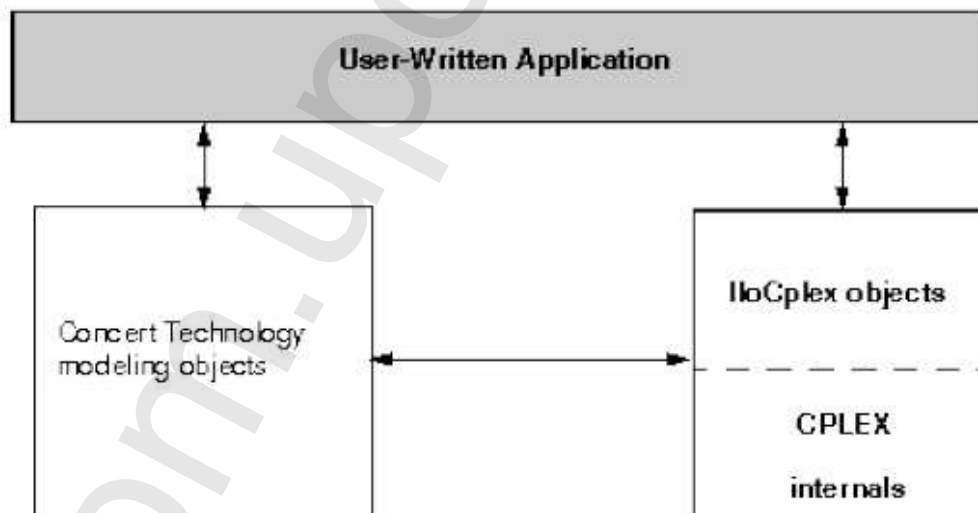


Figura 12 Esquema estructural de *Concert Technology*.

- **Callable library:** Esta mecánica de trabajo se basa en desarrollar un código en C++ donde se crean todos los objetos necesarios mediante el lenguaje habitual de C++. Tras ello podemos llamar a las librerías de la

base de datos de *CPLEX* que asocian los objetos creados con el núcleo de *CPLEX*. De manera esquemática el diagrama de trabajo será el siguiente:

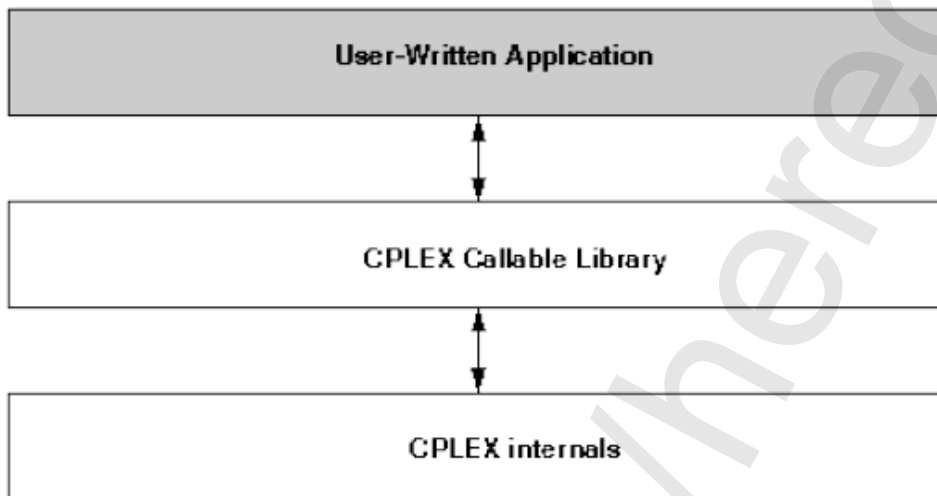


Figura 13 Esquema estructural de *Callabe Library*.

Con el objetivo de valorar el enfoque que mejor se adapta a nuestro problema se procede a analizar los ejemplos incluidos en el manual de *CPLEX*. Estos pequeños ejemplos están implementados en ambos lenguajes y permite observar visualmente las diferencias entre los códigos generados con cada mecánica de trabajo.

El análisis de dichos ejemplos arroja las siguientes conclusiones: A un usuario avanzado de C++ la opción **Callable Library** le permite trabajar en un entorno clásico donde las novedades residen en asociar los objetos creados con la librería correspondiente, sin necesidad de familiarizarse con el metalenguaje creado por *IBM*. Sin embargo para un nuevo usuario, como en nuestro caso, el **Concert Technology** tiene la ventaja de presentar una estructura más ordenada y códigos más compactos por lo que su manejo resultará más sencillo. Además la nomenclatura utilizada en las funciones de **Concert Technology** ayuda a identificar la naturaleza de los objetos.

Teniendo en cuenta el peso específico que la implementación tendrá en el presente Trabajo de Fin de Master se ha preferido utilizar el esquema de Concert Technology de cara a poder emprender la fase computacional en un periodo aceptable de tiempo. Todo ello sin perjuicio de que en futuras mejoras del código se pueda adaptar lo implementado hasta el momento a una estructura Callable Library.

9.3. Problemas de optimización implementados

Durante el desarrollo del presente Trabajo de Fin de Master se han implementado los siguientes problemas de optimización:

- Problema original resuelto mediante CPLEX.
- Relajación lineal del problema original.
- Resolución del dual lagrangiano mediante el método del subgradiente.
- Resolución del dual lagrangiano dividido en subproblemas mediante el método del subgradiente.
- Resolución del dual lagrangiano mediante el *Bundle Method* aproximado propuesto por Lemaréchal.

RESOLUCIÓN DEL PROBLEMA ORIGINAL

Se procede a implementar el problema multimercado de oferta óptima siguiendo la descripción original que aglutina las expresiones (01)~(26). Esta tarea servirá para la familiarización con el metalenguaje *Concert Technology* de IBM y para poder identificar posibles dificultades que podrán generarse al implementar códigos más complejos.

Constatamos que la puesta en marcha del equipo informático se torna más compleja de lo previsto, ya que surgen una serie de incompatibilidades entre algunas versiones de compiladores de C++ y el solver CPLEX. Finalmente se opta por utilizar *Microsoft Visual Studio 2005* como compilador, que si bien no es la versión más reciente de dicho producto permite el enlace con CPLEX de manera robusta. Estas dificultades suponen un retraso importante respecto al plan de trabajo inicial.

No obstante, una vez el equipo informático está configurado correctamente, se procede al aprendizaje de los principales comandos de *Concert Technology* y su aplicación a nuestro problema principal. Siendo esta una implementación que cuenta únicamente con elementos clásicos de optimización matemática (función objetivo y restricciones), los ejemplos existentes en la literatura relacionada con *Concert Technology* resultan de gran ayuda.

Una vez finalizado el proceso se comparan los resultados obtenidos con los correspondientes a una implementación en AMPL previamente creada por miembros del grupo GNOM. Este análisis permite detectar errores de implementación en el código C++ que una vez subsanados presentan los resultados esperados.

Contamos pues con un código básico que nos permite solucionar el problema multimercado de oferta óptima mediante CPLEX y *Concert Technology*. La solución óptima servirá como referencia a la hora de valorar la calidad de los resultados obtenidos mediante el resto de implementaciones consideradas.

RELAJACIÓN LINEAL DEL PROBLEMA ORIGINAL

Si bien en esta ocasión conocemos el resultado óptimo del problema, en general esto no será así ya que nuestro objetivo último es precisamente obtener dichos valores. Por ello tendremos que plantear una serie de resoluciones que conduzcan al cálculo de la solución óptima del problema.

En el primero de ellos nos planteamos resolver la relajación lineal del problema original. Es decir, redefinimos las variables binarias del problema original como variables lineales con valores comprendidos entre 0 y 1:

(01)~(26)

donde

$$0 \leq r_{it}^s \leq 1 \quad t \in T, s \in S, i \in I \quad (53)$$

$$0 \leq v_{it}^s \leq 1 \quad t \in T, s \in S, i \in I \quad (54)$$

La resolución de este problema de optimización persigue dos objetivos: El primero corresponde a obtener una cota superior relativamente ajustada del problema original, ya que el resultado obtenido mediante esta resolución será mejor o igual al resultado óptimo final. El segundo objetivo será almacenar el valor de las variables duales correspondientes a las restricciones (02) y (03) y poder utilizarlas como valores iniciales a la hora de resolver el problema dual lagrangiano.

La implementación de este problema de optimización corresponde a la extensión natural del código implementado para la resolución mediante *CPLEX*. Sin embargo cabe destacar que el proceso de encontrar las funciones que hacen posible el almacenamiento del valor de las variables duales ha sido costoso. Hay que tener en cuenta que este tipo de aspectos exceden de los ejemplos existentes las guías de usuario de *Concert Technology*. No obstante tras una serie de pruebas se comprobó que la información quedaba almacenada correctamente.

DUAL LAGRANGIANO RESUELTO MEDIANTE EL MÉTODO DEL SUBGRADIENTE

Nos centramos ahora en resolver el problema dual lagrangiano (45). Para ello en un primer momento tomamos como método de resolución el método del subgradiente. La razón reside en que antes de abordar dicho problema mediante el algoritmo PB se quiere tener una idea de la complejidad de su resolución y de la calidad de los valores duales iniciales obtenidos mediante la relajación lineal del problema original.

En este contexto el algoritmo del subgradiente presenta la ventaja de ser significativamente más sencillo de implementar que el algoritmo PB. Si bien su convergencia es, en general, lenta se trata de un algoritmo robusto y ampliamente documentado en la literatura de la optimización matemática. Por tanto la selección de los parámetros principales se realizará siguiendo las recomendaciones recibidas al respecto durante la docencia del Master.

Con tal de simplificar el criterio de parada partiremos de la premisa de que conocemos de antemano el valor óptimo del problema principal (38). Así el algoritmo detendrá cuando el valor de la función dual lagrangiana supere un

cierto porcentaje n del valor óptimo de (38). El siguiente esquema general del algoritmo del subgradiente permite interpretar más fácilmente las cuestiones planteadas en las líneas precedentes, siendo:

Algoritmo del subgradiente:

Inicio

Determinar un valor inicial de μ y calcular el valor de la función dual asociada $\phi(\mu)$.

Mientras $\phi(\mu) < n \cdot (P01)^*$ & $k < N_{iter}$ hacer

Calcular el valor de la función dual $\phi(\mu)$ resolviendo el dual lagrangiano asociado a μ

Calcular el valor del subgradiente $z(\mu)$

Calcular la longitud de paso ω

Actualizar las variables duales $\mu \leftarrow \mu + \omega \cdot z(\mu)$

Fin mientras

Fin

Donde la longitud de paso viene definida como:

$$\omega = \delta \cdot \frac{\bar{\phi} - \phi(\mu)}{\|z(\mu)\|} \quad \text{siendo} \quad 0 \leq \delta \leq 2$$

Y el valor del parámetro δ en la iteración $k+1$ se define como:

$$\delta^{k+1} := \begin{cases} \alpha \delta^k & \text{si } \underline{\phi}^k = \underline{\phi}^l, l = l, l-1, \dots, l-l_{\max} \\ \delta^k & \text{en otro caso} \end{cases}$$

Siendo:

$\alpha \in]0,1[$: Factor de reducción del escalar δ

l_{\max} : Número máximo de iteraciones sin mejorar la cota inferior $\underline{\phi}$

Por lo tanto los parámetros $N_{iter}, n, \delta^1, \alpha, l_{\max}$ son definidos por el usuario al inicio de la ejecución del algoritmo, y de su correcta selección depende la convergencia del algoritmo del subgradiente. Si las sucesivas longitudes de paso son demasiado grandes el algoritmo tendrá un comportamiento oscilante mientras que unas longitudes de paso que decrecen rápidamente tendrán como efecto que las sucesivas iteraciones serán cada vez más cercanas entre sí hasta acabar convergiendo a un punto no óptimo.

En cuanto a la implementación propiamente dicha, este método supone una complejidad significativamente mayor respecto a las realizadas hasta el

momento. No sólo por el aumento de variables y elementos en el código, sino por su naturaleza. El hecho de integrar la resolución en un proceso iterativo hace que el esquema general de *Concert Technology* de crear dos bloques principales tenga que ser gestionado con prudencia. Los elementos que son compartidos por todas las iteraciones realizadas, como por ejemplo las restricciones, pueden mantenerse en el bloque sin necesidad de actualizaciones. Sin embargo los valores de las variables duales cambian iterativamente y por tanto la forma de la función objetivo a minimizar $\phi(\mu)$ cambia en cada iteración. Por lo tanto el elemento de la función objetivo deberá ser actualizado en su bloque correspondiente.

DUAL LAGRANGIANO DIVIDIDO EN SUBPROBLEMAS RESUELTO MEDIANTE EL MÉTODO DEL SUBGRADIENTE

Esta implementación es una extensión del método del subgradiente explicado en el punto anterior, y está dirigida hacia el aprovechamiento computacional de la estructura del problema.

La diferencia respecto a la implementación anterior es que la evaluación de la función dual $\phi(\mu)$ resolviendo el dual lagrangiano asociado a μ no se realiza tomando como referencia modelo (39) sino que nos valemos del modelo (42). Para ello será necesario definir cada uno de los subproblemas.

En el contexto de la implementación este cambio supone una reestructuración del código. En un primer momento se trato de poder definir al inicio de cada iteración el conjunto de subproblemas y después ir ejecutándolos secuencialmente o en forma paralela. No obstante no se ha encontrado ninguna evidencia práctica ni literaria de que esto sea posible mediante *Concert Technology*.

La estructura de dos bloques por problema que crea el *Concert Technology* obliga o bien a copiar el bloque de restricciones tantas veces como subproblemas (de manera que cada función objetivo tenga asociado su propio bloque de restricciones) o bien eliminar la función objetivo de un subproblema que acaba de ser resuelto y sustituirlo por el inmediatamente siguiente. Esta segunda opción nos permite disminuir significativamente el tamaño interno del código creado pero nos obliga a trabajar secuencialmente, ya que un problema no es definido hasta haber resuelto en precedente.

Teniendo en cuenta que el ordenador utilizado para realizar las pruebas computacionales del presente trabajo dispone de un único núcleo se ha optado por implementar la segunda opción. Queda claro no obstante que en un contexto en que se pueda trabajar en paralelo la primera opción descrita anteriormente es la más adecuada.

DUAL LAGRANGIANO RESUELTO MEDIANTE *PROXIMAL BUNDLE METHOD*

El esquema de *Proximal Bundle Method* implementado coincide estructuralmente con el descrito en el capítulo “6. MÉTODO DE RESOLUCIÓN: *PROXIMAL BUNDLE METHOD*” no obstante se ha decidido realizar algunos cambios y adaptaciones en el algoritmo. Estos se podrían resumir en:

- Se prescinde del valor $\underline{\delta}$ como indicador de parada del algoritmo. Tal y como se ha podido ver en el ejemplo mostrado anteriormente, este valor es poco intuitivo y difícil de cuantificar. En este contexto sustituimos la condición de parada por una doble condición:

La primera es no rebasar una cierta cantidad límite de iteraciones seleccionada por el usuario. Se decide incluir este aspecto para que si el algoritmo converge a un punto no óptimo el proceso iterativo no continúe eternamente.

La segunda condición, y más importante, es valorar la distancia entre la mejor solución encontrada mediante el *Proximal Bundle Method* y una cota superior ajustada del problema (01)~(26). Será el usuario quien decidirá la tolerancia exigida.

Consecuentemente se podrá prescindir del cálculo del valor δ^k en cada iteración k .

- El segundo cambio corresponde a la gestión del elemento (\hat{s}^k, \hat{e}^k) en el paso **4.01** del algoritmo *Proximal Bundle Method*. Dicho elemento contiene el subgradiente activo de la última iteración realizada, así como su error de linealización respecto al punto de análisis. El objetivo de añadir dicho elemento en cada una de las iteraciones responde a evitar que durante el proceso de eliminación de cortes del *Bundle* se suprima la información referida al último corte activo. Ésta técnica tiene el inconveniente de que si el último corte activo no hubiese sido eliminado pasaría a estar duplicado al final del paso **4.01**.

Para evitar dicha situación de duplicidad lo que se propone es identificar al finalizar cada una de las resoluciones del subproblema *Proximal Bundle* el corte activo. Dicho corte será señalado específicamente para evitar que sea candidato durante el proceso de eliminación de elementos. Su identificación se realizará mediante el análisis de las holguras de las restricciones o por medio del análisis del valor de las variables duales asociadas a dichas restricciones.

Esta estrategia permitirá prescindir del cálculo de los valores \hat{s}^k y \hat{e}^k en cada iteración k .

- Sin embargo todavía queda por definir el criterio de eliminación de elementos cuando el *Bundle* esté lleno. A este respecto se ha decidido eliminar el elemento del *Bundle* que presente mayor error de linealización, siempre que este no coincida con el elemento activo en la última iteración (tal y como se ha explicado en el anterior punto). El objetivo es tener caracterizada la función objetivo de la manera más detallada posible en las áreas próximas al punto de análisis. Además, se ha decidido que el número de elementos eliminados sea unitario.
- Por último señalar que se ha definido el siguiente criterio de actualización de t^k basado parcialmente en el correspondiente al método del subgradiente:

$$t^{k+1} := \begin{cases} \hat{\alpha} \cdot t^k & \text{si } \overline{\mu^{k+1}} = \overline{\mu^k} \\ t^k & \text{en otro caso} \end{cases}$$

Siendo

$\hat{\alpha} \in]0,1[$: Factor de reducción del escalar t

La implementación del *Proximal Bundle Method* ha supuesto un importante salto cuantitativo en cuanto a longitud y complejidad del código. El hecho de trabajar iterativamente resolviendo un problema principal y un subproblema ha hecho que la gestión de los bloques de *Concert Technology* sea sumamente delicada.

Hay que tener en cuenta que los diferentes bloques se actualizan al final de cada iteración y este paso se tiene que realizar sin perturbar el bloque equivocado. Es decir, al final de cada resolución del subproblema se tiene que enviar al problema principal la actualización de los valores de las variables duales sin perturbar el resto de elementos. Tras ello, al resolver el problema principal el valor del subgradiente asociado a las variables duales analizadas debe ser actualizado en el subproblema sin modificar los valores ya existentes en el *Bundle*.

Este proceso a simple vista parece sencillo pero en la práctica provoca un proceso de actualizaciones circular. Es decir, si no se corta el flujo de información y se realizan las actualizaciones paso a paso puede ocurrir que cada cambio en un elemento se propague al resto términos por las relaciones definidas entre sí, perdiendo así el control del usuario de las actualizaciones realizadas.

Otro punto crítico reside en la correcta actualización y almacenado de los errores de linealización asociados a los subgradiantes. Queda claro que si este proceso no se realiza correctamente los diferentes cortes dejarán de caracterizar la función objetivo de manera detallada. Este aspecto es

especialmente crítico a la hora de eliminar e introducir nuevos elementos en el *Bundle*.

Por otra parte, el hecho de no aprovechar la opción de paralelización que presenta la estructura del problema produce proceso iterativo computacionalmente lento en el ordenador utilizado. Consecuentemente la fase de realizar pruebas computacionales para la detección de errores se ha alargado más de lo previsto.

Como consideración final general habría que añadir que la mayor parte del tiempo de realización de este Trabajo de Fin de Master ha sido dedicado a la implementación de estos modelos. Si bien es difícil cuantificar el trabajo realizado sin analizar el código creado cabe subrayar el grueso del trabajo reside en esta parte.

10. RESULTADOS OBTENIDOS

El presente capítulo tiene como objetivo presentar los resultados obtenidos mediante las diferentes implementaciones realizadas. Cabe subrayar una vez más que debido al equipo informático utilizado las comparaciones respecto al tiempo de ejecución no pueden tomarse en su literalidad, sino que servirán como indicadores de las resoluciones realizadas.

En este contexto, con el objetivo de agilizar la obtención de resultados y poder analizar el comportamiento de las implementaciones durante el desarrollo del trabajo, se ha decidido anular la componente cuadrática de la función objetivo (01). Es decir se ha fijado:

$$c_i^q = 0 \quad \forall i \in I \quad (55)$$

A la hora de comparar los resultados obtenidos mediante el solver *CPLEX* y mediante la resolución del dual lagrangiano utilizando en *Proximal Bundle Method*, habrá que tener en cuenta que se espera que la ganancia computacional del *PBM* será significativamente mayor cuando la función objetivo sea cuadrática.

Por las razones expuestas en los párrafos precedentes, se mostrará el tiempo de ejecución pero no será un indicador del que se puedan extraer conclusiones.

10.1. Resolución del problema original mediante el solver *CPLEX*

Se procede a resolver la implementación realizada y verificar que los resultados obtenidos coinciden con los correspondientes a la resolución mediante *AMPL* y *CPLEX*.

En un primer momento los resultados obtenidos mediante la implementación en C++ se presentan ligeramente inferiores a los esperados. Tras comprobar concienzudamente que no existían errores en la implementación se descubrió que la razón residía en la configuración de *CPLEX*. Hasta ese momento se estaba trabajando sin cambiar la configuración presentada por defecto por *CPLEX*, no obstante las resoluciones realizadas mostraron que la precisión obtenida era insuficiente. La configuración finalmente adoptada ha sido utilizada en el resto de implementaciones y contempla los siguientes cambios:

Tolerancia de integralidad: 1E-10	<i>setParam(EpInt, 1E-10)</i>
GAP relativo en el óptimo: 1E-10	<i>setParam(EpGap, 1E-10)</i>
GAP absoluto en el óptimo: 0	<i>setParam(EpAGap, 0)</i>

Los resultados finales coinciden con los obtenidos mediante *AMPL* y *CPLEX* y se resumen en:

Valor de la función objetivo: -6,29312E7 unidades monetarias
 Tiempo de ejecución: 113 segundos

Sorprende que en el resultado óptimo del problema multimercado de oferta óptima el beneficio obtenido corresponda a un valor negativo, es decir, a una situación de pérdidas. Este extremo se explica por el hecho de que no se ha tenido en cuenta los términos relativos a los ingresos derivados de los contratos bilaterales y de futuros, cuya expresión es:

$$\sum_{t \in T} \left(\sum_{bc \in BC} \lambda_{bc}^{BC} L_{bc,t}^{BC} + \sum_{j \in F} (\lambda_j^{FC} - \bar{\lambda}_t^{D,s}) L_j^{FC} \right)$$

Este resultado se tomará como referencia en las siguientes implementaciones consideradas.

10.2. Resolución de la relajación lineal del problema original

Para proceder a la resolución lineal de problema original el único cambio en la implementación es el de la definición de las variables binarias, tal y como se había mencionado anteriormente.

Tras realizar dichos cambios se procede a ejecutar el código resultante y se obtiene un valor óptimo de la función objetivo de -6,24825E7 unidades monetarias. No obstante al analizar los valores de las variables linealizadas se observa un comportamiento anómalo, ya que su valor es entero y conducen a un valor óptimo de la función objetivo menor que el encontrado en el apartado anterior. Queda claro que existe algún aspecto que no se ha tenido en cuenta en el momento de la implementación.

Se verifican tanto la implementación como la configuración de *CPLEX* y finalmente se concluye que la fuente de dicha anomalía reside en las restricciones (11) y (12). Si las recordamos:

$$p_{it}^s - p_{i,(t-1)}^s \leq (1 - r_{it}^s) R_i \quad t \in T, i \in I, s \in S \quad (11)$$

$$p_{it}^s - p_{i,(t-1)}^s \geq (1 - r_{it}^s) (-R_i) \quad t \in T, i \in I, s \in S \quad (12)$$

En concreto la parte problemática corresponde a tomar valores excesivamente grandes del escalar R_i de las diferentes unidades i . Para ilustrar su efecto proponemos el siguiente ejemplo:

Supongamos que a unos índices t, i, s concretos les corresponden los siguientes valores: $p_{it}^s = 175$, $p_{i,(t-1)}^s = 160$. Supongamos además que el usuario ha definido los siguientes valores: $R_i = 1E15 \quad \forall i \in I$.

En este contexto la restricción (11) implica que $r_{it}^s \neq 1$, mientras que la restricción (12) no añade ninguna condición adicional sobre r_{it}^s . Teniendo en

cuenta el signo de r_{ii}^s en la función objetivo, CPLEX tratará de asignar a r_{ii}^s el mayor valor factible posible. Resolviendo este pequeño problema mediante *Microsoft Excel* se obtiene el siguiente resultado óptimo: $r_{ii}^s = 0.9999999999999943$. Este valor se muestra en la solución de CPLEX como $r_{ii}^s = 1$. Pero hay que tener en cuenta que si bien $r_{ii}^s = 1$ es una solución visiblemente entera no es factible en el problema original, ya que el valor asociado a los datos del ejemplo debería ser $r_{ii}^s = 0$. Si bien la solución obtenida es factible en la relajación lineal del problema, el gran valor del escalar R hace que la solución prácticamente obvie la restricción (11).

Teniendo en cuenta dicha problemática se toma el siguiente criterio para el conjunto de las implementaciones:

$$R_i = \max(\bar{P}_i) \quad \forall i \in I \quad (56)$$

Una vez añadido dicho criterio a la implementación de la relajación lineal del problema se obtienen los siguientes resultados:

Valor de la función objetivo:	-6,2576E7 unidades monetarias
Tiempo de ejecución:	4.3125 segundos

Vemos que el valor óptimo encontrado es un 0,57% mayor respecto al valor óptimo del problema original, por lo tanto podemos concluir que la relajación lineal del problema original proporciona una cota superior ajustada al problema original.

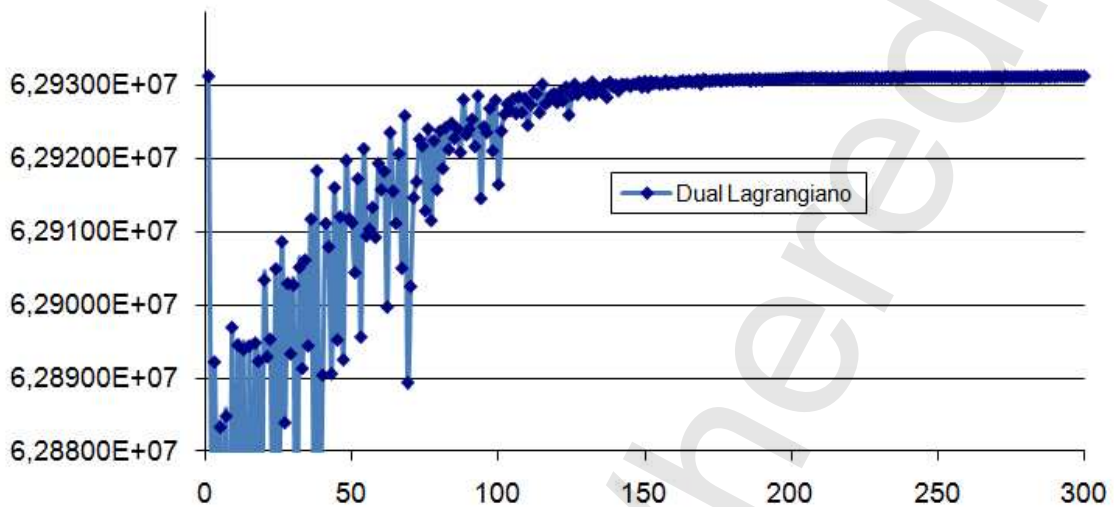
Si bien no nos podemos guiar por los valores del tiempo de ejecución, lo que sí queda claro es que el orden de magnitud del tiempo empleado se reduce significativamente respecto a la resolución del problema original.

10.3. Resolución del dual lagrangiano mediante el método del subgradiente

Tras resolver problemas de optimización basados en el modelo original del problema, pasamos a implementar el método del subgradiente con el objetivo de encontrar la solución óptima del problema dual lagrangiano (45). Hay que recordar que para su formulación se ha transformado el problema original de maximización en uno de minimización, consecuentemente a la hora de comparar los resultados hay que tener en cuenta el cambio de signo correspondiente.

Para ello tomamos como valores iniciales de las variables duales los asociados a las restricciones correspondientes en la solución óptima de la relajación lineal. Ejecutamos este entorno tomando como cota ajustada un valor 0,5% superior al valor óptimo del problema original. El límite de iteraciones se ha fijado en 300.

En el siguiente gráfico se muestra la evolución de las soluciones obtenidas:



La primera conclusión que se extrae de los resultados obtenidos es que la primera iteración realizada proporciona un valor de la función objetivo del dual lagrangiano de $6,29312E7$, que coincide exactamente con el valor óptimo del problema original. Tras alcanzar este valor, el algoritmo del subgradiente realiza una serie de saltos que proporcionan unos valores sensiblemente menores de la función objetivo. Con el desarrollo del proceso iterativo la resolución converge al valor de $6,29312E7$ en 273 iteraciones, y al no encontrar soluciones con un valor mayor al señalado el algoritmo se detiene al finalizar las 300 iteraciones.

Mejor solución del dual lagrangiano: $6,29312E7$ unidades monetarias
 Tiempo de ejecución: 19810 segundos

El hecho de que la solución óptima de la función dual lagrangiana y el valor óptimo del problema original coincidan hasta la precisión mostrada parece indicar que en este caso no existe GAP de dualidad. Esta conclusión ha resultado sumamente sorprendente e invita a que en futuras investigaciones se analice detalladamente la estructura del problema, ya que los resultados parecen indicar que la no existencia del GAP de dualidad es una propiedad del modelo.

10.4. Resolución del dual lagrangiano dividido en subproblemas mediante el método el subgradiente

Este esquema corresponde a la implementación del método del subgradiente sobre la división del dual lagrangiano en subproblemas. La secuencia de resultados obtenidos será lógicamente idéntica a los correspondientes al apartado anterior. No obstante se espera que el tiempo de ejecución se reduzca.

Por las razones relativas al equipo informático ya mencionadas anteriormente, en este caso no sólo no se produce una reducción en el tiempo

de ejecución sino que se produce un aumento. Este efecto se debe a que el tiempo añadido que la implementación dedica a definir y eliminar los subproblemas no queda compensado por la ganancia en el tiempo de ejecución, ya que esta se produce de modo secuencial sin paralelización alguna.

Queda por tanto testar dicha implementación en equipos informáticos que permitan resolver subproblemas en modo paralelo.

10.5. Resolución del dual lagrangiano mediante el *Proximal Bundle Method*

La resolución del dual lagrangiano mediante el *Proximal Bundle Method* se apoya en los resultados obtenidos anteriormente.

Por una parte, de la resolución de la relajación lineal sabemos que los valores de las variables duales obtenidas son óptimos. Por tanto no tendrá sentido ejecutar el *Proximal Bundle Method* partiendo de dichos valores, ya que no se encontrara ningún paso de descenso y el punto de análisis permanecerá constante durante todo el proceso iterativo.

En este contexto y teniendo en cuenta que el objetivo de la resolución es valorar el proceso iterativo y la convergencia del *Proximal Bundle Method* aplicado al problema multimercado de oferta óptima, se decide tomar como nulos los valores iniciales de las variables duales, es decir:

$$\mu_{t,j}^F = 0 \quad \forall t \in T, \forall j \in F \quad (57)$$

$$\mu_t^B = 0 \quad \forall t \in T \quad (58)$$

Por otra parte el hecho de conocer el valor óptimo del problema original nos permite definir un criterio de parada preciso. En este caso para el conjunto de resoluciones realizadas el criterio de parada corresponderá a alcanzar el 1% de distancia respecto al valor óptimo de la función objetivo.

Uno de los objetivos prácticos planteados es observar el efecto de la selección del tamaño máximo del *Bundle* en los resultados obtenidos. A este fin se ha decidido realizar las siguientes resoluciones: máximo de 5 pares de elementos, máximo de 10 pares de elementos, máximo de 20 pares de elementos, máximo de 30 pares de elementos y por último un caso de tamaño máximo de *Bundle* ilimitado.

En cada una de las resoluciones planteadas nos fijaremos en cinco aspectos: el número de iteraciones necesarias para alcanzar el criterio de parada, el tiempo de ejecución, la evolución de los resultados obtenidos, la evolución de la longitud de paso y la evolución de la cota inferior.

En las siguientes líneas se indican los resultados obtenidos:

Tamaño *Bundle* = 5 pares de elementos

Número de iteraciones: 155
 Tiempo de ejecución: 21100 segundos

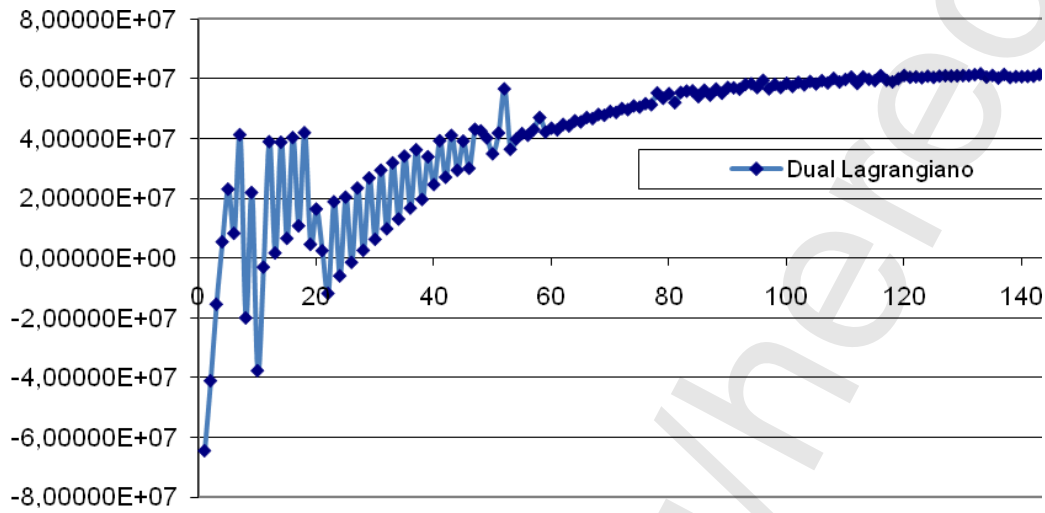


Figura 14 Tamaño de *Bundle* = 5. Valores del dual lagrangiano

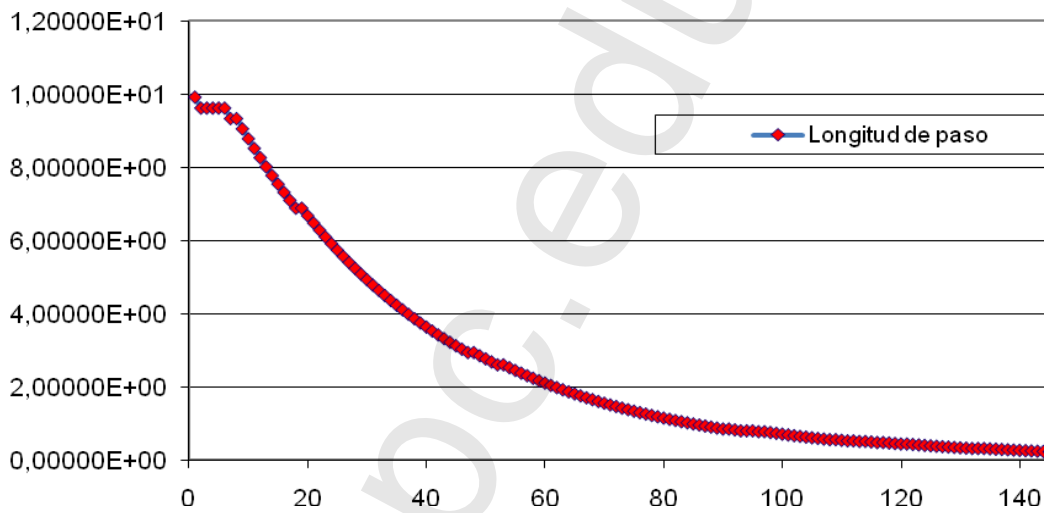


Figura 15 Tamaño de *Bundle* = 5. Longitudes de paso

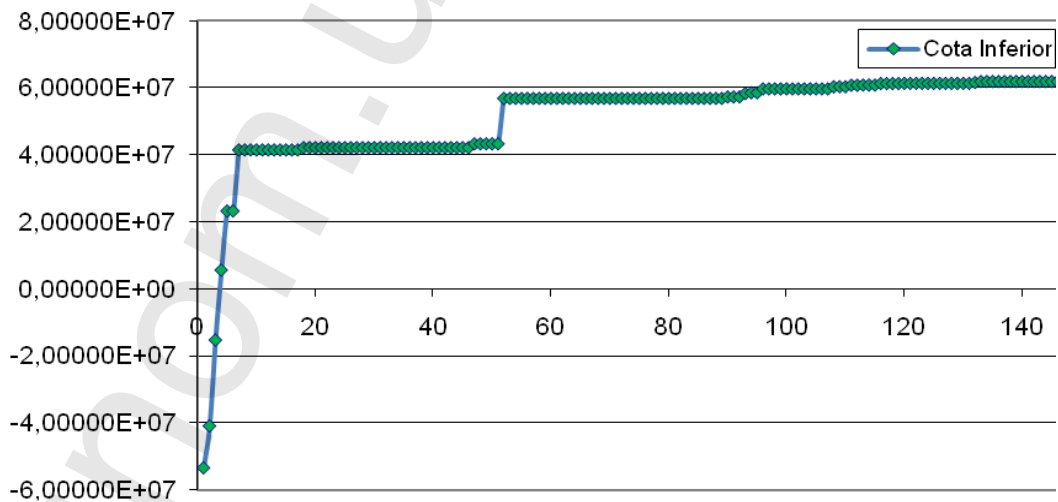


Figura 16 Tamaño de *Bundle* = 5. Cota inferior

Tamaño *Bundle* = 10 pares de elementos

Número de iteraciones: 143
 Tiempo de ejecución: 21824 segundos

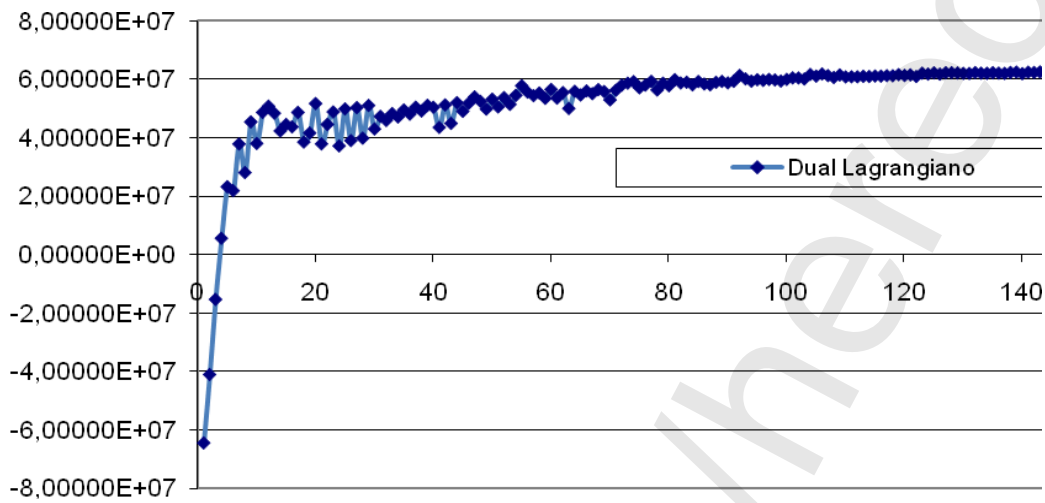


Figura 17 Tamaño de *Bundle* = 10. Valores del dual lagrangiano

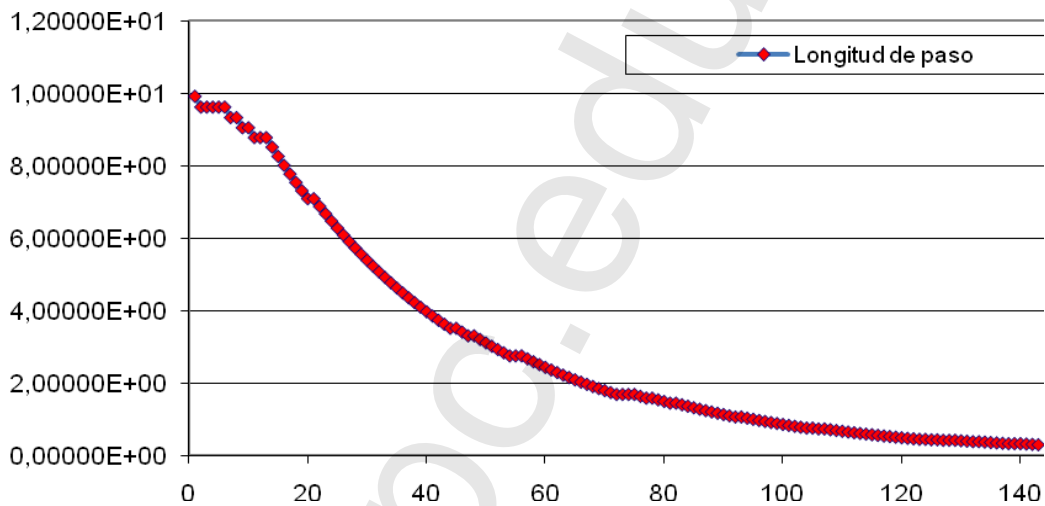


Figura 18 Tamaño de *Bundle* = 10. Longitudes de paso

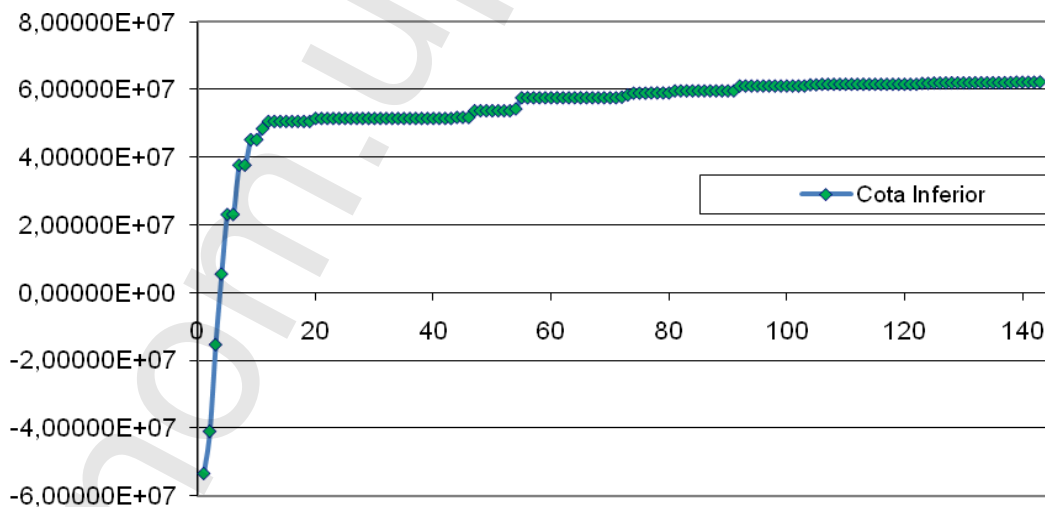


Figura 19 Tamaño de *Bundle* = 10. Cota inferior

Tamaño *Bundle* = 20 pares de elementos

Número de iteraciones: 116
 Tiempo de ejecución: 14538 segundos

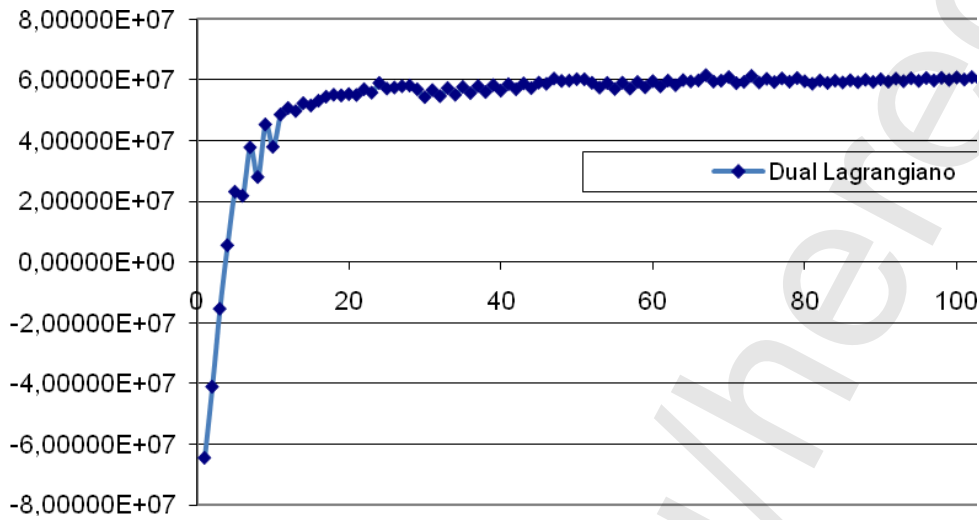


Figura 20 Tamaño de *Bundle* = 20. Valores del dual lagrangiano

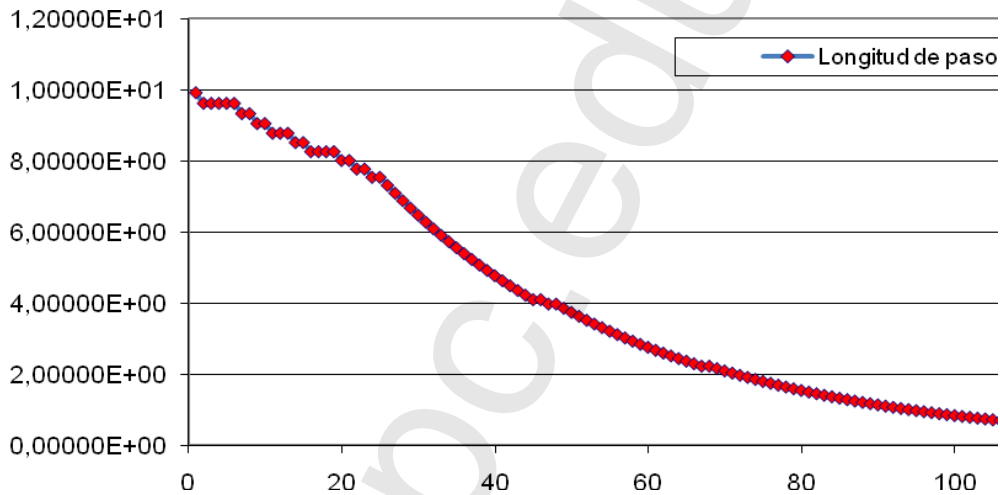


Figura 21 Tamaño de *Bundle* = 20. Longitudes de paso

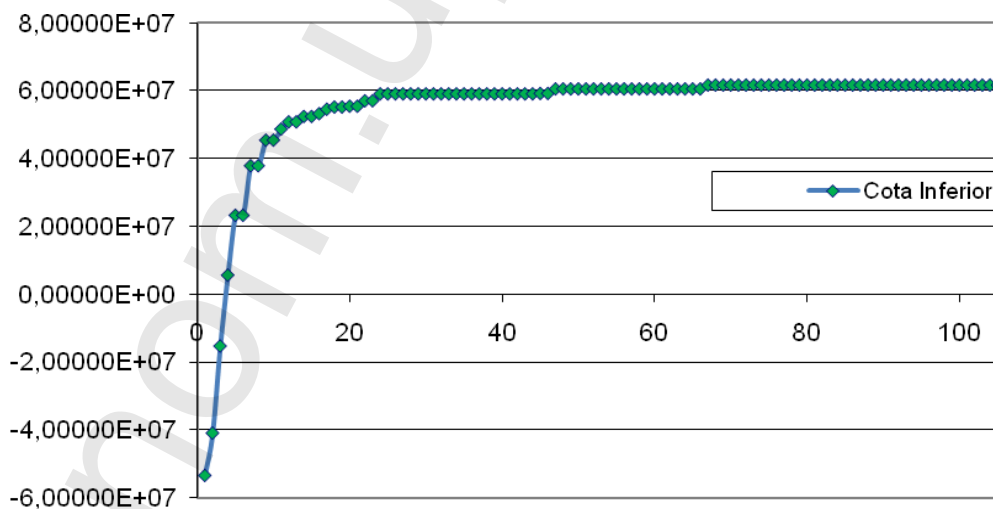


Figura 22 Tamaño de *Bundle* = 20. Cota inferior

Tamaño *Bundle* = 30 pares de elementos

Número de iteraciones: 76
 Tiempo de ejecución: 7698 segundos

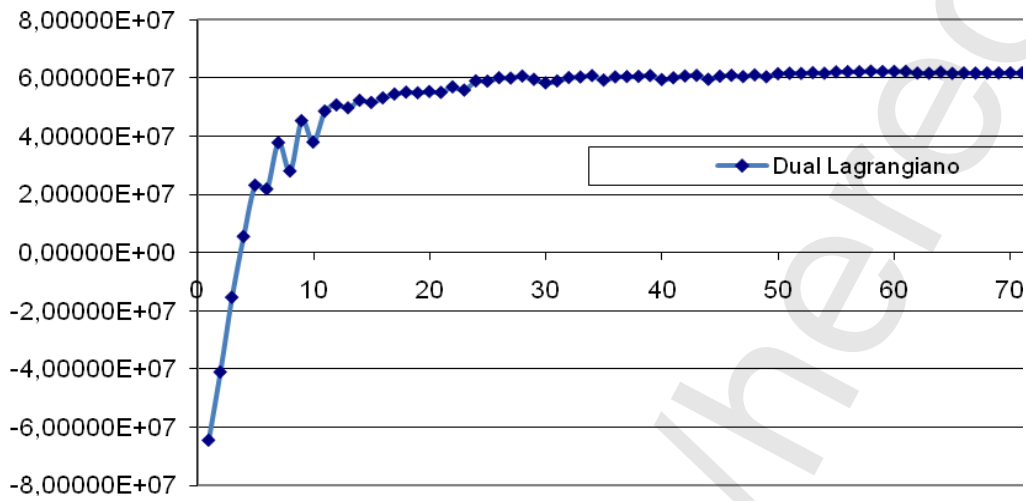


Figura 23 Tamaño de *Bundle* = 30. Valores del dual lagrangiano

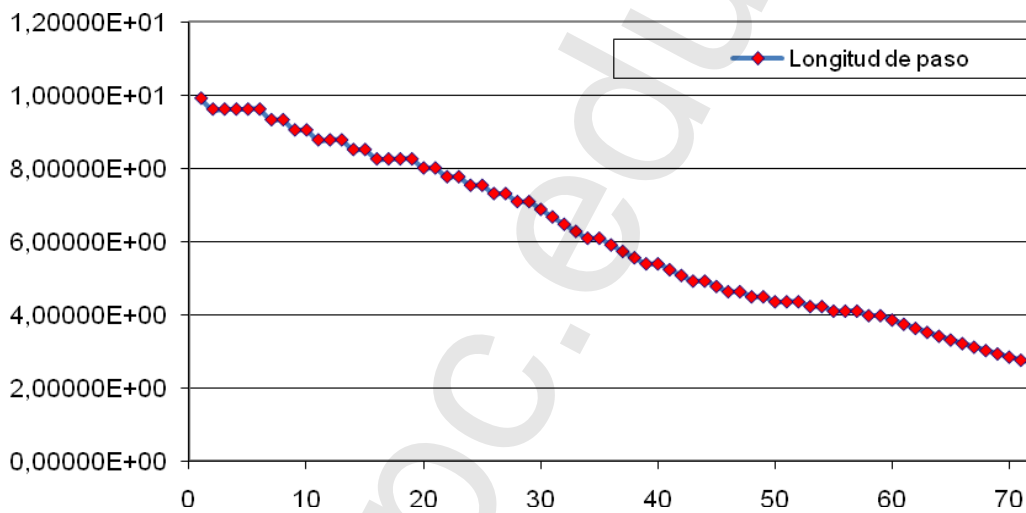


Figura 24 Tamaño de *Bundle* = 30. Longitudes de paso

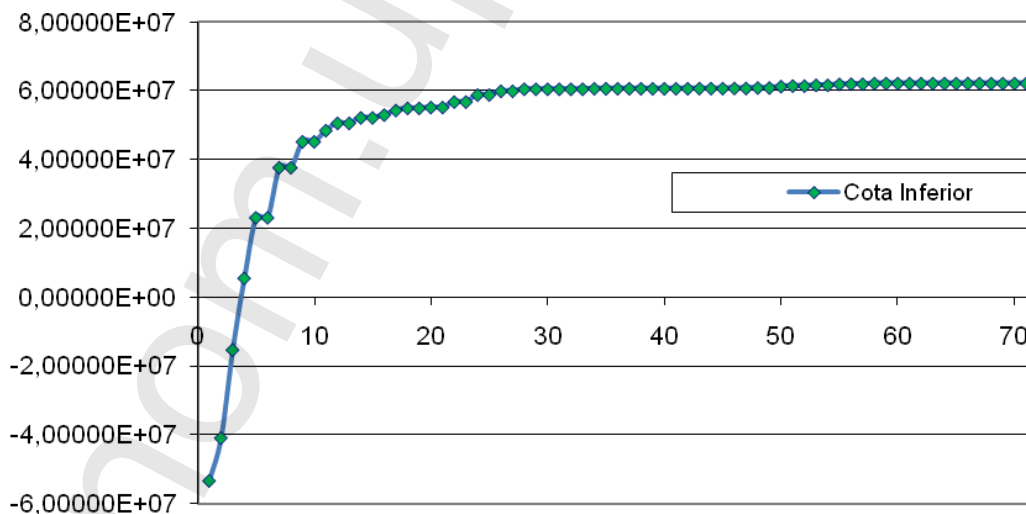


Figura 25 Tamaño de *Bundle* = 30. Cota inferior

Tamaño *Bundle* = Ilimitado

Número de iteraciones: 62
 Tiempo de ejecución: 5535 segundos

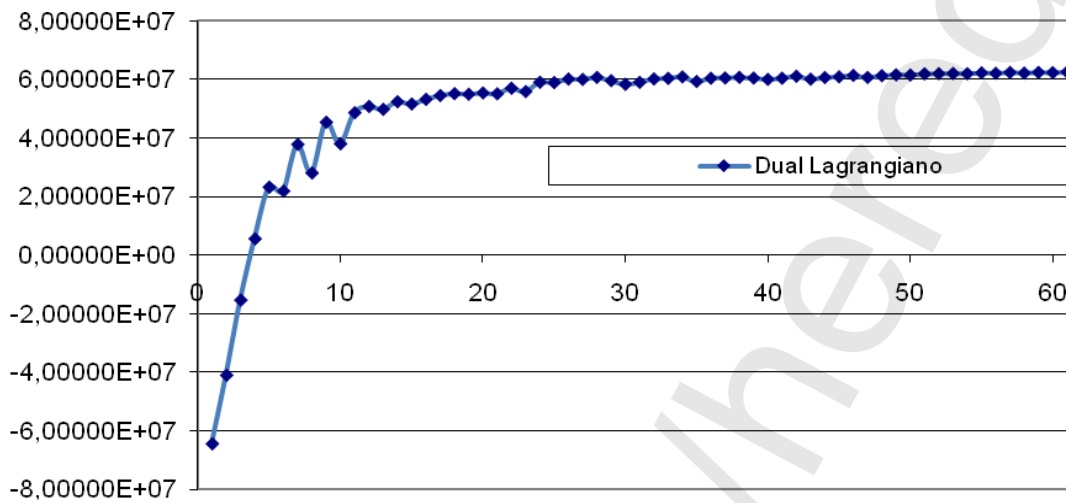


Figura 26 Tamaño de *Bundle* = Ilimitado. Valores del dual lagrangiano

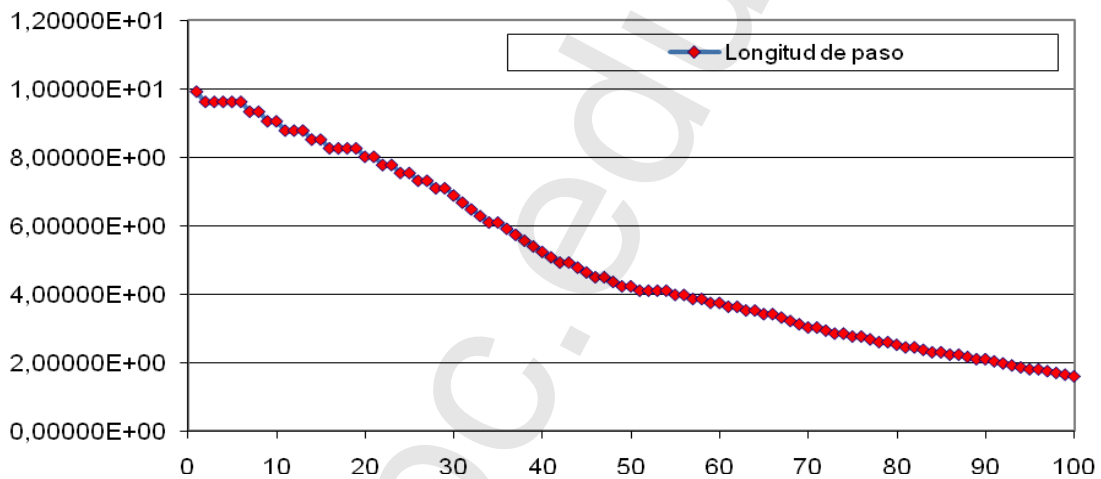


Figura 27 Tamaño de *Bundle* = Ilimitado. Longitudes de paso

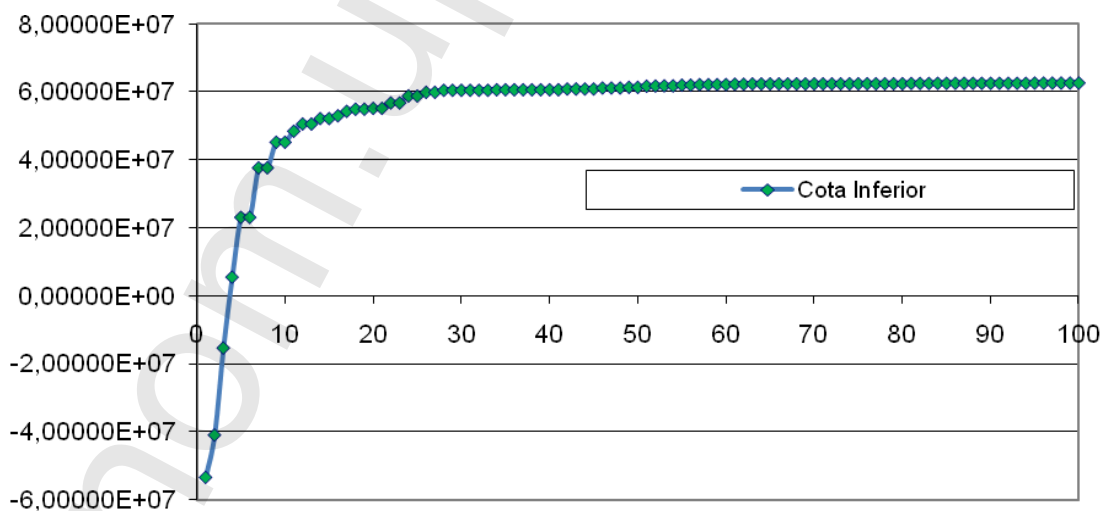


Figura 28 Tamaño de *Bundle* = Ilimitado. Cota inferior

COMPARACIÓN DE RESULTADOS

En las siguientes líneas se realizará un resumen comparativo de los resultados obtenidos.

Tamaño de Bundle	Iteraciones realizadas	Tiempo de ejecución	Tiempo por iteración
5 pares de elementos	155	21100	136,13
10 pares de elementos	143	21824	152,62
20 pares de elementos	116	14538	125,33
30 pares de elementos	76	7698	101,29
Ilimitado	62	5535	89,27
MÉTODO SUBGRADIENTE	300	19810	66,03

Tabla 6. Comparación de resultados entre resoluciones realizadas con Proximal Bundle y con el método del subgradiente

Tal y como se podía preveer, el hecho de aumentar el número de elementos almacenados en el *Bundle* tiene como consecuencia la reducción del número de iteraciones necesarias, ya que la función dual lagrangiana está mejor caracterizada por su aproximación poliédrica cuanto mayor sea el número de elementos almacenados.

No obstante el aspecto que se ha querido valorar mediante esta serie de resoluciones es la penalización que supone en el tiempo de ejecución el aumento de la complejidad del subproblema *Proximal Bundle* debido al aumento de elementos almacenado.

Se aprecia que la relación entre ambos conceptos no es de signo constante, ya que por ejemplo el aumentar el tamaño del *Bundle* de 5 a 10 elementos supone igualmente un aumento en el tiempo de ejecución. No obstante aumentos posteriores acarrear reducciones significativas en el tiempo de ejecución. En concreto el mejor tiempo de ejecución corresponde a la situación de tamaño de *Bundle* ilimitado lo cual hace concluir que la reducción en el número de iteraciones compensa el aumento de complejidad del subproblema. Estas conclusiones se pueden extraer del análisis de tiempo de ejecución por iteración realizada que se incluye en la tabla precedente.

Se espera además que esta conclusión sea igualmente válida para el caso en el que se considere la función de coste cuadrática, ya que la complejidad del subproblema no aumentaría pero sí en cambio el peso de la resolución del dual lagrangiano en el tiempo de ejecución total.

En las representación gráficas incluidas en las páginas precedentes se aprecia cómo la velocidad de disminución de la longitud de paso es más pronunciada cuanto menor sea el número de elementos almacenados en el *Bundle*. Esto se debe a que las oscilaciones en las iteraciones son más comunes y por tanto encuentra menos mejoras de manera consecutiva.

El análisis de la evolución de la cota inferior está directamente ligada a los aspectos mencionados en el párrafo precedente. Cuanto mayor sea el tamaño de *Bundle* se apreciará una menor oscilación en los resultados obtenidos y por tanto la cota inferior crecerá de manera más constante. Esto provoca que la convergencia se realice de manera más directa.

Si bien las dos últimas cuestiones planteadas nos dan una idea del comportamiento iterativo de cada caso, la mejor forma de conocer los resultados obtenidos en centrándonos en los resultados del dual lagrangiano obtenidos iterativamente. Para ello se ha construido un gráfico que permite comparar su evolución iterativa:

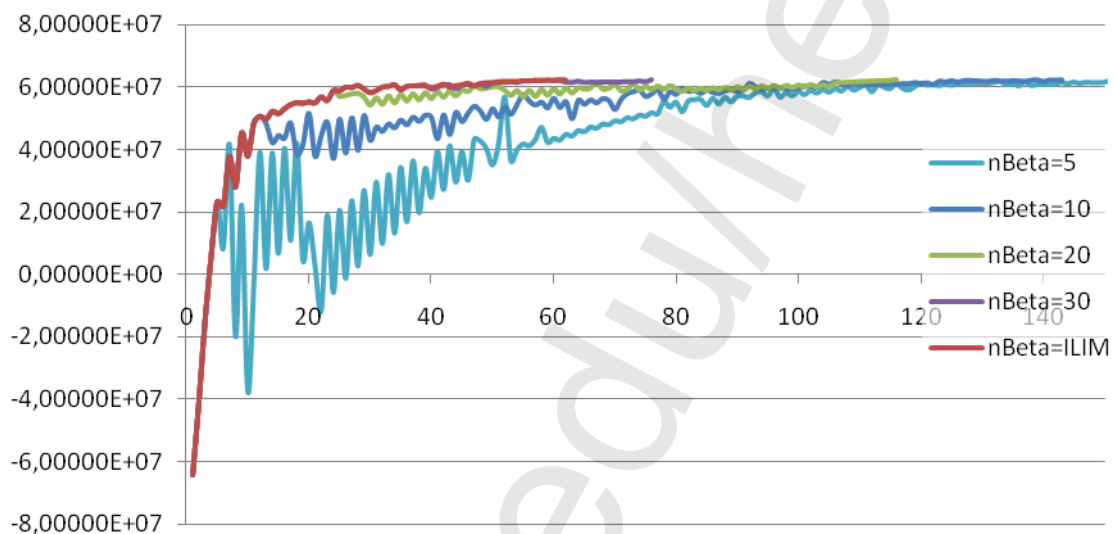


Figura 29 Comparación de los resultados iterativos en el PBM

Vemos que las conclusiones adelantadas sobre las oscilaciones y la rapidez de convergencia se confirman de manera gráfica en la imagen superior. Se observa además que el momento de eliminar los primeros elementos del *Bundle* se traduce en todos los casos con una disminución relativamente brusca respecto a los resultados anteriormente obtenidos. Esto hace pensar que una de las líneas de investigación futuras pueda ser la creación de criterios más sofisticados de elección del elemento a eliminar. Como por ejemplo penalizar la eliminación de subgradientes activos en iteraciones precedentes.

Realizando un zoom podemos apreciar de manera más nítida el efecto advertido anteriormente:

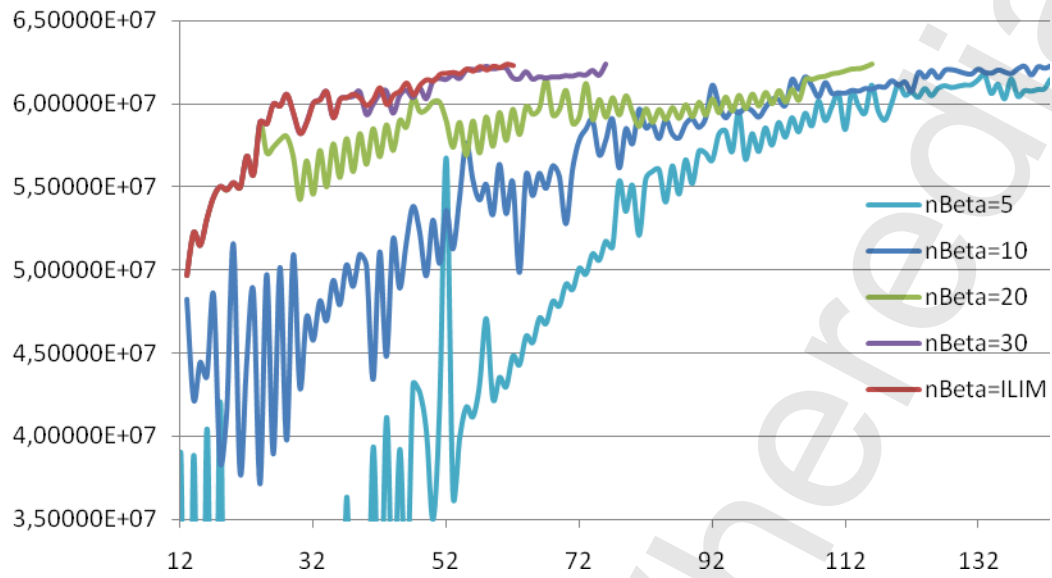


Figura 30 Comparación de los resultados iterativos en el PBM. Zoom

11. PLAN DE TRABAJO

El valor académico del Trabajo de Fin de Master corresponde a 30 créditos ECTS, lo cual equivaldría a 750 horas de dedicación según los estándares habituales.

11.1. Planteamiento inicial

Como referencia inicial de la planificación del trabajo, se definió la siguiente secuencia y duración aproximada de las tareas a realizar:

1. Estudio del modelo multimercado de oferta óptima. **(40 horas)**
2. Estudio y aprendizaje de uso de las librerías CPLEX. **(80 horas)**
3. Implementación y resolución del modelo multimercado con CPLEX: comparativa con AMPL. **(60 horas)**
4. Estudio del método ACCPM. **(40 horas)**
5. Desarrollo de la adaptación del método ACCPM al modelo multimercado: Definición de la estrategia de exploración Branch & Bound y del problema de optimización del oráculo. **(150 horas)**
6. Implementación del método ACCPM en C++ para la resolución del modelo multimercado. **(200 horas)**
7. Comparativa de resultados: CPLEX, Perspective Cuts, ACCPM. **(80 horas)**
8. Elaboración de la memoria. **(100 horas)**

11.2. Desarrollo real

Tal y como suele ser habitual las tareas realizadas han variado respecto a lo previsto por los resultados y conclusiones intermedias obtenidos. En el siguiente esquema se resumen el trabajo realizado:

1. Estudio del modelo multimercado de oferta óptima. **(40 horas)**
2. Puesta a punto del equipo informático. **(10 horas)**
3. Estudio y aprendizaje de *Concert Technology*. **(120 horas)**
4. Implementación y resolución del modelo multimercado con CPLEX: comparativa con AMPL. **(80 horas)**
5. Estudio del método ACCPM. **(40 horas)**
6. Estudio del método *Proximal Bundle Method* y su adaptación al problema multimercado de oferta óptima **(60 horas)**
7. Resolución de la relajación lineal del modelo multimercado. **(10 horas)**
8. Resolución del problema dual lagrangiano asociado al problema multimercado mediante el método del subgradiente. **(80 horas)**
9. Adaptación a la división en subproblemas de la implementación referida al método del subgradiente. **(20 horas)**
10. Implementación del método *Proximal Bundle Method* en C++ para la resolución del modelo multimercado. **(230 horas)**
11. Comparativa de resultados: CPLEX, relajación lineal, método del subgradiente, *Proximal Bundle Method*. **(40 horas)**
12. Elaboración de la memoria. **(100 horas)**

12. CONCLUSIONES

El presente Trabajo de Fin de Master ha cumplido su objetivo principal de recorrer el camino que comienza con el análisis de un modelo de optimización y termina con el análisis de los resultados prácticos referidos al problema multimercado de oferta óptima.

En concreto, el trabajo se ha sumado al proceso de análisis crítico del modelo multimercado de oferta óptima, aportando nuevos resultados que han ayudado a su formulación definitiva. De la comprensión del modelo se deriva la selección de posibles métodos de resolución alternativos e hipotéticamente más eficientes que el uso del solver *CPLEX*. En este contexto se han analizado los aspectos positivos y negativos de los métodos *ACCPM* y *PBM*.

Uno de los pilares del presente trabajo ha consistido en el profundo análisis de las características del método *PBM* así como su potencial adaptación a la resolución del modelo multimercado de oferta óptima. En este contexto se ha procedido a analizar y entender el funcionamiento del algoritmo, su aplicación analítica a un ejemplo de pequeña escala y a la formulación de su aplicación al modelo multimercado de oferta óptima.

El segundo pilar corresponde a la implementación del método *PBM* mediante *C++* y su posterior enlace con *CPLEX* utilizando el metalenguaje *Concert Technology*. Este proceso supone una estrategia de implementación para esta familia de problemas de interés para el grupo *GNOM*, ya que hasta la fecha se ha venido utilizando *AMPL* como lenguaje base en la mayor parte de las ocasiones. La implementación ha proporcionado además resultados satisfactorios en términos de convergencia. Del análisis de los resultados y de las futuras mejoras en la implementación podrá surgir una alternativa real al solver *CPLEX* a la hora de resolver este tipo de problemas.

Podemos concluir por tanto que el objetivo principal del presente trabajo ha sido abordado de manera satisfactoria. Para ampliar el análisis realizado en las siguientes líneas se resumen las reflexiones y conclusiones principales extraídas de la realización del presente trabajo:

- Si bien en la literatura especializada se pueden encontrar ejemplos en los que se ha utilizado el método *ACCPM* para resolver problemas mixtos enteros, no hay evidencia significativa de que su rendimiento algorítmico sea mejor que el de otros métodos de la familia de los métodos duales.

Además el usuario tiene que definir un problema de optimización denominado “*oráculo*” que tiene un gran incidencia en el resultado final. Si a la falta de conocimientos sobre las características que el “*oráculo*” debe presentar le añadimos la dificultad de implementación que el conjunto del método acarrea, concluimos que resulta conveniente considerar otros métodos duales de resolución.

- En el artículo de *Borghetti et al.* [17] se plantea resolver un problema de conmutación de unidades térmicas e hidráulicas mediante una adaptación del *Proximal Bundle Method*. Recogiendo esa idea se decide implementar el *Proximal Bundle Method* propuesto por Hiriart-Urruty y Lemaréchal. A falta de poder añadir nuevos elementos que mejoren el rendimiento algorítmico de la implementación realizada, podemos concluir que los resultados obtenidos satisfacen los objetivos marcados.
- La adecuación del metalenguaje *Concert Technology* al desarrollo de este proyecto ha sido desigual. En un primer momento permitió la rápida familiarización con el entorno de enlace entre C++ y *CPLEX* gracias a los pequeños ejemplos encontrados en la literatura. Esto permitió obtener los primeros resultados en un relativamente breve espacio de tiempo.

No obstante cuando las implementaciones a realizar aumentaron en complejidad este entorno empezó a mostrar sus primeras desventajas prácticas. La conclusión principal es que **Concert Technology no se adapta satisfactoriamente a esquemas iterativos**, ya que su estructura de dos bloques exige elementos estáticos. La necesidad de actualizar constantemente ambos bloques hace que el tiempo computacional aumente considerablemente.

La adopción del lenguaje C++ se basaba principalmente en su versatilidad. En este contexto parece que el trabajo abordado excede del campo de trabajo para el que *Concert Technology* ha sido diseñado y no se obtiene la ganancia esperada en versatilidad.

- Teniendo en cuenta la simplicidad de los ejemplos disponibles en la literatura relacionada con el metalenguaje *Concert Technology*, este trabajo puede servir como base para futuras implementaciones en este entorno. Podría incluso derivarse en una guía de usuario mediante una adecuación del formato y la inclusión de comentarios y sugerencias.
- Las implementaciones realizadas en este trabajo han sido de la ayuda al *GNOM* a la hora de detectar algunas mejoras de modelización del planteamiento inicial.
- El resultado óptimo obtenido mediante la relajación lineal corresponde a un valor muy cercano al óptimo del problema original. Tanto es así que tomando el valor de las variables duales asociadas al resultado óptimo de la relajación lineal como valores iniciales del problema del subgradiente, el resultado obtenido en una única iteración corresponde al óptimo del problema original. Este resultado invita a que en futuras investigaciones se analice detalladamente la estructura del problema, ya que los resultados parecen indicar que la no existencia del GAP de dualidad es una propiedad del modelo.
- Utilizando el mismo equipo informático y sin aprovechar la divisibilidad del dual lagrangiano en subproblemas, el algoritmo *Proximal Bundle*

Method muestra un tiempo de ejecución significativamente inferior al correspondiente al método del subgradiente.

- Los resultados obtenidos muestran que para la escala de problema considerado el tamaño máximo de *Bundle* que presenta mayor eficiencia computacional corresponde a la situación de tamaño ilimitado.

12.1. Futuras líneas de investigación

El trabajo desarrollado pretende ser la base metodológica de una estructura de resolución del problema multimercado de oferta óptima computacionalmente eficiente. A este respecto se proponen las siguientes líneas de mejora e investigación con el objetivo de alcanzar dicho estado:

- Una vez verificada la convergencia del *Proximal Bundle Method* implementado se propone ampliar el código creado incluyendo un **método heurístico de recuperación de factibilidad**. La razón reside en que el resultado óptimo encontrado en la resolución del dual lagrangiano no será, en general, una solución factible del problema multimercado de oferta óptima. En este contexto se sugiere el análisis del método heurístico de recuperación de factibilidad propuesto por *Borghetti et al* en [17].
- La dificultad de acceso a equipos los servidores de cálculo intensivo del grupo *GNOM* durante el desarrollo del proyecto limita el análisis del tiempo computacional empleado a la hora de resolver cada uno de los modelos planteados. Por lo tanto se ve la necesidad de ejecutar estos mismos código en **equipos multinúcleo**, con especial relevancia a la hora de valorar la ganancia en tiempo computacional de la **división en subproblemas**.
- Tal y como se ha mencionado en el apartado de conclusiones, el metalenguaje *Concert Technology* no se ha adaptado satisfactoriamente a los esquemas de implementación planteados. Así, se propone utilizar otro tipo de herramientas como por ejemplo el entorno *Callable Library* para el enlace de *C++* y *CPLEX* o por el contrario valorar la posibilidad de utilizar un *solver* distinto a *CPLEX* como *COIN*, con la ventaja de que este último *solver* es libre y no necesita licencia.
- Para poder contar con una implementación del algoritmo *PB* eficiente es preciso explotar la separabilidad por unidades térmicas del subproblema Lagrangiano y su resolución computacional en paralelo. Una idea prometedora, cuyo estudio ya se ha iniciado, consiste en definir el subproblema asociado a cada térmica como un problema de caminos mínimos, en la línea de lo sugerido en [30]
- Se observa que en el momento de comenzar a eliminar elementos de *Bundle* se produce una disminución relativamente brusca respecto a los resultados anteriormente obtenidos. Por lo tanto una de las líneas de

investigación futuras corresponde a la creación de criterios más sofisticados de elección del elemento a eliminar. Como por ejemplo penalizar la eliminación de subgradientes activos en iteraciones precedentes.

- Los análisis y las implementaciones realizadas tanto del método del subgradiente como del *Proximal Bundle Method* se han basado en el conocimiento del valor óptimo de la resolución del modelo multimercado de oferta óptima. Esto se ha hecho así ya que el objetivo inicial era valorar su validez y convergencia. Una vez finalizada esta fase se tendrá que extender la implementación para casos en los que no se conozca a priori el resultado óptimo. Para ello se propone calcular de manera rápida una solución factible del modelo multimercado de oferta óptima que este relativamente ajustada.

13. BIBLIOGRAFIA

- [01] Goffin, J.L; Vial, J.F, Convex Nondifferentiable Optimization: A Survey Focused on the Analytic Center Cutting Plane Method. University of Geneva HEC/Logilab Technical Report 99.02. 1999.
- [02] Goffin, J.L; Vial, J.F, Multiple cuts in the analytic center cutting plane method. University of Geneva HEC/Logilab Technical Report 98.10. 1999.
- [03] Goffin, J.L; Vial, J.F, Multiple Cuts with a Homogeneous Analytic Center Cutting Plane Method. Computational Optimization and Applications 24(1), p. 37-61. 2004.
- [04] Goffin, J.L; Vial, A tutorial on ACCPM. University of Geneva HEC/Logilab. 2001
- [05] Heredia, F.J; Corchero C., 2010. Optimal Day-Ahead Bidding in the MIBEL's Multimarket Energy Production System. Proceedings of the 7 *International Conference on the European Energy Market*. June 23-25, 2010. Madrid. Spain (pendent de publicació als IEEE Xplore). <http://www.eem10.com/>
- [06] Frangioni, A., Gentile, C., Perspective cuts for 0–1 mixed integer programs. Technical report 577, Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti" (IASI-CNR), 2002.
- [07] Frangioni, A., Gentile, C., Perspective cuts for a class of convex 0–1 mixed integer programs. *Mathematical Programming* 106, p. 225-236. 2005.
- [08] Frangioni, A., Gentile, C., SDP diagonalizations and perspective cuts for a class of Nonseparable MIQP. *Operations Research Letters* 35(2), p. 181 - 185, 2007
- [09] CPLEX, 2008. CPLEX Optimization subroutine library guide and reference. Version 11.0 CPLEX Division, ILOG Inc., Incline Village, NV, USA.
- [10] Fourer, R., Gay, D.M., Kernighan, B.W., 2003. AMPL: A modelling language for mathematical programming. CA: Brooks/Cole-Thomson Learning, 2nd edn.
- [11] J.-L. Goffin, J. Gondzio, R. Sarkissian and J.-P. Vial, Solving Nonlinear Multicommodity Flows Problems by the Analytic Center Cutting Plane Method", *Mathematical Programming, Series B*, vol 76 1, 131-154. 1997
- [12] J. Gondzio, R. Sarkissian and J.-P. Vial, Using an Interior Point Method for the Master Problem in a Decomposition Approach, Technical Report *European Journal of Operational Research*, volume 101,577-587. 1997
- [13] A. Ouorou, P. Mahey and J.-Ph. Vial, A survey of algorithms for convex multicommodity flow problems, HEC Working Paper 97.13, Department of Management Studies, University of Geneva, Switzerland, June 1997.
- [14] O. Bahn, O. du Merle, J.-L. Goffin and J.P. Vial , A Cutting Plane Method from Analytic Centers for Stochastic Programming", *Mathematical Programming, Series B*, 69, 45-73. 1995
- [15] E. Fragnière, J. Gondzio and J.-P. Vial, A Planning Model with one Million Scenarios Solved on an Affordable Parallel Machine. Logilab

- Technical Report 98.11, Department of Management Studies, University of Geneva, Switzerland, June, 1998.
- [16] O. du Merle, J.-L. Goffinn, C. Trouiller and J.-P. Vial, A Lagrangian Relaxation of the Capacitated Multi-Item Lot Sizing Problem Solved with an Interior Point Cutting Plane Algorithm, Logilab Technical Report 97.5, Department of Management Studies, University of Geneva, Switzerland, April 1997.
- [17] A. Borghetti, A. Frangioni, F. Lacalandra and C.A Nucci. Lagrangian Heuristics Based on Disaggregated Bundle Methods for Hydrothermal Unit Commitment. IEEE Transactions on Power Systems, Vol. 1, No. 18 2003.
- [18] J. B. Hiriart-Urruty, C. Lemaréchal, Convex Analysis and Minimization Algorithms II – Advanced Theory and Bundle Methods, ser. A Series of Comprehensive Studies in Mathematics. New York: Springer-Verlag, 1993, vol. 306.
- [19] A. Geoffrion, Lagrangian relaxation for interger programming. Math. Progr. Study, vol.2, pp. 82-114, 1974.
- [20] A. Merlin and P. Sandrin, “A new method for unit commitment at Electricité de France” IEEE Trans. Power Appar. Syst., vol. PAS-102, pp. 1218-1225, 1983.
- [21] S. Takriti, J. R. Birge, and E. Long, “A stochastic model for unit commitment” IEEE Trans. Power Syst., vol. 11, pp. 1497-1508, Aug. 1996.
- [22] N. Jiménez Redondo, A. J. Conejo, Short-term hydro-thermal coordination by Lagrangian relaxation-solution of the dual problems, IEEE Trans. Power Syst. vol. 14, pp. 89-95, Feb. 1999.
- [23] M. Madrigal and V. H. Quintana, An Interior-Point/Cutting-Plane method to solve unit commitment problems, IEEE Trans. Power Syst. vol. 15, pp. 1022-1027, Aug. 2000.
- [24] C. Lemaréchal, F. Pellegrino, A. Renaud, and C. Sagastizábal, Bundle methods applied to the unit commitment problem” in Proc. 17th IFIP – Conf. Syst. Modeling and Optimization, Prague, Czechoslovakia, July 10-14, 1995.
- [25] L. Bacaud, C. Lemaréchal, A. Renaud, and C. Sagastizábal. Bundle methods in stochastic optimal power management: A disaggregated approach using preconditioners” Comput. Optim. Appl., vol. 20, pp. 227-244, 2001.
- [26] P. B. Luh, D. Zhang, and R. N. Tomastik, “An algorithm for solving the dual problem of hidrothermal scheduling” IEEE Trans. Power Syst. vol. 13, no. 2, pp. 593-600, 1998.
- [27] J. Batut and A. Renaud, Daily generation scheduling optimization with transmission constraints: A new class of algorithms. IEEE Trans. Power Syst. vol. 7, pp.982-989, Aug. 1992.
- [28] C. Beltran Royo, Generalized unit commitment by the radar multiplier method. Ph. D. dissertation, Universitat Politècnica de Catalunya, 2001.
- [29] A. Frangioni, Generalized Bundle Methods” SIAM J. Optim, to be published.
- [30] Fan W., Guan X, Zhai, Q. A new method for unit commitment with ramping constraints. Electric Pow Syst Res. 2002; 62:215-24.

ANEXO A. ÍNDICE DE ACRÓNIMOS

ACCPM	Analytic Center Cutting Plane Method
AMPL	Modelling Language for Mathematical Programming
BC	Bilateral Contracts
DAM	Day-Ahead Market
FC	Futures Contracts
IM	Intraday Markets
MIBEL	Mercado Ibérico de la Electricidad
TFM	Trabajo de Fin de Master
PBM	Proximal Bundle Method
RM	Reserve Market