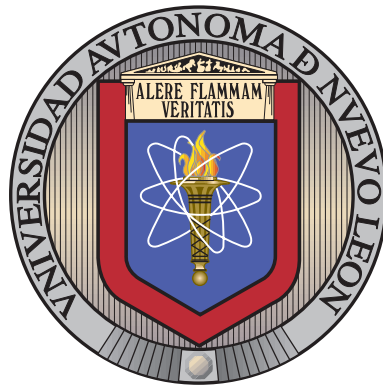


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

DIVISIÓN DE ESTUDIOS DE POSGRADO



PROBLEMAS CON MÚLTIPLES RELAJACIONES  
LAGRANGIANAS

POR

EDITH LUCERO OZUNA ESPINOSA

EN OPCIÓN AL GRADO DE

MAESTRÍA EN CIENCIAS

EN INGENIERÍA DE SISTEMAS

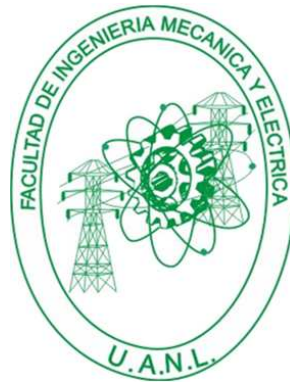
SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN

NOVIEMBRE 2008

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

DIVISIÓN DE ESTUDIOS DE POSGRADO



PROBLEMAS CON MÚLTIPLES RELAJACIONES  
LAGRANGIANAS

POR

EDITH LUCERO OZUNA ESPINOSA

EN OPCIÓN AL GRADO DE

MAESTRÍA EN CIENCIAS

EN INGENIERÍA DE SISTEMAS

SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN

NOVIEMBRE 2008

**Universidad Autónoma de Nuevo León**  
**Facultad de Ingeniería Mecánica y Eléctrica**  
**División de Estudios de Posgrado**

Los miembros del Comité de Tesis recomendamos que la Tesis «Problemas con Múltiples Relajaciones Lagrangianas», realizada por la alumna Edith Lucero Ozuna Espinosa, con número de matrícula 1117587, sea aceptada para su defensa como opción al grado de Maestría en Ciencias en Ingeniería de Sistemas.

El Comité de Tesis

---

Dr. Igor S. Litvinchev

Asesor

---

Dr. César Emilio Villarreal Rodríguez

Revisor

---

Dr. Fernando López Irarragorri

Revisor

Vo. Bo.

---

Dr. Moisés Hinojosa Rivera

División de Estudios de Posgrado

San Nicolás de los Garza, Nuevo León, noviembre 2008

*A mi familia, que siempre han creído en mí y me han apoyado, sin ellos nada de esto hubiera podido concretarse, a ellos junto con todo mi amor y agradecimiento.*

# ÍNDICE GENERAL

---

<b>Agradecimientos</b>	<b>XII</b>
<b>Resumen</b>	<b>XIII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Estructura de la Tesis . . . . .	1
1.2. Estructuras especiales . . . . .	2
1.3. Motivación . . . . .	3
1.4. Objetivos . . . . .	3
1.5. Hipótesis . . . . .	3
1.6. Metodología . . . . .	4
<b>2. Antecedentes</b>	<b>5</b>
2.1. Relajación lagrangiana . . . . .	5
2.1.1. Relajaciones de problemas de optimización . . . . .	7
2.1.2. Relajación lagrangiana ( $LR$ ) . . . . .	8
2.1.3. Solución factible lagrangiana . . . . .	11
2.1.4. Interpretación geométrica . . . . .	12

---

2.1.5.	Construcción de una relajación lagrangiana . . . . .	16
2.1.6.	Las características de la función lagrangiana . . . . .	23
2.2.	Métodos primales y duales para resolver relajación dual . . . . .	26
2.2.1.	Método del subgradiente . . . . .	27
2.2.2.	Métodos duales ascendentes . . . . .	29
2.2.3.	Método generación de restricciones . . . . .	30
2.2.4.	Generación de columnas (CG) . . . . .	30
2.2.5.	Métodos de funciones Lagrangianas modificadas. . . . .	33
2.2.6.	Métodos híbridos de dos fases . . . . .	35
2.2.7.	Descomposición de subproblemas . . . . .	37
2.2.8.	Relajar y cortar (“Relax & cut”) . . . . .	40
2.2.9.	Heurísticas lagrangianas y Branch-and-Price . . . . .	44
<b>3.</b>	<b>Construcción de cotas</b>	<b>47</b>
3.1.	Construcciones básicas . . . . .	47
3.2.	Algoritmo . . . . .	53
3.3.	Aplicación al Problema de Mochila Multidimensional . . . . .	55
3.3.1.	Formulación del problema MKP . . . . .	55
3.3.2.	Definición de relajaciones . . . . .	56
3.4.	Aplicación al Problema de Asignación Generalizada . . . . .	56
3.4.1.	Formulación del problema GAP . . . . .	57
3.4.2.	Definición de relajaciones . . . . .	58

---

<b>4. Experimentación Computacional</b>	<b>59</b>
4.1. Generalidades . . . . .	59
4.2. Resultados Experimentales MKP . . . . .	59
4.2.1. Análisis estadístico de resultados MKP . . . . .	64
4.3. Resultados Experimentales GAP . . . . .	67
4.3.1. Análisis estadístico de resultados GAP . . . . .	69
<b>5. Conclusiones</b>	<b>73</b>
5.1. Contribuciones . . . . .	73
5.2. Trabajo a futuro . . . . .	73
<b>A. Resultados</b>	<b>75</b>
<b>B. GAMS</b>	<b>87</b>
B.1. Generalidades . . . . .	87
B.1.1. Generalidades de GAMS . . . . .	87
B.1.2. Generalidades de Cplex . . . . .	89
B.1.3. GAMS/Cplex . . . . .	91
<b>C. Programación lineal</b>	<b>92</b>
C.1. Breve historia . . . . .	92
C.2. El problema general de la programación lineal . . . . .	93
C.2.1. Hipótesis de la programación lineal . . . . .	94
C.2.2. Manipulaciones del problema general . . . . .	95

---

C.3. Representación matricial de un problema lineal . . . . .	96
C.3.1. Formas canónica y estándar . . . . .	96
<b>D. Programación entera</b>	<b>98</b>
D.1. Breve descripción . . . . .	98
D.2. Clasificación y formulaciones . . . . .	98
D.3. Algoritmo de ramificar y acotar . . . . .	100
D.3.1. Divide y vencerás . . . . .	100
D.3.2. Enumeración implícita . . . . .	100
D.3.3. Algoritmo de Ramificar y Acotar (maximizar) . . . . .	101



# ÍNDICE DE FIGURAS

---

2.1. Interpretación geométrica de la descomposición[45] . . . . .	17
2.2. Función lagrangiana par el caso de maximización [45] . . . . .	24
2.3. Contorno de la función lagrangiana [45] . . . . .	26
2.4. Definición de un poliedro acotado [45] . . . . .	37
2.5. Conjunto convexo entero de soluciones que no cambia [45] . . . . .	42

# ÍNDICE DE TABLAS

---

4.1. Resultados para casos pequeños MKP . . . . .	61
4.2. Resultados para casos medianos MKP . . . . .	62
4.3. Resultados para casos grandes MKP . . . . .	63
4.4. Algunos otros resultados MKP . . . . .	63
4.5. Cálculo de medias para diferentes tamaños del problema MKP . . . . .	64
4.6. Resultados para casos pequeños GAP . . . . .	68
4.7. Resultados para casos medianos GAP . . . . .	69
4.8. Cálculo de medias para diferentes tamaños del problema GAP . . . . .	70
A.1. Resultados MKP tamaño 5x3 . . . . .	76
A.2. Resultados MKP tamaño 10x5 . . . . .	78
A.3. Resultados MKP tamaño 20x10 . . . . .	80
A.4. Resultados MKP tamaño 40x20 . . . . .	82
A.5. Resultados MKP tamaño 60x30 . . . . .	83
A.6. Resultados MKP tamaño 100x50 . . . . .	83
A.7. Resultados GAP tamaño 5x3 . . . . .	84

---

A.8. Resultados GAP tamaño 10x5 . . . . .	85
A.9. Resultados GAP tamaño 20x5 . . . . .	86

# AGRADECIMIENTOS

---

Primeramente a Dios por darme la oportunidad de estar aquí, y sobre todo por poner en mi camino a las personas con las que he compartido mi vida, mis metas, mis sueños.

A mis padres y mi hermana que han sido la base de todos mis logros, sin ellos no habría podido llegar a donde estoy ahora, por su apoyo incondicional, su amor, su confianza, por ayudarme a ser la persona que soy, por estar conmigo siempre.

A mis amigos, que han estado conmigo a lo largo de este camino, que me han ayudado aún en los malos tiempos, por ser como son y por apoyarme incondicionalmente.

A el Dr. Igor Litvinchev por todas sus enseñanzas, su apoyo, su confianza, por creer en mí y compartir su tiempo y conocimientos conmigo. A el Dr. Miguel Mata por toda la ayuda que me brindó a lo largo de este proyecto, a la M.C Jania Saucedo por estar siempre ahí para ayudarme y apoyarme. A el Dr. César Villarreal y el Dr. Fernando López por el tiempo que le dedicaron a mi trabajo, por sus comentarios y su ayuda, a la Dra. Socorro Rangel por su tiempo, por recibirme, por toda su ayuda.

A todo el cuerpo académico de PISIS que siempre está dispuesto a ayudarte si lo necesitas, aconsejarte, explicarte, por siempre estar ahí, por ser más que maestros para nosotros.

A CONACYT, FIME y la UANL por el apoyo económico que nos brindan para poder realizar este posgrado.

# RESUMEN

---

Edith Lucero Ozuna Espinosa.

Candidato para el grado de Maestro en Ciencias  
con especialidad en Ingeniería de Sistemas.

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Título del estudio:

## PROBLEMAS CON MÚLTIPLES RELAJACIONES LAGRANGIANAS

Número de páginas: 111.

**OBJETIVOS Y MÉTODO DE ESTUDIO:** La mayoría de los problemas de optimización a gran escala cuentan con una estructura especial que podemos explotar para encontrar técnicas de solución más eficientes. Una de las estructuras más generales y comunes es en la que se pueden diferenciar las restricciones de un problema en *fáciles* y *difíciles*. En otras palabras, el problema se torna “fácil” de resolver si logramos eliminar las restricciones difíciles. Un ejemplo típico, es el problema con la estructura de restricciones tipo “escalón” que se descompone en subproblemas independientes relajando las restricciones de acoplamiento (“difíciles”).

Una manera conocida de explotar dicha estructura es relajar de forma lagrangiana las restricciones difíciles. Esto es, cuando relajamos las restricciones difíciles

se agrega un término de penalización a la función objetivo para prevenir la violación de restricciones relajadas. Usualmente, la penalización es una combinación lineal correspondiente a holguras de restricciones difíciles tomadas con los coeficientes llamados multiplicadores de Lagrange. Dado un vector de multiplicadores fijo y resolviendo el problema lagrangiano, obtendremos una cota superior (para problemas de maximización) del valor objetivo óptimo original. El problema dual lagrangiano consiste en encontrar la mejor cota, es decir, buscar los multiplicadores lagrangianos que minimizan la cota.

Existen diversas maneras en las que un problema puede ser relajado de forma lagrangiana. Por lo tanto, podemos tener varias cotas lagrangianas para el mismo problema. Así aparece el problema de cómo escoger la mejor cota lagrangiana. Pretendemos formular un solo problema para encontrar el valor de la mejor cota lagrangiana sin calcular todas las cotas existentes.

CONTRIBUCIONES Y CONCLUSIONES: La principal contribución es la obtención de la mejor cota lagrangiana así como la implementación computacional del algoritmo propuesto.

También presentamos la implementación del algoritmo en problemas de mochila multidimensional y de asignación generalizada resueltos por medio del método de descomposición de Benders. Además de el análisis estadístico de los resultados de cada problema y las comparaciones de las cotas obtenidas contra el valor óptimo y el valor de la relajación lineal.

Algunos resultados de este trabajo de tesis fueron publicados como sigue:

- “Problemas con múltiples relajaciones Lagrangianas”, 6° Congreso Internacional de Ciencia y Tecnología (CIINDET), Cuernavaca, Morelos, México, octubre 2008.
- “Algoritmo de búsqueda de la mejor cota Lagrangiana aplicado al problema de asignación generalizada”, XXI Congreso Nacional y VII Congreso Internacional

de Informática y Computación, Monterrey, Nuevo León México, octubre 2008.

Y se presentaron en los siguientes foros:

- 6° Congreso Internacional de Ciencia y Tecnología (CIINDET), Cuernavaca, Morelos, México, octubre 2008.
- XXI Congreso Nacional y VII Congreso Internacional de Informática y Computación, Monterrey, Nuevo León, México, octubre 2008.
- XLI Congreso Nacional de la Sociedad Matemática Mexicana, Valle de Bravo, Estado de México, México, octubre 2008.
- XVIII Escuela Nacional de Optimización y Análisis Numérico (ENOAN), Saltillo Coahuila, México, abril 2008.
- Ciclo de Seminarios de la Universidad Estatal Paulista campus S.J. do Rio Preto, SP, Brasil, diciembre 2008.
- Seminario de Investigación del Postgrado en Ingeniería de Sistemas, Universidad Autónoma de Nuevo León, México, abril 2008 y septiembre 2008.

Firma del asesor: \_\_\_\_\_

Dr. Igor S. Litvinchev

## CAPÍTULO 1

# INTRODUCCIÓN

---

### 1.1 ESTRUCTURA DE LA TESIS

Este trabajo de tesis se encuentra estructurado de la siguiente forma:

- En el primer capítulo se presenta el objetivo del trabajo de tesis, así como las hipótesis, justificación y la metodología empleada durante el mismo.
- El segundo capítulo está dedicado a la revisión de la literatura existente principalmente sobre la relajación lagrangiana y se describen algunas metodologías para resolver el problema dual lagrangiano.
- Las construcciones básicas se encuentran en el capítulo tres, en este se describe la técnica empleada así como el algoritmo utilizado durante este trabajo. También se presentan algunas aplicaciones a los problemas de mochila multidimensional y asignación generalizada, se formula cada problema y sus posibles relajaciones.
- El cuarto capítulo incluye los experimentos computacionales, sus generalidades y algunos resultados importantes así como un análisis estadístico de los mismos.
- Por último se presentan las conclusiones obtenidas de esta investigación y algunas sugerencias para trabajo futuro.



## 1.2 ESTRUCTURAS ESPECIALES

La solución de problemas que envuelven sistemas reales se ve a menudo limitada por el tamaño, es decir, el problema es simplemente muy grande. En la programación matemática, el tamaño del problema es determinado por el número de variables, el número y complejidad de las restricciones y la complejidad de la función objetivo. El significado de “grande” depende de la capacidad de los algoritmos de solución, la velocidad y capacidad del equipo de cómputo disponible, etc.

Afortunadamente, este tipo de problemas generalmente cuentan con una estructura especial. Para programas lineales, el estudio de la estructura del sistema se refleja en el patrón de elementos cero y distintos de cero de la matriz de coeficientes. Cuando el programa es grande, la densidad de la matriz de elementos distintos de cero es inferior a un pequeño porcentaje. Por otra parte, estas por lo general están organizadas de manera ordenada, por ejemplo, en bloques a lo largo de la diagonal principal con excepción de unas pocas filas o columnas. De manera alternativa, algunos programas lineales tienen muchas filas o columnas (pero no ambas) y el conjunto de columnas (filas) está definido por alguna relación explícita.

En el caso no lineal, los problemas son más difíciles de clasificar. Se puede definir una matriz cuyo elemento  $(i, j)$  es unitario si la variable  $i$  aparece en la restricción  $j$  y cero de otro modo, y esta matriz refleja la estructura del problema.

Los enfoques para resolver problemas grandes pueden ser divididos en dos clases: Métodos directos, y técnicas de descomposición o separación. Los métodos directos especializan un algoritmo existente a una clase particular de problemas. En el caso de matrices con estructuras especiales, las operaciones necesarias pueden ser realizadas a menudo guardando una matriz de tamaño reducido.

Los métodos indirectos se caracterizan por la descomposición del sistema original en subsistemas, cada uno con subproblemas independientes más pequeños. Dado que los subsistemas interactúan entre sí, resolver los subproblemas no provee, en

general, la solución correcta.

Todos los algoritmos de descomposición existentes tienen la estructura de dos niveles. Las ventajas de estos incluyen la libertad para resolver cada subproblema usando cualquier algoritmo deseado, la capacidad de escribir programas más pequeños e independientes y la reducción de memoria computacional requerida. El principal inconveniente es que los subproblemas deben ser resueltos una serie de veces, entonces, puede que se use mucho tiempo computacional en ello.

### 1.3 MOTIVACIÓN

Existen diversas maneras en las que un problema puede ser relajado de forma lagrangiana. Por lo tanto podemos tener varias cotas lagrangianas para el mismo problema, entonces aparece el problema de cómo escoger la mejor cota lagrangiana.

### 1.4 OBJETIVOS

Pretendemos formular un solo problema para encontrar el valor de la mejor cota lagrangiana sin calcular todas las cotas existentes y proponer el método numérico correspondiente.

### 1.5 HIPÓTESIS

Suponemos que el problema puede ser formalizado como el problema de búsqueda de la mejor cota dual lagrangiana.

## 1.6 METODOLOGÍA

La metodología que se siguió a lo largo de este trabajo consta primero de una parte teórica en la cual se formuló paso a paso el problema de encontrar la mejor cota lagrangiana sin calcular explícitamente todas las posibles cotas. Se propone un algoritmo para encontrar el índice de la mejor cota y se utiliza el método de descomposición de Benders para resolverlo.

Consta también de la implementación de dicho algoritmo a dos problemas distintos que se describen formalmente mas adelante y al final se analizan los resultados para obtener las conclusiones necesarias.

## CAPÍTULO 2

# ANTECEDENTES

---

A lo largo de este capítulo se hará una revisión acerca de la literatura existente sobre la relajación lagrangiana, como solucionar el problema relajado, encontrar soluciones factibles, características de la función lagrangiana, entre otros.

## 2.1 RELAJACIÓN LAGRANGIANA

Para comenzar esta sección se reafirmarán propiedades esenciales de la relajación y función lagrangiana y describiremos varios algoritmos para resolver el problema dual lagrangiano considerando heurísticas lagrangianas, *ad hoc* o genéricas, debido a la importancia de éstas en cualquier esquema de aproximación lagrangiana. Análogamente se discuten algunos esquemas que pueden mejorar potencialmente la cota de la relajación lagrangiana y se describen varias aplicaciones de ésta, las cuales demuestran la flexibilidad de la aproximación y permite el cálculo de cualquier cota fuerte de un valor óptimo de problemas *MIP* (por sus siglas en inglés de: problema entero mixto), o el uso de heurísticas lagrangianas, las cuales probablemente son seguidas por una heurística iterativa de mejoramiento. Por último analizaremos el porqué algunas veces podemos conseguir una cota fuerte solucionando problemas sencillos y porqué una relajación a priori débil, algunas veces es mejor o igual que una a priori fuerte.

Empezaremos con una descripción de las relajaciones, en particular la relajación lagrangiana (*LR*), continuando con la interpretación geométrica de *LR*. Tam-

bién daremos un repaso a las extensiones de  $LR$ : descomposición lagrangiana y de forma más general la sustitución.

Enfocaremos una sección a los métodos de solución para el problema dual, empezando con optimización subgradiente, y prosiguiendo con métodos basados en la propiedad lagrangiana: planos cortantes (o generación de restricciones), Dantzig-Wolfe (o generación de columnas), algoritmo volumen, métodos lagrangianos de agregación y desagregación, también algunos métodos de aproximación híbrida. Continuaremos dando un breve repaso de varias características de la función lagrangiana, las cuales son importantes para el diseño de métodos eficientes de optimización.

Además dedicaremos una sección a las heurísticas lagrangianas, las cuales son un complemento de acotación lagrangiana tratando de hacer una transformación de soluciones lagrangianas infactibles en soluciones factibles.

La literatura de la relajación lagrangiana, sus extensiones y aplicaciones es enorme. Por lo tanto no es posible citar a todos los artículos que se refieren a relajación lagrangiana. Por lo que si se desea ver una visión más amplia acerca del gran campo sobre este tema el lector podrá ver: [6, 27, 28, 29, 37, 49, 50, 59, 80, 81].

Si  $P$  es un problema de optimización, se usa la siguiente notación:

- $FS(P)$ : el conjunto de soluciones factibles del problema  $P$
- $OS(P)$ : el conjunto de soluciones óptimas del problema  $P$
- $v(P)$ : el valor óptimo del problema  $P$
- $u^k, s^k, \text{etc.}$ : el valor de  $u$  y  $s$ , etc., usado en la iteración  $k$
- $x^T$ : transpuesta de  $x$
- $x^k$ : el  $k$ -ésimo punto extremo de algún poliedro
- $x^{(k)}$ : una solución encontrada en la iteración  $k$
- $\text{conv}(X)$ : envolvente convexo del conjunto  $X$ .

### 2.1.1 RELAJACIONES DE PROBLEMAS DE OPTIMIZACIÓN

En 1974 Geoffrion [37] define formalmente la relajación de un problema de optimización como sigue:

**Definición 2.1** *El problema  $(RP_{\min}) : \min\{g(x) \mid x \in W\}$  es una **relajación** del problema  $(P_{\min}) : \text{Hallar } \min\{f(x) \mid x \in V\}$ , con la misma variable de decisión  $x$ , si y sólo si:*

- i) *el conjunto factible de  $(RP_{\min})$  contiene a  $(P_{\min})$ , esto es  $W \supseteq V$  y*
- ii) *sobre el conjunto factible de  $(P_{\min})$ , la función objetivo de  $(RP_{\min})$  domina (es mejor que) a  $(P_{\min})$ , esto es,  $\forall x \in V, g(x) \leq f(x)$ .*

Se ve claramente que  $v(RP_{\min}) \leq v(P_{\min})$ , en otras palabras  $(RP_{\min})$  es una versión *optimista* de  $(P_{\min})$ : tiene más soluciones factibles que  $(P_{\min})$ , y para las soluciones factibles de  $(P_{\min})$ , la función objetivo es mejor que (más pequeña que)  $(P_{\min})$ ; así que tiene valor mínimo.

Por supuesto, si el problema original es un problema de maximización, decimos que  $(P_{\max}) : \max\{f(x) \mid x \in V\}$ , una relajación de  $(P_{\max})$  es un problema  $(RP_{\max})$  con la misma variable de decisión  $x$  de la forma  $(RP_{\max}) : \max\{g(x) \mid x \in W\}$ , tal que:

- i) *el conjunto factible de  $(RP_{\max})$  contiene a  $(P_{\max})$ , esto es  $W \supseteq V$  y*
- ii) *sobre el conjunto factible de  $(P_{\max})$ , la función objetivo de  $(RP_{\max})$  domina (es mejor que) a  $(P_{\max})$ , esto es,  $\forall x \in V, g(x) \geq f(x)$ .*

A esto le sigue que  $v(RP_{\max}) \geq v(P_{\max})$ , y similarmente al caso de minimización,  $(RP_{\max})$  es una versión *optimista* de  $(P_{\max})$ . Por lo tanto podemos considerar

de forma indiferente un problema de maximización o minimización. Debido a que puede ser transformado fácilmente de un formato a otro, recordando que:

$$\text{máx}\{f(x) \mid x \in V\} = \text{mín}\{-f(x) \mid x \in V\} \quad (2.1)$$

La relajación tiene un rol ambiguo, pues da cotas y soluciones (factibles) del valor óptimo de un problema difícil de optimización y usa las soluciones de un problema infactible original como puntos de inicio (guía) para heurísticas especializadas.

También nos concentraremos en *problemas lineales de programación entera*, los cuales poseen un conjunto de restricciones  $V$  el cual está definido por un **poliedro de restricciones racionales**, agregando las condiciones de integralidad, esté último es un subconjunto de los componentes de  $x$ , esto es,  $V = \Pi \cap \Gamma$ , donde  $\Pi$  es un poliedro racional ( $\Pi$  también puede contener restricciones de signo para  $x$ ) y  $\Gamma$  es  $\mathbb{R}^{n-p} \times \mathbb{Z}^{p-q} \times \{0, 1\}^q$ ,  $n \geq p \geq 1$ ,  $p \geq q \geq 0$ , con  $p$  y  $q$  enteros. Podemos llamar “problema de programación entera” a cualquier problema, i.e., no haremos distinción entre problemas generales puros (i.e., con  $p = n$ ) y mixtos (i.e., con  $1 \leq p < n$ ) enteros. En el caso especial de programación 0 – 1 usaremos  $\Gamma = \mathbb{R}^{n-p} \times \{0, 1\}^q$ ,  $q \geq 1$ .

La relajación ampliamente usada para problemas de programación como  $P : \text{mín( ó máx)}\{f(x) \mid x \in V\}$  es la relajación continua ( $CR$ ) (conocida comunmente como relajación lineal) i.e., el problema  $P$  con las restricciones de integralidad sobre  $x$  ignoradas.

### 2.1.2 RELAJACIÓN LAGRANGIANA ( $LR$ )

Ahora introducimos la definición de  $LR$  dada por Held y Karp,[49, 50], tomando en cuenta la definición de relajación de Geoffrion [37]. Sin perdida de generalidad, suponemos que  $P$  es de la forma:

$$P : \text{Hallar mín } \{fx \mid Ax \leq b, Cx \leq d, x \in X\} \quad (2.2)$$

donde  $X$  contiene restricciones de signo y de integralidad para  $x$ , esto es  $X = \mathbb{R}^{n-p} \times \mathbb{Z}^p$ ,  $X = \mathbb{R}_+^{n-p} \times \mathbb{R}_+^p$ ,  $X = \mathbb{R}_+^{n-p} \times \{0, 1\}^p$  ó  $X = \mathbb{R}^{n-p} \times \mathbb{Z}^{p-q} \times \{0, 1\}^q$ . Sea  $I(X)$  el conjunto de  $p$  índices de  $x$  que están restringidos a ser enteros (o binarios). Se supone que las restricciones de la forma  $Ax \leq b$  son *difíciles*, en el sentido de que el problema  $P$  sin estas sería mucho más simple de resolver. Las restricciones tipo  $Cx \leq d$  (posiblemente vacío) podrían ser guardadas, junto con  $X$ , para formar la *relajación lagrangiana* como sigue:

Sea  $\lambda$  un vector de pesos no negativo, llamado *multiplicadores de Lagrange*.

**Definición 2.2** La *relajación lagrangiana* de  $P$  relacionada a las restricciones difíciles  $Ax \leq b$ , con multiplicadores lagrangianos  $\lambda$  no negativos, es el problema:

$$(LR_\lambda) : \text{Hallar } \min \{fx + \lambda(Ax - b) \mid Cx \leq d, x \in X\} \quad (2.3)$$

En  $(LR_\lambda)$ , la holgura de las restricciones difíciles  $Ax \leq b$  han sido agregadas a la función objetivo con pesos  $\lambda$  y las restricciones  $Ax \leq b$  eliminadas. Decimos que las restricciones  $Ax \leq b$ , han sido *dualizadas*.  $(LR_\lambda)$  es una relajación de  $P$  entonces:

- i.  $FS(LR_\lambda)$  contiene  $FS(P)$ ,
- ii. para cualquier  $x$  de  $P$ , y
- iii. cualquier  $\lambda \geq 0$ ,  $fx + \lambda(Ax - b)$  es menor o igual que  $fx$  (i.e. *mejor*, cuando estemos minimizando). Ahora podemos decir que  $v(LR_\lambda) \leq v(P)$ ,  $\forall \lambda \geq 0$ , esto es, que el valor óptimo  $v(LR_\lambda)$ , el cuál depende de  $\lambda$ , es una cota mínima del valor óptimo de  $P$ .

**Definición 2.3** El problema de encontrar la mejor cota lagrangiana de  $v(P)$  es *hallar*:



$$\max_{\lambda \geq 0} v(LR_\lambda) \quad (2.4)$$

este es llamado **dual lagrangiano** de  $P$  relacionado a las restricciones difíciles.

$LR$  es un problema en el espacio *dual* de los multiplicadores lagrangianos, mientras que  $(LR_\lambda)$  es sobre  $x$ .

A partir de ahora, cuando hablemos acerca de la *cota por relajación lagrangiana*, o simplemente *la cota lagrangiana*, siempre podremos referirnos a  $v(LR)$ , y no a  $v(LR_\lambda)$  para cualquier  $\lambda$  arbitrario.

**Nota.** *Supongamos que el problema bajo consideración tiene restricciones de igualdad en lugar de restricciones de desigualdad. Podemos referirnos a tal problema como  $(Q)$  a continuación:*

$$(Q) : \text{Hallar } \min_x \{fx \mid Ax = b, Cx \leq d, x \in X\} \quad (2.5)$$

Podemos dualizar las restricciones  $Ax = b$  siendo remplazadas por un par de restricciones de desigualdad:  $Ax \leq b$  y  $-Ax \leq -b$ . Entonces sea  $\mu \geq 0$  y  $\nu \geq 0$  los multiplicadores lagrangianos de dimensiones apropiadas. La relajación lagrangiana para  $\mu \geq 0$  y  $\nu \geq 0$  dados es:

$$(LR_{\mu,\nu}) : \text{Hallar } \min_x \{fx + \mu(Ax - b) + \nu(-Ax + b) \mid Cx \leq d, x \in X\} \quad (2.6)$$

el cual es equivalente a:

$$(LR_\lambda) : \text{Hallar } \min_x \{fx + \lambda(Ax - b) \mid Cx \leq d, x \in X\} \quad (2.7)$$

con  $\lambda = \mu - \nu$ . Nótese que en el caso de igualdad,  $\lambda$  no necesariamente debe de tener valores no negativos para que  $(LR_\lambda)$  sea una relajación de  $(Q)$ .

### 2.1.3 SOLUCIÓN FACTIBLE LAGRANGIANA

Sea  $x(\lambda)$  la solución óptima de  $(LR_\lambda)$  para algún  $\lambda \geq 0$ , entonces  $x(\lambda)$  es llamada una *solución lagrangiana*. Podríamos pensar que la solución lagrangiana  $x(\lambda)$  es factible (hecho que por lo general no sucede) para un problema entero (esto es, que satisface las restricciones dualizadas).

Lo que sabemos acerca del valor óptimo  $v(P)$  de  $P$ , es que se encuentra entre  $fx(\lambda) + \lambda[Ax(\lambda) - b]$  y  $fx(\lambda)$ , y en tal caso  $fx(\lambda)$  es el valor de una solución factible de  $P$ , el cual es una cota superior de  $v(P)$  y  $fx(\lambda) + \lambda[Ax(\lambda) - b]$  es el valor óptimo del problema lagrangiano  $(LR_\lambda)$ , y es un cota inferior de  $v(P)$ . Sin embargo, si cumple las condiciones de *holgura complementaria*, esto es, si,  $\lambda[Ax(\lambda) - b] = 0$ , entonces  $fx(\lambda) + \lambda[Ax(\lambda) - b] = v(P) = fx(\lambda)$  y  $x(\lambda)$  es una solución óptima para  $P$ .

**Teorema 2.4** 1. Si  $x(\lambda)$  es una solución óptima de  $(LR_\lambda)$  para algún  $\lambda \geq 0$ ,

entonces  $fx(\lambda) + \lambda[Ax(\lambda) - b] \leq v(P)$ .

2. Además si  $x(\lambda)$  es una solución factible para el problema  $P$ , entonces  $fx(\lambda) + \lambda[Ax(\lambda) - b] \leq v(P) \leq fx(\lambda)$ .

3. Y si además  $\lambda[Ax(\lambda) - b] = 0$ , entonces  $x(\lambda)$  es la solución óptima de  $P$  y  $v(P) = fx(\lambda)$ .

**Nota.** Primero haremos notar que la condición 3 del teorema 2.4 es una condición suficiente de optimalidad, pero no necesaria. Esto es, que es posible para una  $x(\lambda)$  factible ser óptima para  $P$ , aún cuando esta no satisfaga las condiciones de *holgura complementaria*.

Si las restricciones son dualizadas como restricciones de *igualdad* y si  $x(\lambda)$  es una solución factible para  $(Q)$ , las condiciones de *holgura complementaria* se cumplen automáticamente, por lo tanto  $x(\lambda)$  es una solución óptima de  $(Q)$ , con  $v(P) = fx(\lambda)$ .

### 2.1.4 INTERPRETACIÓN GEOMÉTRICA

El siguiente teorema, es de donde probablemente Geoffrion [37] desprende la idea de la relajación lagrangiana. El cual da una interpretación geométrica del problema dual lagrangiano en el espacio de  $x$ , esto es, en el espacio primal (el espacio dual es donde están los multiplicadores lagrangianos  $\lambda$ ) y esto nos permite estudiar los esquemas de la relajación lagrangiana.

**Teorema 2.5** *El dual lagrangiano  $LR$  es equivalente a la relajación primal*

$$(PR) : \text{Hallar } \min_x \{fx \mid Ax \leq b, x \in \text{conv}\{x \in X \mid Cx \leq d\}\} \quad (2.8)$$

en el sentido que  $v(LR) = v(PR)$ .

Este resultado está basado en la dualidad del problema lineal  $LP$  y las propiedades de las soluciones óptimas de programas lineales. Este resultado podría no ser verdadero si la matriz de restricciones es no racional, o mejor dicho para un poliedro no racional que no es igual al envolvente convexo de sus puntos extremos. Aunque en la práctica los números en las computadoras son almacenados como números racionales y por lo tanto todas las matrices son racionales, pero ocasionalmente esto modifica la verdadera estructura del poliedro asociado.

La siguiente definición es importante y los resultados son obtenidos de la interpretación geométrica.

**Definición 2.6** *Decimos que el problema dual lagrangiano  $LR$  tiene la **propiedad de integralidad** si  $\text{conv}\{x \in X \mid Cx \leq d\} = \{x \mid Cx \leq d\}$ .*

Si  $LR$  tiene la propiedad de integralidad (la cual de aquí en adelante será representada por  $IP$ ), entonces los puntos extremos de  $\{x \mid Cx \leq d\}$  están en  $X$ . Desafortunadamente la consecuencia es tal que un esquema  $LR$  no puede producir

una cota mejor a la de  $LP$ , esto es lo que se indica en los siguientes corolarios. Sin embargo, algunas veces, esto se puede usar de todas formas porque la relajación  $LP$  no puede ser calculada fácilmente. Esto puede ocurrir para algunos ejemplos de algún problema que tenga un número exponencial de restricciones, las cuales pueden ser relajadas de cualquier forma para resolver fácilmente los subproblemas. Un ejemplo de los problemas que tiene un número exponencial de restricciones es el del agente viajero.

Recordemos que cualquier cota de relajación lagrangiana siempre es menor que la mejor cota obtenida por  $LP$ , nunca peor.

**Corolario.** *Si el conjunto  $\text{conv}\{x \in X \mid Cx \leq d\} = \{x \mid Cx \leq d\}$ , entonces  $v(LP) = v(PR) = v(LR) \leq v(P)$ . En este caso, la cota de relajación lagrangiana es igual (no puede ser mejor que) a la cota de  $LP$ .*

**Corolario.** *Si el conjunto  $\text{conv}\{x \in X \mid Cx \leq d\} \subset \{x \mid Cx \leq d\}$ , entonces  $v(LP) \leq v(PR) = v(LR) \leq v(R)$  y puede ocurrir que la cota de relajación lagrangiana sea estrictamente mejor que la cota  $LP$ .*

Lo que dicen estos dos corolarios es que a menos que  $LR$  no tenga la propiedad de integralidad, podría mejorar la cota obtenida por la relajación  $LP$ . Es importante saber esto si todos los vértices del poliedro racional  $\{x \mid Cx \leq d\}$  están en  $X$ . El siguiente análisis puede demostrar la importancia de este concepto.

**Ejemplo.** *El problema de asignación generalizada GAP consiste en asignar un conjunto de trabajos ( $j \in J$ ) a uno de máquinas ( $i \in I$ ) con el menor costo total de asignación posible (o posiblemente con el mayor valor de ganancia). El costo (o ganancia) de asignar  $j$  a  $i$  es  $c_{ij}$ , entonces el problema puede ser un problema de minimización o maximización, usando la notación “mín (ó máx)”. Cada trabajo debe ser realizado por una máquina (así que las restricciones de múltiples elecciones (MC)). Cada máquina  $i$  es válida por  $b_i$  unidades de tiempo y asignando el trabajo  $j$  a la máquina  $i$  usada por  $a_{ij}$  unidades de tiempo (como las restricciones de la mochila (KP)). Entonces el modelo es:*

$$\begin{aligned}
& \text{mín (ó máx)} \quad \sum_i^m \sum_j^n c_{ij} x_{ij} && (GAP) \\
& \text{s.a:} \quad \sum_{j=1}^n a_{ij} x_{ij} \leq b_i \quad i = 1, \dots, m && (KP) \\
& \quad \sum_{i=1}^m x_{ij} = 1 \quad j = 1, \dots, n && (MC) \\
& \quad x_{ij} \in \{0,1\} \quad \forall i \in I, j \in J
\end{aligned} \tag{2.9}$$

- Si dualizamos la restricción tipo  $(MC)$  con multiplicadores sin signo  $\lambda_j$ , el problema de relajación lagrangina se descompone en subproblemas para cada máquina  $i$ :

$$\begin{aligned}
& \text{mín (ó máx)} \quad \sum_i^m \sum_j^n c_{ij} x_{ij} + \sum_j^n \lambda_j (1 - \sum_i^m x_{ij}) && (LR_\lambda) \\
& \text{s.a:} \quad \sum_{j=1}^n a_{ij} x_{ij} \leq b_i \quad i = 1, \dots, m && (KP) \\
& \quad x_{ij} \in \{0,1\} \quad \forall i \in I, j \in J
\end{aligned} \tag{2.10}$$

Podemos escribir este problema como:

$$\text{mín( ó máx)} \left\{ \sum_i^m \sum_j^n (c_{ij} - \lambda_j) x_{ij} + \sum_j^n \lambda_j \mid \sum_{j=1}^n a_{ij} x_{ij} \leq b_i, x_{ij} \in \{0, 1\}, \forall i, j \right\} \tag{2.11}$$

Que se puede representar también de la siguiente forma:

$$\sum_j^n \lambda_j + \sum_i^m \text{mín (ó máx)} \left\{ \sum_j^n (c_{ij} - \lambda_j) x_{ij} \mid \sum_{j=1}^n a_{ij} x_{ij} \leq b_i, x_{ij} \in \{0, 1\}, \forall j \right\} \tag{2.12}$$

Así que el  $i$ -ésimo subproblema lagrangiano es un problema de la mochila para la  $i$ -ésima máquina. Este problema no tiene la propiedad de integralidad entonces la relajación  $LP$  de un problema de mochila 0 – 1 no siempre tiene una solución entera óptima. Entonces este esquema  $LR$  puede (y usualmente lo hace) dar un cota más fuerte que la cota de  $LP$  y en particular esto fue usado en [31, 46].

- Si dualizamos la restricción tipo  $(KP)$ , el problema de relajación lagrangiana se descompone en subproblemas por trabajos  $j$  (con  $\lambda$  no positivos o no negativos, dependiendo si el problema es de mín ó máx):

$$\text{mín( ó máx)} \sum_i^m \sum_j^n c_{ij} x_{ij} + \sum_i^m \lambda_i (b_i - \sum_j^n a_{ij} x_{ij}) \quad (LR'_\lambda)$$

$$\text{s.a: } \sum_{i=1}^m x_{ij} = 1 \quad \forall j \in J \quad (MC) \quad (2.13)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in I, j \in J$$

Es decir,

$$\begin{aligned} \text{mín}_x \text{ (ó máx}_x \text{)} \left\{ \sum_i^m \sum_j^n (c_{ij} - \lambda_i a_{ij}) x_{ij} + \sum_i^m \lambda_i b_i \right\} \\ \text{s. a: } \sum_i^m x_{ij} = 1, \forall j \in J, i \in I, x_{ij} \in \{0, 1\} \end{aligned} \quad (2.14)$$

De manera equivalente:

$$\sum_i^m \lambda_i b_i + \sum_j^n \left\{ \text{mín}_x \text{ (ó máx}_x \text{)} \sum_i^m (c_{ij} - \lambda_i a_{ij}) x_{ij} : \sum_i^m x_{ij} = 1, \forall i, x_{ij} \in \{0, 1\} \right\} \quad (2.15)$$

El  $j$ -ésimo subproblema lagrangiano es un problema de elección múltiple para el  $j$ -ésimo trabajo. La relajación  $LP$  para cada problema siempre encuentra una solución entera óptima (escoge la mejor asignación para cada  $j$ ), así que los subproblemas lagrangianos tienen la Propiedad de Integralidad y la cota de  $LR$  es igual a la cota de  $LP$ . Se espera que la cota de  $LP$  no pueda ser mejorada. Sin embargo cabe mencionar que es peor resolver la relajación  $LP$  del  $GAP$  que la relajación lagrangiana ya que  $LR$  debido a las ventajas que esta presenta al solucionarla. Primero, el dual lagrangiano puede ser más fácil de resolver que el dual  $LP$  para problemas de tamaño grande. Después, agregando la cota de  $LP$ ,  $LR$  encuentra soluciones lagrangianas,

las cuales son factibles para múltiples restricciones elegidas pero pueden violar una o más restricciones del tipo  $(KP)$ . La solución lagrangiana obtenida puede ser usada como puntos para empezar en una heurística lagrangiana. Esta relajación es definida en [75].

### 2.1.5 CONSTRUCCIÓN DE UNA RELAJACIÓN LAGRANGIANA

Existen varias formas en las cuales puede relajarse un problema dado una forma lagrangiana. Aquí enlistaremos pocas, la mayoría de algunas relajaciones pueden ayudar a una reformulación a priori de la relajación, y para algunos modelos complejos, la intuición e interpretación de las interacciones entre restricciones puede sugerir esquemas de relajación eficientes.

1. **Se puede aislar un subproblema interesante y dualizar otras restricciones.**

Esta es la aproximación comúnmente más usada. Tiene la ventaja de que los subproblemas lagrangianos “interesantes” (en el sentido de que usualmente tienen una estructura especial que puede ser explotada) pueden ser resueltos por algoritmos existentes especializados para resolverlos de forma eficiente.

2. **Si hay dos (o más) subproblemas interesantes con variables en común, primero podemos partir estas variables, después dualizar las restricciones copiadas.**

Este proceso es llamado *descomposición lagrangiana (LD)* [83], *partición de variables* [67] o *layering variables*, [39, 73, 82] son de los primeros que introdujeron artículos acerca de esta aproximación. Primero se debe reformular el problema usando partición de variables, en otras palabras, debemos renombrar las variables en parte de las restricciones como si fueran variables independientes. El problema  $P : \min_x \{fx \mid Ax \leq b, Cx \leq d, x \in X\}$  es claramente equivalente al problema

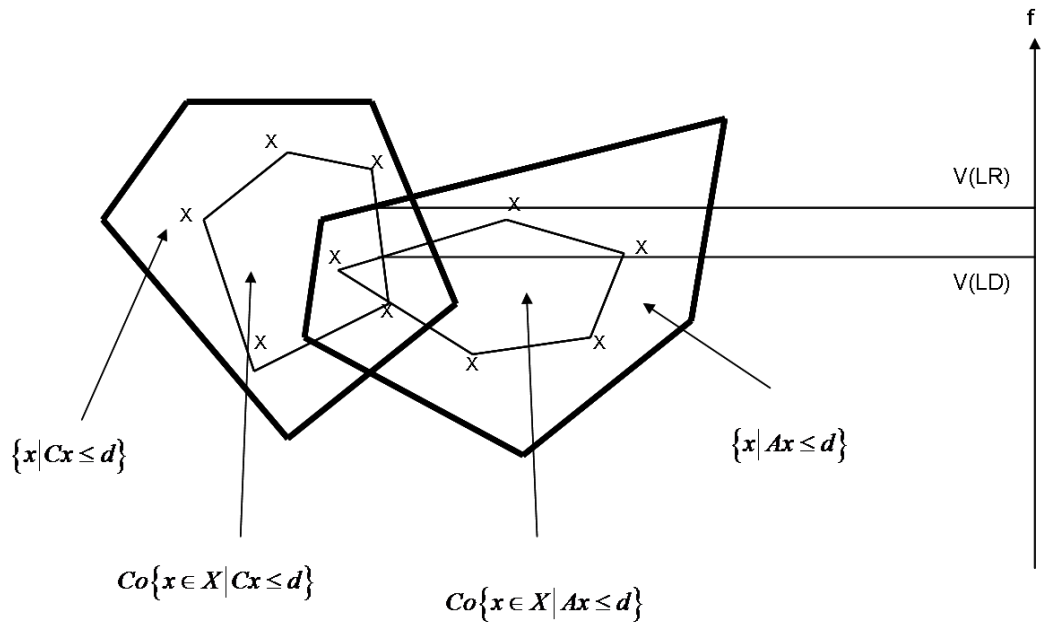


Figura 2.1: Interpretación geométrica de la descomposición[45]

$(P')$  :  $\min_{x,y} \{fx \mid Ax \leq b, x \in X, Cy \leq d, y \in Y, x = y\}$ , en el sentido que tiene valores óptimos iguales (pero ellos tienen diferentes espacios para las variables).

Además si  $x^*$  es la solución óptima de  $P$ , entonces la solución  $(x, y) \equiv (x^*, x^*)$  es un óptimo para  $(P')$  y si  $(x^*, y^*)$  es una solución óptima de  $(P')$ , entonces  $x^* = y^*$  y  $x^*$  es un óptimo para  $P$ . Dualizando la copia de las restricciones  $x = y$  en  $(P')$  con multiplicadores  $\lambda$ , lo cual separa al problema en uno para  $x$  y otro para  $y$ :

$$(LD_\lambda): \text{Hallar } \min_{x,y} \{fx + \lambda(y - x) \mid Ax \leq b, Cy \leq d, x \in X, y \in X\}$$

$$\min_x \{(f - \lambda)x \mid Ax \leq b, x \in X\} + \min_y \{\lambda y \mid Cy \leq d, y \in X\}$$

(2.16)

Este proceso crea un caso de estructura de escalera y por lo tanto el modelo puede ser descompuesto. Nótese que  $\lambda$  no requiere ser no negativo. Recordemos también que cuando dualicemos una restricción de igualdad, automáticamente una solución factible lagrangiana es óptima para el problema de programación entero original.



Las restricciones copiadas son restricciones de igualdad, si ambos subproblemas lagrangianos tienen la misma solución óptima, la solución también es óptima para el problema  $IP$ .

En [41] se muestra que la cota  $LD$  puede dominar estrictamente las cotas  $LR$  obtenidas de la dualización de un conjunto de restricciones:

**Teorema 2.7** *Si se cumple que*

$$v(LD) = \max_{\lambda} \left\{ \min_x \{(f - \lambda)x \mid Ax \leq b, x \in X\} \right\} + \min_y \{\lambda y \mid Cy \leq d, y \in X\} \quad (2.17)$$

entonces

$$v(LD) = \min \{fx \mid x \in \text{conv}\{x \in X \mid Ax \leq b\} \cap \text{conv}\{x \in X \mid Cx \leq d\}\}. \quad (2.18)$$

Esta interpretación geométrica es mostrada en la Figura 2.1.

**Corolario.**

- Si uno de los subproblemas tiene la propiedad de integralidad, entonces  $v(LD)$  es igual que el mejor que las dos cotas  $LR$  correspondientes a dualizar  $Ax \leq b$  o  $Cx \leq d$ .
- Si ambos subproblemas tienen la propiedad de integralidad, entonces  $v(LD) = v(LP)$ .

Si uno aplica  $LD$  a  $GAP$  haciendo una partición de restricciones en dos subconjuntos no intersecados, es decir en las restricciones tipo  $(KP)$  y  $(MC)$ , entonces se obtiene la misma cota como cuando dualizamos las restricciones de múltiple elección.

Entonces puede parecer poco interesante partir las variables, debido a que esto requiere un número de multiplicadores igual al número de tiempo máquina y de

trabajos, comparando sólo el número de trabajos con la relajación fuerte lagrangiana. Es posible pensar que las soluciones lagrangianas pueden explotar las dos soluciones lagrangianas obtenidas, además de que puede ser peor el trabajo extra de resolver la descomposición dual lagrangiana, [53].

Ocasionalmente la partición de las variables puede corresponder a la partición física de una variable de decisión del problema.

### 3. Podemos dualizar restricciones de acoplamiento.

Algunas veces, después de alguna reformulación, frecuentemente los problemas pueden contener estructuras independientes ligadas por varias restricciones:  $\min_{x,y} \{fx + gy \mid Ax \leq b, x \in X, Cx \leq d, y \in Y, Ex + Fy \leq h\}$ . Dualizando las restricciones de enlace  $Ex + Fy \leq h$  dividen el problema en uno sobre  $x$  y otro sobre  $y$ . Algunas veces el problema original sólo contiene  $x$  y varias reformulaciones introducen una nueva variable  $y$ , mientras la relación entre  $x$  y  $y$  es capturada por una nueva restricción  $Ex + Fy \leq h$ .

**Ejemplo.** *Un problema de producción múltiple sobre fábricas contiene restricciones relacionadas a fábricas individuales y todas las restricciones de demanda están relacionadas a la producción de las plantas. Si se dualizan las restricciones de demanda, el problema lagrangiano se descompone en un problema de producción para cada fábrica, el cuál normalmente es mucho más fácil de resolver que los demás problemas. Si el último de estos subproblemas no tiene la propiedad de integralidad, este LR puede reducir el intervalo de diferencia entre la cota inferior y superior las cuales son menores que la cota LP.*

**Ejemplo.** *Considere un modelo multi-período en el cual las fábricas construidas en un período pueden ser usadas en ese o un período después. Podemos usar las variables (se dice que la variable binaria  $x_{it}$ , es igual a uno si la fábrica  $i$  es construida en el período  $t$ ) de acción (construcción) en parte del “diseño” del modelo y variables (se dice que la variable binaria  $y_{it}$ , es 1 si la fábrica  $i$  existe en el período  $t$ )*

de “estado” (existencia) en el resto del modelo. Así que  $y_{it} \geq y_{i,t-1} \forall i, t$ . La relación entre los dos conjuntos de variables es almacenada por las siguientes restricciones:

$$x_{it} \geq y_{it} - y_{i,t-1} \quad y \quad y_{it} \geq x_{i\tau}, \quad y \quad \forall i, t, \quad \tau \leq t. \quad (2.19)$$

Ambos tipos de restricciones son necesarias para que se cumpla que  $x_{it}$  es 1 sólo en el período de construcción, esto es, cuando  $y_{it}$  es 1 y  $y_{i,t-1}$  es 0 y  $y_{i\tau}$  permanece en 0 hasta el período más pequeño  $\tau = t$  para el cual  $x_{it}$  es 1.

Dualizando éstas relaciones de acoplamiento entre “construcción en el período  $t$ ” y “construido por el período  $t$ ” el modelo se partirá en un problema de construcción de fábricas y el problema del uso de fábricas. Si también tiene la propiedad de integralidad, la cota de relajación lagrangiana puede ser más fuerte que la cota de LP. Ver casos prueba en [9].

Actualmente este es un caso especial de *substitución lagrangiana (LS)*, donde  $Ex + Fy \leq h$  es la copia de la restricción introducida en la reformulación.

#### 4. Algunas veces es mejor dualizar un conjunto de copias individuales de variables.

En lugar de crear una copia  $y$  de la variable  $x$  e introducir  $y$  en el modelo  $P$  para reescribir la restricción  $Cx \leq d$  como  $Cy \leq d$ , dando el modelo equivalente  $(P')$  :  $\min_{x,y} \{fx \mid Ax \leq b, x \in X, Cy \leq d, y \in X, x = y\}$ , también se puede crear el problema  $(P'')$  equivalente al problema  $P$  introduciendo una nueva variable  $y$  y forzando la restricción  $Dy = Cx$ . Esta restricción en general es más débil que la restricción  $x = y$ . El modelo  $(P'')$  es  $\min_{x,y} \{fx \mid Ax \leq b, x \in X, Dy \leq d, y \in X, Dx = Cy\}$ . Aquí el LR introducido dualiza una copia de la restricción agregada  $Dx = Cy$ .

Nuevamente aquí la restricción copiada es una restricción de igualdad, por lo tanto si el problema lagrangiano tiene soluciones óptimas  $x$  y  $y$  que satisfacen la copia de la restricción agregada, esto es, si  $Dx = Cy$ , entonces la solución  $x$  es óptima para el problema IP.

Considere el problema bi-mochila

$$(BKP) : \text{Hallar } \max_x \left\{ \sum_i c_i x_i \mid \sum_i b_i x_i \leq m, \sum_i d_i x_i \leq n, x_i \in \{0, 1\}, \forall i \right\} \quad (2.20)$$

Se puede introducir una nueva variable  $y$ , se escribe  $\sum_i d_i x_i = \sum_i d_i y_i$ . El problema equivalente es:

$$(BK'P) : \text{Hallar } \max_x \left\{ \sum_i c_i x_i \mid \sum_i b_i x_i \leq m, \sum_i d_i x_i \leq n, \sum_i b_i x_i = \sum_i b_i y_i, y_i \in \{0, 1\}, \forall i \right\} \quad (2.21)$$

y el problema  $LR$  es:

$$\begin{aligned} (LR_\lambda) : \quad & \max_{x,y} \sum_i c_i x_i - \lambda \left( \sum_i b_i x_i - \sum_i b_i y_i \right) \\ \text{s.a:} \quad & \sum_i b_i x_i \leq m \\ & \sum_i d_i x_i \leq n \\ & x_i, y_i \in \{0, 1\}, \forall i \end{aligned} \quad (2.22)$$

Es decir:

$$\begin{aligned} & \max_x \left\{ \sum_i (c_i - \lambda b_i) x_i \mid \sum_i d_i x_i \leq n, x_i \in \{0, 1\}, \forall i \right\} \\ & + \max_y \left\{ \lambda \sum_i b_i y_i \mid \sum_i b_i y_i \leq m, y_i \in \{0, 1\}, \forall i \right\} \end{aligned} \quad (2.23)$$

Aquí  $\lambda$  es un multiplicador real simple de signo arbitrario. La cota lagrangiana producida por este esquema esta en medio de la cota de  $LP$  y la cota de descomposición lagrangiana obtenida de dualizar  $x_i = y_i \forall i$ . Este procedimiento es similar al usado para la restricción copiada en [72].

Parece natural que una reducción en el número de multiplicadores debe implicar una reducción en la calidad de la cota obtenida por  $LR$ . Sin embargo, no es siempre el caso, como se muestra en el siguiente ejemplo:

**Ejemplo.** En [11] se considera una relajación lagrangiana agregada al problema de localización de fábricas capacitado. El modelo usa variables continuas  $x_{ij}$  que representan el porcentaje de demanda  $d_j$  del cliente  $j$  suministrado por la fábrica  $i$  y las variables binarias  $y_i$ , igual a 1 si la fábrica  $i$  con capacidad  $a_i$  está operando. La restricción  $\sum_j d_j x_{ij} \leq a_i y_i$  impone una restricción de capacidad condicional en el total de beneficio que puede ser embarcado de una fábrica potencial  $i$ .

$$\begin{aligned}
 \text{mín}_{x,y} \quad & \sum_i \sum_j c_{ij} x_{ij} + \sum_i f_i y_i && (CPLP) \\
 \text{s.a:} \quad & \sum_i x_{ij} = 1 && \forall j \quad (D) \\
 & x_{ij} \leq y_{ij} && \forall i, j \quad (B) \\
 & \sum_i a_{ij} y_i \geq \sum_j d_j && \forall j \quad (T) \\
 & \sum_j d_{ij} x_{ij} \leq a_i y_i && \forall i \quad (C) \\
 & x_{ij} \geq 0, y_i = 0 \text{ ó } 1 && \forall i, j
 \end{aligned} \tag{2.24}$$

donde  $(D)$  se encarga de cumplir el 100% de la demanda del cliente,  $(B)$  no permite embarcar nada si la planta está cerrada,  $(T)$  se encarga de tener las suficientes plantas para satisfacer la demanda total y  $(C)$  prohíbe embarcar más de la capacidad de la planta. Cabe mencionar que la restricción  $(T)$  es redundante, pero puede ayudar a obtener una cota por relajación lagrangiana más delgada.

Los tres mejores esquemas lagrangianos son:

**LR** [38, 77].

Dualizar  $(D)$ , después usar la propiedad de linealización entera. Los subproblemas a resolver son: un problema de la mochila continuo por planta y un problema 0 – 1 de la mochila para todas las plantas. La cota de la relajación lagrangiana es estrecha y es obtenida con un costo computacional pequeño.

**(LD)** [41].

Duplicar  $(T)$ . Hacer copias de  $x_{ij} = x'_{ij}$  y  $y_{ij} = y'_{ij}$ , usar  $x'_{ij}$  y  $y'_{ij}$  en  $(C)$  y una en  $(T)$ . Se obtiene la partición  $\{(D), (B), (T) \rightarrow APLP\}$  (ver [77] para métodos de solución para  $APLP$ )

$$\{(B), (T), (C) \rightarrow \text{esto como en } LR\}$$

Esta cota  $LD$  es más estrecha que la cota obtenida por  $LR$ , pero más cara en calcular, en particular por el gran número de multiplicadores.

**(LS)** [11].

Copiar  $\sum_j d_j x_{ij} = \sum_j d_j x'_{ij}$  y  $y_i = y'_i$  en  $(C)$ . El beneficio es el mismo que el de la partición de  $LD$ , la misma cota, como se muestra en [11]. Sorprendentemente, es más fácil de resolver  $(LS)$  que  $LD$ , en particular porque  $(LS)$  tiene un poco menos de multiplicadores.

En el ejemplo anterior, se crea una nueva copia de variables  $x'_{ij}$  y  $y'_i$ , se puede crear un  $LS$  dualizando una copia de las restricciones agregadas (relacionadas)  $\sum_j d_j x_{ij} = \sum_j d_j x'_{ij}$  y  $a_i y_i = a_i y'_i$ . Puede pasar que la cota  $LS$  para este problema es más fuerte que la cota  $LD$  obtenida por la dualización individual de copias  $x_{ij} = x'_{ij}$  y  $y_{ij} = y'_{ij}$ . Esto sugiere que agregar variables antes puede ser una alternativa atractiva para la descomposición lagrangiana, al menos para algunos problemas estructurados. Una estructura más general que  $CPLP$  está descrito en [11].

### 2.1.6 LAS CARACTERÍSTICAS DE LA FUNCIÓN LAGRANGIANA

La *función lagrangiana*  $z(\lambda) = v(LR_\lambda)$  es una función implícita de  $\lambda$ . Suponga que el conjunto  $\text{conv}\{x \in X \mid Cx \leq d\}$  es un politopo, esto es, un poliedro acotado, entonces existe una familia finita  $\{x^1, x^2, \dots, x^K\}$  de vértices de  $\text{conv}\{x \in X \mid Cx \leq d\}$ , esto es, los puntos de  $\{x \in X \mid Cx \leq d\}$ , como  $\text{conv}\{x \in X \mid Cx \leq d\} = \text{conv}\{x^1, x^2, \dots, x^K\}$ .

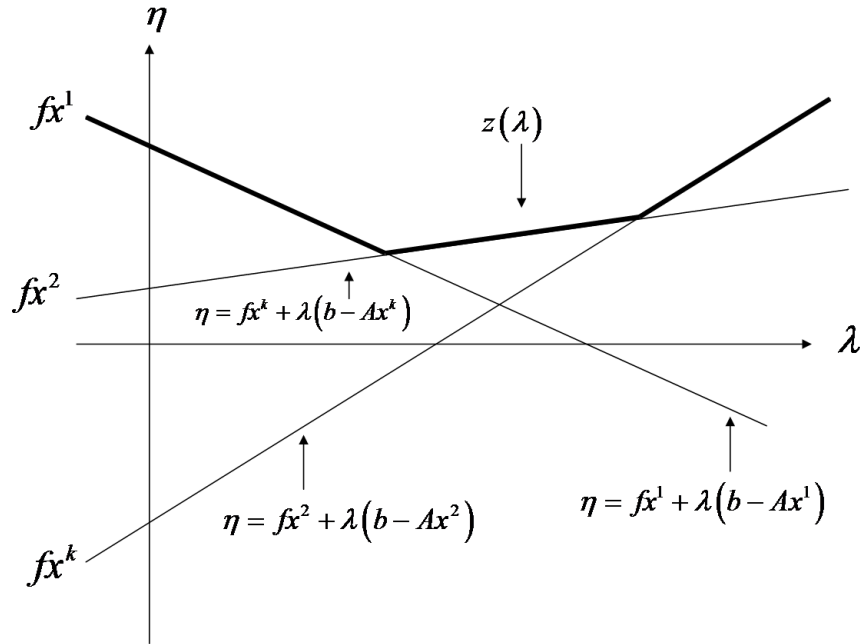


Figura 2.2: Función lagrangiana par el caso de maximización [45]

De aquí sigue que:

$$\min_x \{fx + \lambda(Ax - b) \mid Cx \leq b, x \in X\} = \min_{k=1, \dots, K} \{fx^k + \lambda(Ax^k - b)\} \quad (2.25)$$

y  $z(\lambda)$  es la *cubierta menor* de la familia de funciones lineales de  $\lambda$ ,  $fx^k + \lambda(Ax^k - b)$ ,  $k = 1, \dots, K$  y es una función *cóncava* de  $\lambda$ , con saltos no diferenciable, donde la solución óptima de  $(LR_\lambda)$  no es única. La Figura 2.2 muestra una función lagrangiana para el caso donde  $P$  es un problema de maximización,  $LR$  es un problema de minimización y  $z(\lambda)$  es una función convexa de  $(\lambda)$ .

Una función cóncava  $f(x)$  es continua sobre el dominio interior relacionado y es diferenciable casi en todos lados, esto es, excepto sobre un conjunto de medida 0.

Estos puntos donde no es diferenciable, la función no tiene gradiente, pero siempre tiene subgradiente.

**Definición 2.8** Un vector  $y \in (\mathbb{R}^n)^*$  es un **subgradiente** de una función cóncava  $f(x)$  con el punto  $x^0 \in \mathbb{R}^n$  si  $\forall x \in \mathbb{R}^n$

$$f(x) - f(x^0) \leq y \cdot (x - x^0). \quad (2.26)$$

**Definición 2.9** El conjunto de todos los subgradientes de una función cóncava  $f(x)$  en el punto  $x^0$  es llamado **subdiferenciable** de  $f$  en  $x^0$  y es denotado  $\partial f(x^0)$ .

**Teorema 2.10** El subdiferenciable  $\partial f(x^0)$  de una función cóncava  $f(x)$  en el punto  $x^0$  siempre es no vacío, cerrado, convexo y acotado.

Si el subdiferencial de  $f$  en  $x^0$  consiste de un solo elemento, ese elemento es el subgradiente de  $f$  en  $x^0$ , denotado por  $\nabla f(x^0)$ .

El problema dual  $LR$  es

$$\begin{aligned} \max_{\lambda \geq 0} (LR_\lambda) &= \max_{\lambda \geq 0} z(\lambda) \\ &= \max_{\lambda \geq 0} \min_{k=1, \dots, K} \{fx^k + \lambda(Ax^k - b)\} \\ &= \max_{\lambda \geq 0, \eta} \{\eta \mid \eta \leq fx^k + \lambda(Ax^k - b), k = 1, \dots, K\} \end{aligned} \quad (2.27)$$

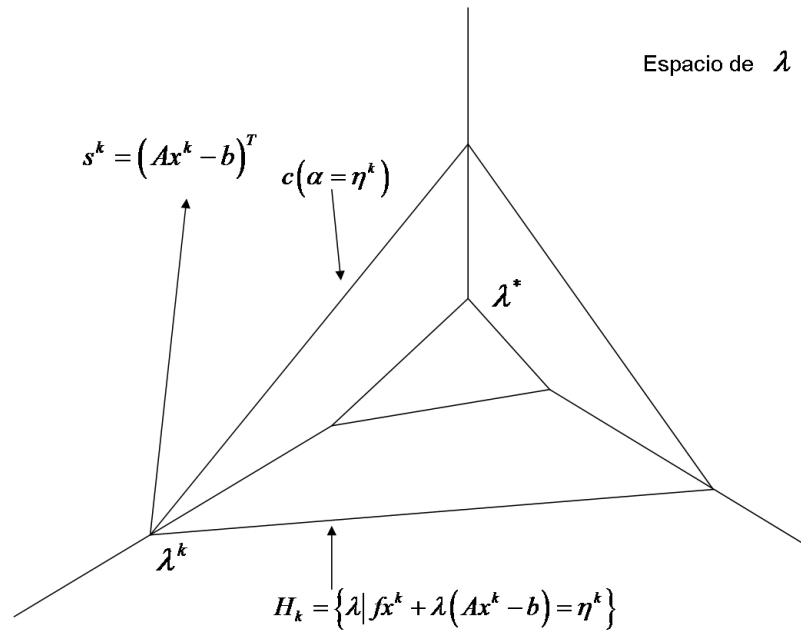
Sea  $\lambda^*$  un minimizador de  $z(\lambda)$ ,  $\eta^* = z(\lambda^*)$ . Sea  $\lambda^k$  un valor “actual” de  $\lambda^*$ , sea  $\eta_k = z(\lambda^k)$  y sea  $H_k = \{\lambda \mid fx^k + \lambda(Ax^k - b) = \eta_k\}$  es un nivel del hiperplano que pasa por  $\lambda^k$ .

- Si  $z(\lambda)$  es diferenciable en  $\lambda^k$ , esto es, si  $(LR_\lambda)$  tiene una única solución óptima  $x^k$ , tiene un gradiente  $\nabla z(\lambda^k)$  en  $\lambda^k$ :

$$\nabla^T z(\lambda^k) = (Ax^k - b) \perp H_k. \quad (2.28)$$

- Si  $z(\lambda)$  es no diferenciable en  $\lambda^k$ , esto es, si  $(LR_\lambda^k)$  tiene soluciones óptimas múltiples, se puede mostrar que el vector  $s^k = (Ax^k - b)^T$  es un subgradiente de  $z(\lambda)$  en  $\lambda^k$ . El vector  $s^k$  es ortogonal a  $H^k$ .





Donde la región  $x^k$  es óptima para  $(LR_\lambda)$

Figura 2.3: Contorno de la función lagrangiana [45]

Se considera el contorno  $C(\alpha) = \{\lambda \in \mathbb{R}_+^m \mid z(\lambda) \geq \alpha\}$ , donde  $\alpha$  es un escalar, estos contornos son conjuntos de poliedros convexos. Ver la Figura 2.3.

## 2.2 MÉTODOS PRIMALES Y DUALES PARA RESOLVER RELAJACIÓN DUAL

Una cantidad de métodos han sido propuesto para resolver duales lagrangianos. Cualquier de ellos es adecuado, como métodos duales ascendentes, o para resolver un problema de optimización convexo no lineal genérico. Esta sección revisa las aproximaciones más importantes.

### 2.2.1 MÉTODO DEL SUBGRADIENTE

El método del subgradiente fue propuesto en [50] y posteriormente fue validado en [51]. Este es un método iterativo en el cual en la iteración  $k$ , da un vector multiplicador actual  $\lambda^k$  y la longitud del paso es tomado con subgradiente  $z(\lambda^k)$ , entonces, si es necesario, el punto resultante es proyectado en un ortogonal no negativo.

Sea  $x^{(k)}$  una solución óptima de  $(LR_{\lambda^k})$ . Entonces  $s^k = (Ax^{(k)} - b)^T$  es un subgradiente de  $z(\lambda)$  en  $\lambda^k$ . Si  $\lambda^*$  es una solución óptima (no conocida) de  $LR$ , con  $\eta^* = z(\lambda^*)$ , sea  $\lambda^{k+1}$  es una proyección de  $\lambda^k$  en un hiperplano  $H^*$  paralelo a  $H_k$  definido por

$$H^* = \{\lambda \mid fx^k + \lambda(Ax^{(k)} - b) = \eta^*\}. \quad (2.29)$$

El vector  $s^k$  es perpendicular a  $H_k$  y  $H^*$ , por lo tanto  $\lambda^{k+1} - \lambda^k$  es un multiplicador no negativo de  $s^k$ :

$$\lambda^{k+1} - \lambda^k = \mu s^k, \mu \geq 0. \quad (2.30)$$

También,  $\lambda^{k+1}$  pertenece a  $H^*$ :

$$fx^{(k)} + \lambda^{k+1}(Ax^{(k)} - b) = \eta^* \quad (2.31)$$

por lo tanto

$$fx^k + \lambda^k(Ax^k - b) + \mu s^k(Ax^{(k)} - b) = \eta^* + \mu s^k \cdot s^k = \eta^* \quad (2.32)$$

y

$$\mu = \frac{\eta^* - \eta^k}{\|s^k\|^2} \quad (2.33)$$

por eso

$$\lambda^{k+1} = \lambda^k + \left[ \frac{s^k(\eta^* - \eta^k)}{\|s^k\|^2} \right] \quad (2.34)$$

Finalmente definimos  $\lambda^{k+1} = [\lambda^{k+1}]^+$ , esto es, define la siguiente iteración  $\lambda^{k+1}$  como la proyección de  $\lambda^{k+1}$  en un ortogonal no negativo, como  $\lambda$  debería ser no negativo. Dadas las proyecciones geométricas descritas, es claro que  $\lambda^{k+1}$  está más cerca de  $\lambda^*$  que de  $\lambda^k$ , así que la secuencia  $\|\lambda^k - \lambda^*\|^2$  es monótonamente no creciente.

**Nota.** *Esta formula desafortunadamente usa el valor óptimo desconocido de  $\eta^*$  de LR. Se puede tratar de usar un estimado para ese valor, pero entonces también podría ser usado un múltiplo más pequeño o más grande de  $s^k$ . Si se ve que los valores de la función objetivo no mejoran en muchas iteraciones, se debe sospechar que  $\eta^*$  ha sido sobreestimado (para un problema de maximización) y se está “pasando”, así se debería tratar de reducir la diferencia entre  $\eta^* - \eta^k$ . Esto puede ser obtenido introduciendo desde el comienzo un factor positivo  $\epsilon_k \in (0, 2)$ , en la formula del subgradiente:*

$$\lambda^{k+1} = \lambda^k + \left[ \frac{s^k \cdot \epsilon_k (\eta^* - \eta^k)}{\|s^k\|^2} \right] \quad (2.35)$$

por lo que  $\epsilon_k$  reduce el escalar cuando este no es mejorado por mucho.

Prácticamente la convergencia del método de subgradiente es impredecible. Para algunos problemas, la convergencia es rápida y altamente confiable, mientras otros problemas tienden a producir multiplicadores en secuencia, o el valor lagrangiano, o ambos, de forma equivocada. En un caso “bueno”, usualmente se podría observar un patrón sawtooth en el valor lagrangiano para las primera iteraciones, seguido por un mejoramiento rigurosamente monótono y convergencia asintótica para un valor optimista del óptimo de la cota lagrangiana. En un caso “malo”, el patrón sawtooth continúa, ó empeora y el valor lagrangiano siempre se está deterio-

rando. Algunos autores han estudiado este problema y han propuesto remedios. Para mejorar el comportamiento del algoritmo mediante el mejoramiento del tamaño del paso para calcular el subgradiente vease [8].

### 2.2.2 MÉTODOS DUALES ASCENDENTES

Este tipo de aproximación, toma la ventaja de la estructura del dual lagrangiano para crear una secuencia de multiplicadores que garantiza un valor monótono creciente en la función lagrangiana. Una aproximación para resolver la aproximación de la relajación  $LP$  del problema de localización de fábricas no capacitado ( $UFLP$ ) ha sido propuesta en [7]. Independientemente en [26] se desarrolló un método dual ascendente para resolver la relajación lagrangiana para el mismo problema de localización no capacitado. Realmente esta cota lagrangiana es igual a la de  $LP$  de [7] debido a la propiedad de integralidad. El tamaño de  $LP$  para el problema  $UFLP$  es muy grande para instancias de tamaño moderado y ambas aproximaciones fueron exitosas para la producción de valores óptimos de  $LP$  en gran parte de los casos tratados. Además las soluciones primas encontradas por el método propuesto en [26] frecuentemente fueron óptimas para  $UFLP$  con un alto porcentaje de soluciones de  $LP$  incluso enteras.

Aunque en general no se puede esperar que la mayoría de las soluciones de  $LP$  siempre pudieran ser enteras debido a que los métodos duales ascendentes normales se concentran en optimizar problemas duales lagrangianos. Éstas aproximaciones son de estructura independiente por lo que son problemas específicos. Algunos ejemplos de diseño dual lagrangiano ascendente exitoso se encuentran en [30, 31, 32, 46]. Si se desea desarrollar un método dual lagrangiano ascendente exitoso vease [42] para ver los principios generales.

### 2.2.3 MÉTODO GENERACIÓN DE RESTRICCIONES

Este método también es llamado método de planos cortantes, o CP, en éste [12] y [54], usaron el hecho de que  $z(\lambda)$  es la cubierta mínima de una familia de funciones lineales:

$$\begin{aligned}
 \max_{\lambda \geq 0} v(LR_\lambda) &= \max_{\lambda \geq 0} z(\lambda) && LR \\
 &= \max_{\lambda \geq 0} \min_{k=1, \dots, K} \{fx^k + \lambda(Ax^k - b)\} \\
 &= \max_{\lambda \geq 0, \eta} \eta \\
 \text{s.a: } &\eta \leq fx^k + \lambda(Ax^k - b) \quad k = 1, \dots, K
 \end{aligned} \tag{2.36}$$

Cada iteración  $k$ , genera uno o más cortes de la forma:

$$\eta \leq fx^k + \lambda(Ax^{(k)} - b), \tag{2.37}$$

para resolver el subproblema lagrangiano  $(LR_\lambda^k)$  con solución  $x^{(k)}$ . Estos cortes son agregados a los generados en iteraciones previas para formar el problema maestro actual  $LP$ :

$$(MP^k): \text{ Hallar } \max_{\lambda \geq 0, \eta} \{\eta \mid \eta \leq fx^{(h)} + \lambda(Ax^{(h)} - b), h = 1, \dots, k\} \tag{2.38}$$

cuya solución es la próxima iteración  $\lambda^{k+1}$ . El proceso termina cuando  $v(MP^k) = z(\lambda^{k+1})$ . Este valor es el óptimo de  $LR$ .

### 2.2.4 GENERACIÓN DE COLUMNAS (CG)

( $CG$ ) ha sido utilizado exhaustivamente, en particular para resolver problemas de secuenciamiento muy grande (aerolíneas, autobuses, ...). Este consiste en

reformular un problema como uno de  $LP$  (o uno  $IP$ ) cuyas actividades (o columnas) corresponde a soluciones factibles de un subconjunto de problemas restringidos, sujetos a las restricciones restantes. Las variables son sujetas a pesos para estas soluciones.

Existen dos aspectos para generar columnas: primero, el proceso es dual para la relajación lagrangiana y para  $CP$ . Posteriormente, puede ser visto como una aplicación del algoritmo de descomposición de [17, 18]. Sea  $x^k \in \{x \in X \mid Cx^k \leq d\}$ ,  $k \in K$ , a ser elegidas tales que  $\text{conv}\{x^k\} = \text{conv}\{x \in X \mid Cx \leq d\}$ . Una elección posible para las  $x^k$ 's son todos los puntos de  $\text{conv}\{x \in X \mid Cx \leq d\}$  pero una opción más barata son todos los vértices de  $\text{conv}\{x \in X \mid Cx \leq d\}$ .

El problema  $P : \min_x \{fx \mid Ax \leq b, Cx \leq d, x \in X\}$  beneficia al problema dual lagrangiano (esto es, en el espacio  $\lambda$ )

$$LR : \text{Hallar } \max_{\lambda \geq 0} \left\{ \min_z \{fx + \lambda(Ax - b) \mid Cx \leq d, x \in X\} \right\} \quad (2.39)$$

el cual es equivalente al problema primal (esto es, en el espacio  $x$ )

$$(PR) : \text{Hallar } \min_x \{fx \mid Ax \leq b, x \in \text{conv}\{x \in X \mid Cx \leq d\}\}$$

y se reescribe como

$$\begin{aligned} \min_x \quad & f \sum_{k \in K} \mu_k x^k && (PR) \\ \text{s. a:} \quad & A \left( \sum_{k \in K} \mu_k x^k \right) x \leq b \\ & \sum_{k \in K} \mu_k = 1 \\ & \mu_k \geq 0 \end{aligned} \quad (2.40)$$

Entonces:

$$\begin{aligned}
& \underset{x}{\text{mín}} && f \sum_{k \in K} \mu_k \cdot f(x^k) \\
& \text{s. a:} && A \left( \sum_{k \in K} \mu_k \cdot Ax^k \right) \leq b \\
& && \sum_{k \in K} \mu_k = 1 \\
& && \mu_k \geq 0
\end{aligned} \tag{2.41}$$

dado que uno puede escribir  $x \in \text{conv}\{x \in X \mid Cx \leq d\}$  como  $x = \sum_{k \in K} \mu_k x^k$ , con  $\sum_{k \in K} \mu_k = 1$  y  $\mu_k \geq 0$ .

La separación de un problema en un subproblema y un maestro es equivalente a la separación de las restricciones que fueron guardadas y las restricciones dualizadas. Las columnas generadas son soluciones de subproblemas enteros que tienen las mismas restricciones que los subproblemas lagrangianos. La generación de las columnas fue usado para instancias en [79] para la relajación lagrangiana más fuerte del *GAP*. Las cotas obtenidas usualmente fueron muy delgadas, esto es, más cerradas del verdadero valor de *IP* que la cota *LP*.

El valor de la relajación lagrangiana del problema maestro es igual a la cota de la relajación lagrangiana. La fuerza del esquema de una *CG* o *LR* podría haberse basado en el hecho de que los subproblemas no tienen la propiedad de integralidad. Sin embargo puede pasar que tal esquema puede ser exitoso para resolver problemas con la propiedad de integralidad porque esto permite el cálculo indirecto de  $v(LP)$  cuando este valor no puede ser calculado directamente, por ejemplo, a causa de un número exponencial de restricciones, [49, 50].

Una ventaja substancial de (*CP*) o (*CG*) sobre los algoritmos subgradientes es la existencia de un criterio de terminación  $v(MP^k) = z(\lambda^{k+1})$ .

Aunque para ciertas familias de problemas, por ejemplo problemas de lote capacitados multi producto con o sin tiempos de ensamble, [43], (*CG*) puede converger rápidamente (en no más de 15 a 20 iteraciones en la aplicación de tamaño de lote), frecuentemente en la práctica el proceso es generado con suficientes restricciones o

columnas (en  $CP$  o  $CG$  respectivamente) para conseguir la convergencia la cual toma un tiempo largo para obtenerla. Primero el paso inicial solo se consideran un poco de columnas/restricciones las cuales son conocidas y la aproximación de la función lagrangiana puede ser totalmente pobre. Puede tomar tiempo hasta que la familia de restricciones/columnas generadas permitan una localización relativa exacta del vector multiplicador óptimo. Posteriormente hacia el final del proceso, algunos de los problemas que son altamente degenerados por muchas iteraciones no muestran un verdadero mejoramiento en cualquier multiplicador o valor lagrangiano. Se han realizado muchos intentos para corregir este comportamiento. En [23, 24, 86] se describen algunas aproximaciones de este método.

### 2.2.5 MÉTODOS DE FUNCIONES LAGRANGIANAS MODIFICADAS.

Los métodos de agregación lagrangiana han sido usados principalmente en programación continua no lineal y en optimización estocástica. Sin embargo también pueden ser usados en programación entera no lineal ( $NLIP$ ) – y como consecuencia también en programación lineal entera – resolviendo directamente problemas de relajación primal, a diferencia de resolver problemas en el espacio dual. Tal aproximación para un caso lineal puede ser encontrado en [22].

En [40] se incluye una relajación primal para  $NLIP$ . La cual es equivalente a la relajación lagrangiana en un caso lineal, pero usualmente no lo es para un caso no lineal.

La *Relajación de un Problema Primal* de un problema de programación entera no lineal es:

$$(IP) : \text{Hallar } \min_x \{f(x) \mid g(x) = 0, x \in P \cap X\} \quad (2.42)$$

relacionado a las restricciones de igualdad  $g(x) = 0$ , con un poliedro racional  $P$  y  $X$  un conjunto de restricciones de integralidad sobre las variables, es el problema



continuo no lineal

$$(PR): \text{ Hallar } \inf_x \{f(x) \mid g(x) = 0, x \in \text{conv}\{P \cap X\}\} \quad (2.43)$$

Si la función  $f(x)$  es convexa y  $g(x)$  lineal,  $(PR)$  es equivalente al problema dual lagrangiano como lo hace [74]

$$(LR^*): \text{ Hallar } \max_{\lambda} \left\{ \inf_x f(x) + \lambda g(x) \mid x \in \text{conv}\{P \cap X\} \right\} \quad (2.44)$$

Por otro lado,  $(PR)$  es equivalente al problema *Proximal para el Lagrangiano Modificado*

$$\begin{aligned} \inf_x \quad & \left\{ f(x) + \left(\frac{\alpha}{2\rho}\right) |x - x^*|^2 + u^* \cdot g(x) + \frac{1}{2}\rho |g(x)|^2 \right\} \quad (PLM) \\ \text{s. a:} \quad & x \in \text{conv}\{P \cap X\} \end{aligned} \quad (2.45)$$

para cualquier  $\rho > 0$ , suficientemente grande y cualquier  $\alpha$  positivo, donde  $x^*$  es una solución óptima del problema original  $(PR)$  y  $u^*$  el multiplicador óptimo asociado a las restricciones dualizadas  $g(x) = 0$ . Como se muestra en [14],  $(PLM)$  puede ser resuelto por una adaptación del método proximal de multiplicadores, el cual toma en cuenta las restricciones implicadas  $x \in \text{conv}\{P \cap X\}$ . La función proximal para el lagrangiano modificado,  $L(x, w, u, \alpha, \rho) = f(x) + \left(\frac{\alpha}{2\rho}\right) |x - w|^2 + u \cdot g(x) + \left(\frac{1}{2}\right)\rho |g(x)|^2$  depende de la aproximación de  $u$  a  $u^*$ , la aproximación de  $w$  a  $x^*$ , el parámetro proximal  $\alpha$  y el parámetro penalizador  $\rho$ .

Existen varias ventajas de usar un *agregado lagrangiano* a un *método de penalización*. Primero, existe un valor finito  $\bar{\rho}$  del coeficiente de penalización  $\rho$  tal que para cualquier  $\rho \geq \bar{\rho}$ , el problema  $(PLM)$  es equivalente al problema  $(PR)$ . Después, los multiplicadores son actualizados por medio de una forma-cerrada, fijando el paso en la formula del subgradiente  $u_i(k+1) = u_i(k) + \rho(A_i x(k) - b_i)$ , que garantice la *convergencia* al multiplicador óptimo de Kuhn-Tucker  $u_i^*$ , *sin necesidad de ajustar o estimar un parámetro*, como debe ser en el caso de métodos de subgradiente.

Finalmente, en el caso lineal, la convergencia es conseguida en un *número finito de iteraciones*.

La ventaja de incluir un *término proximal*  $(\frac{\alpha}{2\rho})|x - x^*|^2$ ,  $\alpha > 0$ , es tal que si  $f(x)$  y  $g(x)$  son convexos,  $L(x, w, u, \alpha, \rho)$  es *estrictamente* convexa en  $x$  y tiene un *único* mínimo sobre  $x$  dados  $w, u, \alpha, \rho$ . (PAL) puede ser resuelto por un método de *linealización* tal como el método de Frank y Wolfe, o, por descomposición simple, conocido por propiedades de convergencia mejorada.

La ventaja de usar  $\text{conv}\{P \cap X\}$  en lugar de  $P \cap X$  es que el problema (PLM) puede ser resuelto eficientemente por métodos de linealización tales como descomposición simple porque este conjunto de restricciones es un poliedro, mientras generalmente el problema

$$\begin{aligned} \min_x \quad & f(x) + (\frac{\alpha}{2\rho})|x - x^*|^2 + u^* \cdot (Ax - b) + (\frac{1}{2})\rho|Ax - b|^2 \\ \text{s. a:} \quad & x \in (P \cap X) \end{aligned} \tag{2.46}$$

no lo es.

En [15, 16] se describe una implementación exitosa del proximal para el lagrangiano agregado para resolver problemas de localización de fábricas capacitado con función objetivo no lineal.

### 2.2.6 MÉTODOS HÍBRIDOS DE DOS FASES

En [44] se presentó un método que combina el método del subgradiente en una primera fase y generación de restricciones en la segunda. Los multiplicadores primero son ajustados de acuerdo a la formula del subgradiente y al mismo tiempo, las restricciones correspondientes a todas las soluciones conocidas de los subproblemas conocidos son agregados a la LP del problema maestro. El valor de LP del problema maestro es tomado como una estimación actual del dual lagrangiano óptimo. La exactitud de esta estimación va en aumento conforme vayan pasando las iteraciones,

por eso no hay necesidad que para cualquier ajuste del tamaño del paso: cuidar que  $\epsilon_k = 1, \forall k$ . Dado un intervalo en el óptimo dual de la cota de relajación lagrangiana y el valor del problema maestro se ve beneficiada la convergencia del método, como un método de generación de restricciones puro.

Debe estar seguro que el proceso no es cíclico. Si se obtienen restricciones repetidas, el problema maestro no puede ser mejorado. Después de que el mismo corte ha sido generado un número de veces (digamos, 5 veces), se puede cambiar a una fase de generación de restricciones pura.

Un método híbrido similar ha sido abogado recientemente en [47], combinando (*CG*) con el método subgradiente. Sabemos que es difícil de generar un buen conjunto de columnas que comiencen el algoritmo. El artículo sugiere usar una fase inicial que genere las “paredes de afuera” del domo de la función lagrangiana, usando una explicación gráfica del procedimiento. Si se ve la función lagrangiana (cóncava) como un domo en  $\mathbb{R}$ , donde  $\mathbb{R}^m$  corresponde a los multiplicadores lagrangianos  $\lambda$  y  $\mathbb{R}$  a la función lagrangiana, entonces en la fase inicial, se tratará de generar fases además definir un poliedro acotado que contienen el domo lagrangiano. En la Figura 2.4, las 3 fases definidas por las líneas firmes (son las “paredes de afuera”, aunque no todas) definen un poliedro acotado. Las líneas punteadas definen las fases del domo que puede ser descubiertas durante el transcurso del algoritmo.

La fase inicial usa un método sugradiente con un paso grande, el propósito es “pasarse” para descubrir las paredes de afuera. Una vez que el problema maestro de generación de columnas es factible (o equivalentemente una vez que el plano cortante *LP* es factible), o posiblemente algunas iteraciones después si uno piensa existe algunas ventajas en generar un poco más de fases, se cambia cualquier restricción o columna de generación. Además sabemos que las columnas generadas en la primera fase del método (*CG*) para el *GAP*, contienen más información adicional que aquellas generadas por el método estándar “fase 1” (esto es, se empieza con columnas artificiales con alto costo), como evidenciando el factor de que pocas nuevas columnas necesitan ser generadas para converger.

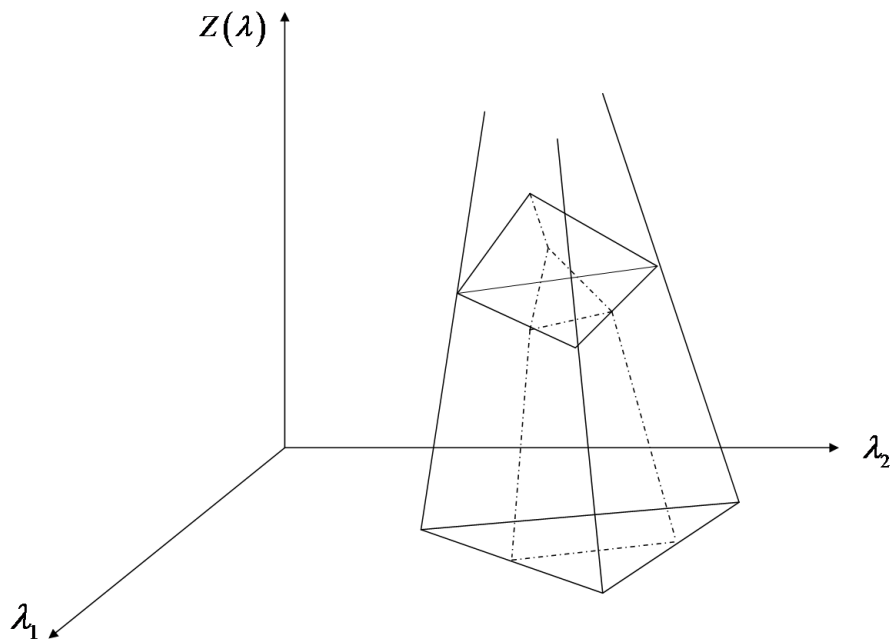


Figura 2.4: Definición de un poliedro acotado [45]

### 2.2.7 DESCOMPOSICIÓN DE SUBPROBLEMAS

En algunos casos, el subproblema lagrangiano se descompone en problemas pequeños, esto significa que la región factible es realmente el producto cartesiano de varias regiones pequeñas. Una ventaja clara es la reducción en la complejidad computacional para los subproblemas lagrangianos: generalmente es más fácil resolver 50 problemas con 100 variables binarias cada uno, que, un problema con 5,000 (esto es,  $50 \times 100$ ) variables binarias.

También significa que en generación de columnas, las columnas (esto es, los vectores que son soluciones factibles de las restricciones guardadas) se descomponen en pequeñas subcolumnas y cada subcolumna es una combinación convexa de vértices de una pequeña región. Por asignación de conjuntos de pesos diferentes a estas combinaciones convexas, sigue “mezclar y relacionar” las soluciones, en otras palabras, se trata de combinar una subcolumna para el primer subproblema que

fue generado en la iteración 10, con una subcolumna para el segundo subproblema generado en la iteración 7, etc., para formar una columna de tamaño completo. Si no se tiene la descomposición del problema antes del tiempo, podríamos esperar un largo tiempo para generar completamente la columna.

Por dualidad, esto significa que en un ambiente de *plano cortante*, también pueden ser generados “sub cortes” para cada subproblema, esta cantidad para el primer reemplazo  $\eta$  por  $z + \lambda b$  en

$$\begin{aligned}
 & \underset{\lambda \geq 0, \eta}{\text{máx}} && \eta && (MP^k) \\
 \text{s.a:} & && \eta \leq f x^{(h)} + \lambda(Ax^{(h)} - b) && h = 1, \dots, k
 \end{aligned} \tag{2.47}$$

$$\begin{aligned}
 & \underset{\lambda \geq 0, z}{\text{máx}} && z + \lambda b \\
 \text{s. a:} & && z \leq (f + \lambda A)x^{(h)} && h = 1, \dots, k
 \end{aligned}$$

y entonces  $z$  por una suma de escalares  $z_l$ , con  $z_l \leq (f^l + \lambda A_l)x_l^{(h)}$ , donde  $l$  es el índice del subproblema lagrangiano,  $f^l$ ,  $A_l$  y  $x_l^{(h)}$  son la  $l$ -ésima porción de la submatriz y vectores correspondientes y  $x_l^{(h)}$  es una solución lagrangiana del  $l$ -ésimo subproblema encontrado en la iteración  $h$ , beneficiado la desagregación del problema maestro

$$\begin{aligned}
 & \underset{\lambda \geq 0, z_l}{\text{máx}} && \sum_l z_l + \lambda b && (MPD^k) \\
 \text{s. a:} & && z_l \leq (f + \lambda A)^l x_l^{(h)} && h = 1, \dots, k
 \end{aligned} \tag{2.48}$$

**Ejemplo.** Considere de nuevo el GAP para el caso de minimización (aunque esto trabajaría en exactamente la misma forma con maximización). Hemos visto que la relajación lagrangiana fuerte es:

$$\begin{aligned}
 & \underset{i,j}{\text{mín}} && \sum_{i,j} c_{ij} x_{ij} + \sum_j \lambda_j (1 - \sum_i x_{ij}) && (LR_\lambda) \\
 \text{s.a} & && \sum_j a_{ij} x_{ij} \leq b_i && \forall i \in I && (KP) \\
 & && x_{ij} \in \{0, 1\}, && \forall i \in I, j \in J
 \end{aligned} \tag{2.49}$$

$$\begin{aligned}
\text{mín} \quad & \sum_{i,j} (c_{ij} - \lambda_j) x_{ij} + \sum_j \lambda_j \\
\text{s. a:} \quad & \sum_j a_{ij} x_{ij} \leq b_i, \forall i \\
& x_{ij} \in \{0, 1\}, \forall i, j
\end{aligned} \tag{2.50}$$

$$\sum_j \lambda_j + \sum_j \text{mín} \left\{ \sum_j (c_{ij} - \lambda_j) x_{ij} \mid \sum_j a_{ij} x_{ij} \leq b_i, \forall i, x_{ij} \in \{0, 1\}, \forall i, j \right\}$$

y  $LR$  es el máximo con respecto a  $\lambda$  de  $v(LR_\lambda)$ .

Sea  $EP(KP) = \{x^k \mid k \in K\}$  es el conjunto de todas las soluciones factibles de las restricciones  $(KP)$ , sea  $EP(KP_i) = \{x_i^k \mid k \in K_i\}$  es el conjunto de todas las soluciones enteras factibles de la  $i$ -ésima mochila, con  $K = \Pi_i K_i$ .

Entonces la solución factible de  $(LR_\lambda)$  puede ser descrito por:

$$x_{ij} = \sum_{k \in K_i} \mu_k^i x_{ij}^k, \forall i, j. \tag{2.51}$$

El dual lagrangiano es equivalente a agregar el problema maestro  $AMP$ : hallar

$$\text{máx}_{\lambda, \zeta} \left\{ \zeta \mid \zeta \leq \sum_{i,j} c_{ij} x_{ij}^k + \sum_j \lambda_j \left( 1 - \sum_i x_{ij}^k \right), k \in K \right\} \tag{2.52}$$

es decir

$$= \text{máx}_{\lambda, z} \left\{ z + \sum_j \lambda_j \mid z \leq \sum_{i,j} (c_{ij} - \lambda_j) x_{ij}^k, \forall k \in K \right\} \tag{2.53}$$

con sustitución  $\zeta = z + \sum_j \lambda_j$ .

Si primero tenemos que escribir la formulación de la generación de columnas para el dual lagrangiano, naturalmente podemos tener las soluciones de ambos sub-problemas de la mochila independientes, usando los conjuntos  $K_i$  independientes a diferencia de  $K$ , la generación de columnas del problema maestro podría ser desagregado:

$$(DMP): \text{ Hallar } \max_{\lambda, z} \left\{ \sum_i z_i + \sum_j \lambda_j \mid z_i \leq \sum_j (c_{ij} - \lambda_j) x_{ij}^k, \forall i, k \in K_i \right\} \quad (2.54)$$

y el dual es:

$$\begin{aligned} \min_{\mu} \quad & \sum_{k \in K_i} \sum_j c_{ij} x_{ij}^k \mu_k^{(i)} \\ \text{s. a:} \quad & \sum_{k \in K_i} \sum_j x_{ij}^k \mu_k^{(i)} \quad \forall j \\ & \sum_{k \in K_i} \mu_k^{(i)} = 1 \quad \forall i \\ & \mu_k^i \geq 0 \end{aligned} \quad (2.55)$$

está claro que la descomposición Dantzig-Wolfe del primal equivalente ( $PR$ ) es:

$$(PR): \text{ Hallar } \min_x \left\{ \sum_{i,j} c_{ij} x_{ij} \mid \sum_i x_{ij} = 1, x_{ij} \geq 0 \right\} \quad (2.56)$$

de  $LR$ .

### 2.2.8 RELAJAR Y CORTAR (“RELAX & CUT”)

Una pregunta que frecuentemente vemos en el contexto de relajación lagrangiana es: ¿cuál es la fortaleza de la cota obtenida por la relajación lagrangiana?. Una posible respuesta es la adición de cortes que son violados actualmente por la solución lagrangiana. Sin embargo, está claro que agregar estos al problema lagrangiano podría cambiar esta estructura y lo haría mucho más difícil de resolver. Una salida es dualizar estos cortes. ¡Recordemos que dualizando no significa descartar!. Los cortes pueden ser agregados al conjunto de “restricciones complicadas” e intuitivamente pueden ser usados solo si la intersección N1 (para “una nueva intersección”) de un poliedro nuevo relajado y del envolvente convexo de las soluciones enteras de restricciones guardadas es “más pequeño” que la intersección OI (para “una intersección

antigua”) del poliedro antiguo relajado y el envolvente convexo de las soluciones enteras de restricciones guardadas. Solo es posible en este turno si el poliedro relajado nuevo es más pequeño que el antiguo, entonces las restricciones guardadas son las mismas en ambos casos. Esto tiene las siguientes implicaciones. Considere un corte que es violado por la solución lagrangiana actual:

1. si el corte es igual a una combinación convexa de las restricciones dualizadas actuales o guardadas, no es posible reducir la intersección desde cada punto de intersecciones “antiguas”; además no ayuda *sustituir restricciones de las dualizadas*. Ver Figura 2.5.
2. si el corte es una desigualdad válida para el problema lagrangiano, entonces cada punto en el envolvente convexo de puntos enteros de las restricciones guardadas se satisface por cada solución entera factible del subproblema lagrangiano;
3. por lo que es necesario usar para el corte la información “entera” y dualizar y guardar ambas restricciones y remover *parte de la intersección*. (La solución lagrangiana es un punto entero requerido para satisfacer solo las restricciones guardadas).

Un esquema de relajar y cortar puede proceder como sigue:

1. inicializar el multiplicador lagrangiano  $\lambda$ .
2. resolver el problema lagrangiano actual, sea  $x(\lambda)$  la solución lagrangiana. Si el dual lagrangiano no ha sido resuelto aún, actualizar  $\lambda$ . Si no terminar.
3. Identificar el corte que fue violado por  $x(\lambda)$  y dualizarlo. Ir a (2).

El termino “dualizar y cortar” fue usado por primera vez en [25]. Este artículo, usó una descripción parcial de restricciones y las cuales fueron identificadas como restricciones violadas (no cortes), agregadas al modelo e inmediatamente dualizadas.



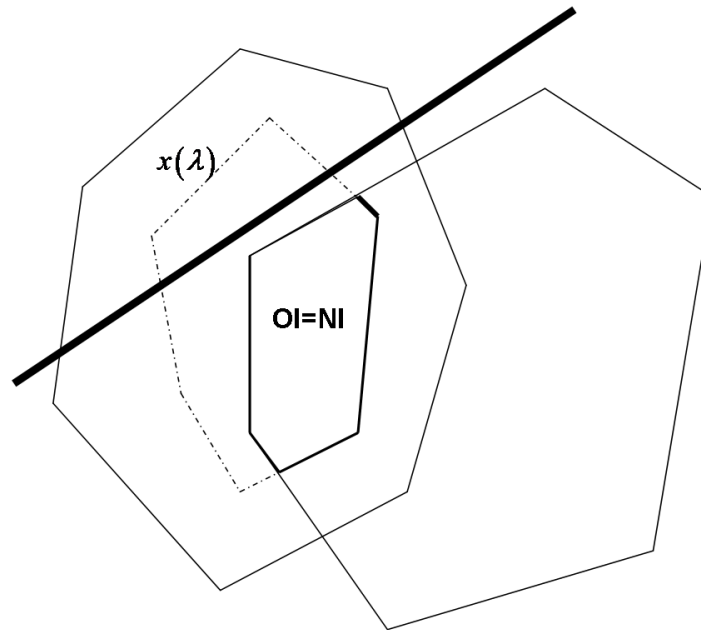


Figura 2.5: Conjunto convexo entero de soluciones que no cambia [45]

La idea, sino el nombre, ha sido usado tempranamente. En [3].se generó un subtour de eliminación de restricciones e inmediatamente se dualizó, esto fue aplicado a instancias del problema *TSP*. Lucena usó una idea similar en [62]. La utilidad de las restricciones es obvia, contraria a la de estos cortes. Obviamente una restricción perdida puede cambiar la solución del problema.

Ahora daremos ejemplo de *cortes* que si fue posible dualizarlos pero no pueden ser cotas de relajación lagrangianas delgadas.

CORTES DUALIZADOS NO MEJORADOS: EJEMPLO PARA EL *GAP*. Ya hemos introducido el *GAP* y este es el modelo:

$$\begin{aligned}
 \text{mín} \quad & \sum_i \sum_j c_{ij} x_{ij} && (GAP) \\
 \text{s. a} \quad & \sum_j a_{ij} x_{ij} \leq b_i \quad \forall i \in I && (KP) \\
 & \sum_i x_{ij} = 1 \quad \forall j \in J && (MC) \\
 & x_{ij} \in \{0, 1\} \quad \forall i \in I, j \in J && 
 \end{aligned} \tag{2.57}$$

Si se dualiza ( $MC$ ), el problema de relajación lagrangiana se descompone en un subproblema por  $j$ :

$$\begin{aligned}
 \text{mín} \quad & \sum_{I,j} c_{ij}x_{ij} + \sum_j \lambda_j(1 - \sum_i x_{ij}) && (LR_\lambda) \\
 \text{s. a} \quad & \sum_j a_{ij}x_{ij} \leq b_i && \forall i \in I \quad (KP) \\
 & x_{ij} \in \{0, 1\} && \forall i \in I, j \in J
 \end{aligned} \tag{2.58}$$

Se puede escribir como:

$$\text{mín} \left\{ \sum_{i,j} (c_{ij} - \lambda_j)x_{ij} + \sum_j \lambda_j \mid \sum_j a_{ij}x_{ij} \leq b_i, \forall i, x_{ij} \in \{0, 1\}, \forall i, j \right\} \tag{2.59}$$

es decir

$$\text{mín} \sum_j \lambda_j + \sum_i \text{mín} \left\{ \sum_j (c_{ij} - \lambda_j)x_{ij} \mid \sum_j a_{ij}x_{ij} \leq b_i, \forall i, x_{ij} \in \{0, 1\}, \forall i, j \right\} \tag{2.60}$$

Así el  $i$ -ésimo subproblema lagrangiano es un problema de la mochila para la  $i$ -ésima máquina. Después de resolver todos los problemas de la mochila, la solución  $x(\lambda)$  puede violar algunas restricciones de opción múltiple, esto es, existe alguna  $j$  para la cual  $\sum_i x_{ij} \neq 1$  y como una consecuencia la condición  $\sum_i \sum_j x_{ij} = |J|$  puede ser violada. Agregando este “corte” (es decir, ¡recortar la solución lagrangiana actual!) e inmediatamente dualizarlo, sin reducir la intersección, cada punto de la intersección antigua OI ya satisface todas las restricciones de múltiple opción ( $MC$ ), esto es, las restricciones dualizadas.

¿SE PUEDEN GUARDAR LOS CORTES MÁS FUERTES DE LA COTA LAGRANGIANA?  
 ¿Pero qué pasa si guardamos los cortes en lugar de dualizarlos?. Está claro que agregar estos al problema lagrangiano puede cambiar esta estructura, pero todavía puede ser resuelta fácilmente. Los cortes pueden ser agregados al conjunto de “restricciones fáciles” e intuitivamente pueden ser usados solo si la intersección NI del poliedro relajado y del nuevo envolvente convexo de las soluciones enteras de restricciones guardadas es “más pequeño” que la intersección OI del poliedro relajado y del

envolvente convexo antiguo de las soluciones enteras de las restricciones guardadas. Esto sólo es posible si el envolvente convexo del envolvente nuevo es más pequeño que el antiguo, entonces las restricciones dualizadas son las mismas en ambos casos.

**Ejemplo.** *Considere de nuevo el GAP y la relajación lagrangiana débil: en la cual las restricciones de la mochila (KP) son dualizadas. Se puede agregar las restricciones restantes de múltiple opción y sustituir una restricción de las restricciones dualizadas, por ejemplo la suma de todas las restricciones, las cuales obviamente es más débil que la restricción de la mochila original. El problema lagrangiano no se descompone en más, pero esta nueva estructura es del problema de la mochila de múltiple opción, la cual usualmente es fácil de resolver con software especializado y mucho más fácil que agregar las restricciones de la mochila sin múltiple opción.*

El fortalecimiento de la cota lagrangiana es simple y aún no es potencialmente poderosa.

### 2.2.9 HEURÍSTICAS LAGRANGIANAS Y BRANCH-AND-PRICE

La relajación lagrangiana da cotas, pero también genera soluciones lagrangianas. Si la solución lagrangiana satisface las restricciones de holgura complementaria ( $CS$ ), sabemos que es una solución óptima del problema  $IP$ . Si este es factible pero  $CP$  se cumple, esto es al menos una solución factible del problema  $IP$  y aún se tiene que determinar, por  $BB$  o de otra forma, este es óptimo. Si no, la relajación genera soluciones enteras infactibles. Frecuentemente todas estas soluciones están cerca de la factible, las cuales se pueden obtener mediante penalizaciones para las restricciones más violadas. Existe un amplia gama de literatura acerca de posibles formas de modificar las soluciones lagrangianas existentes infactibles a factibles. Las heurísticas lagrangianas son esencialmente dependientes del problema y solo trataremos de obtener un poco de pistas sobre como proceder. Por ejemplo se puede tratar de obtener soluciones factibles de la siguiente forma:

1. modificando la solución correcta en infactibilidades cuidando que el deterioro

de la función objetivo sea pequeño.

Ejemplo: en el secuenciamiento de la producción, si relaja las restricciones de demanda, se puede tratar de cambiar la producción (incrementar o disminuir) hasta conocer la demanda [20].

2. fijando (en 1 ó 0) algunos de las variables de decisión significativas según sus valores actuales en la solución lagrangiana y resolviendo óptimamente el problema restante. Podemos llamar a esta heurística “perezosa” [10]. Para empezar se puede fijar las variables que satisfacen las restricciones relajadas.

Parte del éxito de la relajación lagrangiana se debe al mejoramiento de manera inteligente de los métodos de solución para el dual lagrangiano, con heurísticas poderosas grabadas en cada iteración. En algunos casos, el intervalo restante de la dualidad, esto es, el porcentaje relativo entre la mejor cota lagrangiana encontrada y la mejor solución factible encontrada por la heurística es suficientemente pequeña para realizar una enumeración. Sin embargo en algunos ejemplos un óptimo o al menos una solución óptima es deseada y puede ser usado un esquema de ramificar y acotar para remplazar las cotas  $LP$  por  $LR$ . Si el dual lagrangiano es resuelto por generación de columnas, el esquema es llamado “*branch and price*”, como nuevas columnas podrían necesitadas para ser “*priced-out*” como un ramificado guardado [5, 21]. En este caso, las reglas de ramificado necesitan ser diseñadas con cuidado como fue hecho en [76]. La esperanza es que estos esquemas pueden converger más rápido que  $LP$  el cual está basado en ramificar y acotar, normalmente las cotas podrían ser más delgadas y cortarse nodos más rápido. La cantidad de trabajo realizado en un nodo, puede ser sustancialmente resuelto como un  $LP$ .

- La relajación lagrangiana es una familia de herramientas poderosa para resolver de forma aproximada los problemas de programación entera. Dando
  - Cotas más fuertes que la relajación  $LP$  cuando el problema(s) no tengan la propiedad de integralidad.

- Buenos puntos de comienzo para heurísticas de búsqueda.
- La disponibilidad de interfaces poderosas (GAMS, AMPL, . . .) y la flexibilidad de paquetes IP que hacen esto posible para el usuario para varios esquemas e implementar y probarlos.
- La relajación lagrangiana es muy flexible. Frecuentemente algunas reformulaciones son necesarias para que aparezca un buen esquema.
- No es necesario tener estructuras especiales grabadas en un problema para tratar de usar esquemas lagrangianos. Probablemente es una buena idea descomponer el problema estructuralmente en componentes significativos y partirlos a través de dualización de restricciones, posiblemente después de haber introducido nuevas expresiones de variables.
- Finalmente las soluciones para uno o más subproblemas lagrangianos se puede prestar para emplear heurísticas lagrangianas, posiblemente seguidas por heurísticas intercambiables, para obtener buenas soluciones factibles.
- Emparejar las cotas de relajación lagrangiana con heurísticas lagrangianas dando el análisis con brackets alrededor del valor entero óptimo. Usualmente estos son más delgados que los brackets viniendo de cotas basadas en  $LP$  y heurísticas.

## CAPÍTULO 3

# CONSTRUCCIÓN DE COTAS

---

En este capítulo se muestra paso a paso la formulación del problema para encontrar la mejor cota lagrangiana así como el algoritmo utilizado durante los experimentos computacionales. También se muestran algunas aplicaciones al algoritmo, su formulación y la definición de las relajaciones lagrangianas para cada caso.

### 3.1 CONSTRUCCIONES BÁSICAS

Supongamos que tenemos  $K$  presentaciones lagrangianas del mismo problema original, donde la  $k$ -ésima presentación ( $k \in K = \{1, 2, \dots, K\}$ ) tiene la forma:

$$z = \max \{cx \mid D_k x \leq d_k, x \in X_k\} \quad (3.1)$$

Donde  $D_k = d_k$  son las  $m_k$  restricciones complicadas, las restricciones  $x \in X_k$  son agradables, en el sentido de que el problema de optimización formado únicamente con las restricciones agradables es más fácil de resolver que el problema original.

Denotamos por  $x^*$  la solución óptima de (3.1), lo mismo para toda  $k$  donde (3.1) se obtiene del problema original simplemente reordenando las restricciones.

Sea  $u_k \geq 0$  un  $m$ -vector de multiplicadores de Lagrange. El  $k$ -ésimo problema lagrangiano se define como:

$$z_k(u_k) = \max_{x \in X_k} \{cx + u_k(d_k - D_kx)\} \quad (3.2)$$

Asumimos por simplicidad que se tiene una solución óptima finita para todo  $u_k \geq 0$ .  $x^*$  es factible para el problema lagrangiano, entonces:

$$cx^* + u_k(d_k - D_kx^*) \leq z_k(u_k) \quad (3.3)$$

para toda  $u_k \geq 0$

Como  $x^*$  es factible para (3.1) y  $u_k \geq 0$ , tenemos que  $u_k(d_k - D_kx^*) \geq 0$ , lo que resulta en la bien conocida cota Lagrangiana

$$z^* \leq z_k(u_k) \quad (3.4)$$

para cualquier  $u_k \geq 0$

La  $k$ -ésima mejor cota lagrangiana y los multiplicadores lagrangianos correspondientes  $u_k^*$  se definen del problema dual lagrangiano:

$$z_k(u_k^*) = z_k = \min_{u_k \geq 0} z_k(u_k) \quad (3.5)$$

La mejor cota entre todas las cotas duales lagrangianas se define como:

$$z_D = \min_{k \in K} \{z_k\} = \min_{k \in K} \left\{ \min_{u_k \geq 0} z_k(u_k) \right\} = \min_{u_k \geq 0} \min_{k \in K} \{z_k(u_k)\} \quad (3.6)$$

Donde  $u = (u_1, u_2, \dots, u_k)$ . Se utiliza  $z(u)$  para denotar al  $\min_{k \in K} \{z_k(u_k)\}$

En lo siguiente asumimos que todas las  $z_k(u_k)$  están acotadas para  $u_k \geq 0$  tal que

$$0 \leq z_k(u_k) \leq C \quad (3.7)$$

Introduciendo las variables binarias  $y_k \in \{0, 1\}$  y usando la representación del  $\min_k \{T_k\}$  para las restricciones enteras mixtas (Wolsey, 1998) podemos expresar  $z(u)$  para cualquier  $u \geq 0$  como:

$$\begin{aligned} 0 \leq z(u) \leq z_k(u_k) \leq C \\ z_k(u_k) - z(u) \leq C(1 - y_k), y_k \in \{0, 1\} \\ \sum_k y_k = 1, k \in K \end{aligned} \tag{3.8}$$

Entonces el problema (3.6) se escribe como:

$$\min_{u \geq 0} z(u) \tag{3.9}$$

s.a.(3.8) con  $z_k(u_k)$  definido en (3.2).

Asumimos ahora por simplicidad que cada conjunto  $X_k$  tiene un número muy grande pero finito de puntos  $\{x_1, x_2, \dots, x_{T_k}\}$  o es un poliedro acotado con vértices correspondientes. Denotamos como  $T_k = \{1, 2, \dots, T_k\}$ . Usando la definición de  $z_k(u_k)$  formulamos el siguiente problema maestro:

$$\begin{aligned} \min z \\ z_k \geq cx_{t_k} + u_k(d_k - D_k x_{t_k}), t_k \in T_k \\ z_k - z \leq C(1 - y_k), 0 \leq z \leq z_k \leq C \\ y_k \in \{0, 1\}, k \in K \\ \sum_k y_k = 1 \end{aligned} \tag{3.10}$$

Donde  $z_k$  representa a  $z_k(u_k)$  y  $z$  a  $z(u)$ .

**Proposición 1.** El valor óptimo del problema maestro (3.10) coincide con el valor óptimo del problema (3.5).



**Demostración.** Sean  $\bar{z}, \bar{y}_k, \bar{z}_k$  y  $\bar{u}_k$  la solución óptima del problema maestro (3.10). Donde (3.10) es una relajación del problema (3.9) que es equivalente al problema (3.6), entonces  $\bar{z} \leq z_D$

Ahora demostramos que siempre existe una solución óptima de (3.10) factible a (3.9). Sin pérdida de generalidad podemos asumir que  $\bar{y}_1 = 1$  y entonces, por las restricciones del problema maestro (3.10), sabemos que  $\bar{y}_k = 0$  para  $k > 1$ . Entonces, tenemos  $\bar{z} = \bar{z}_1 \leq \bar{z}_k$  para  $k > 1$ . Si  $\bar{z}_1 > \max_{t_1 \in T_1} \{cx_{t_1} + \bar{u}_1(d_1 - D_1x_{t_1})\}$ , podemos decrementar  $\bar{z}_1$  (manteniendo  $\bar{z} = \bar{z}_1$ ) sin pérdida de factibilidad en el problema maestro (3.10), lo cual contradice la optimalidad de  $\bar{z}$ . Entonces,

$$\bar{z} = \bar{z}_1 = \max_{t_1 \in T_1} \{cx_{t_1} + \bar{u}_1(d_1 - D_1x_{t_1})\} = \max_{x \in X_1} \{cx + \bar{u}_1(d_1 - D_1x)\} \quad (3.11)$$

Sean  $K_ =$  y  $K_ <$  subconjuntos de  $K$  que cumplen que  $\bar{z}_1 = \bar{z}_k$  y  $\bar{z}_1 < \bar{z}_k$  respectivamente. Como se hizo anteriormente, no es difícil demostrar que

$$\bar{z} = \bar{z}_1 = \bar{z}_k = \max_{t_k \in T_k} \{cx_{t_k} + \bar{u}_k(d_k - D_kx_{t_k})\}, k \in K_ = \quad (3.12)$$

Mientras tanto, podemos tener

$$\bar{z}_k > \max_{t_k \in T_k} \{cx_{t_k} + \bar{u}_k(d_k - D_kx)\}, k \in K_ < \quad (3.13)$$

En este caso, podemos construir otra solución óptima del problema maestro (3.10) decrementando  $\bar{z}_k$  para  $k \in K_ <$  para cumplir con las expresiones mencionadas como igualdades. Note que no podemos decrementar  $\bar{z}_k$  para  $k \in K_ <$  por debajo de  $\bar{z}_1$  ya que esto contradice la optimalidad de  $\bar{z}_1$ .

Entonces, siempre existe la solución óptima al problema maestro (3.10) que cumple:

$$\bar{z}_k = \max_{x \in X_k} \{cx + \bar{u}_k(d_k - D_kx)\} \quad (3.14)$$

y por lo tanto factible al problema (3.9). De aquí sigue que  $\bar{z} \geq z_D$  y con esto se completa la demostración.

**Observación 1.** Supongamos que  $\bar{z} = \bar{z}_1 < \bar{z}_k$  para  $k > 1$ . Esto es, la cota lagrangiana  $\bar{z}_1$  es estrictamente más fuerte que todas las otras cotas lagrangianas. En este caso siempre existe la solución óptima del problema maestro (3.10) con  $\bar{z}_k > \max_{t_k \in T_k} \{cx_{t_k} + \bar{u}_k(d_k - D_k x_{t_k})\}$  para  $k \neq 1$  y por lo tanto, las restricciones  $z_k \geq cx_{t_k} + u_k(d_k - D_k x_{t_k}), t_k \in T_k, k \neq 1$  no están activas para esta solución óptima. Así, entre todas las restricciones del problema maestro (3.10) podemos esperar relativamente pocas restricciones activas.

El problema maestro (3.10) tiene un número muy grande de restricciones del tipo  $z_k \geq cx_{t_k} + u_k(d_k - D_k x_{t_k}), t_k \in T_k$ , una para cada  $x_{t_k} \in X_k, k \in K$ . Para resolver el problema maestro (3.10) usamos la técnica de relajación de restricciones en la forma del algoritmo de descomposición de Benders.

Consideramos que en la  $r$ -ésima iteración tenemos un problema de maestro restringido con pocas restricciones de tipo  $z_k \geq cx_{t_k} + u_k(d_k - D_k x_{t_k})$  en comparación con el problema maestro (3.10). Nótese que sólo relajamos restricciones de ese tipo, las demás restricciones siempre son incluidas en el problema maestro restringido. Supongamos que se incluyen en el problema maestro las restricciones de tipo  $z_k \geq cx_{t_k} + u_k(d_k - D_k x_{t_k})$  para toda  $t_k \in T_k^r \subseteq T_k$ . Denotamos su solución óptima como  $(u_k, z_k, y_k, z)^r$ . Para verificar la factibilidad de esta solución en todas las restricciones de ese tipo, necesitamos verificar:

$$z_k^r \geq cx_{t_k} + u_k^r(d_k - D_k x_{t_k}), \forall t_k \in T_k, k \in K \quad (3.15)$$

o de manera equivalente:

$$z_k^r \geq \max_{x \in X_k} \{cx + u_k^r(d_k - D_k x)\}, \forall k \in K \quad (3.16)$$

Esto es, para verificar (3.16) necesitamos resolver  $K$  problemas lagrangianos independientes para multiplicadores fijos  $u_k^r$ . Denotando como  $x_k^r$  sus soluciones óptimas.

Si las condiciones (3.16) se satisfacen, entonces paramos con  $(u_k, z_k, y_k, z)^r$  óptimo para el problema maestro (3.10). Si (3.16) falla, denotamos como  $K_r \subseteq K$  el conjunto indexado de restricciones (3.16) violadas y añadimos:

$$z^r \leq z_D \leq \min_{s=1,2,\dots,r} \left\{ \min_{k \in K} z_k(u_k^r) \right\} \quad (3.17)$$

Donde el mín exterior se toma sobre todas las iteraciones previas. Entonces, podemos parar si la diferencia entre las cotas en (3.17) es más pequeña que un umbral dado.

**Observación 2.** El problema maestro (restringido) es un problema de optimización entero-mixto. Por la restricción  $\sum_k y_k = 1, y_k \in \{0, 1\}$  sigue que sólo un componente  $y_k$  es positivo e igual a 1, mientras que los demás son 0. Entonces, podemos tratar de resolver el problema maestro restringido por enumeración completa, calculando  $z^r(y_k)$ , el objetivo óptimo del problema maestro restringido para  $y_k = 1$  y entonces el conjunto  $z = \min_k z^r(y_k)$ . Por ejemplo, para  $y_1 = 1, y_k = 0$  para  $k \neq 1$ , el problema maestro restringido se vuelve:

$$\begin{aligned} z^r(y_1) &= \min z_1 \\ z_k &\geq cx_{t_k} + u_k(d_k - D_k x_{t_k}), t_k \in T_k^r \subseteq T_k, \\ z_1 &\leq z_k, u_k \geq 0, 0 \leq z_k \leq C, k \in K. \end{aligned} \quad (3.18)$$

Si este problema es infactible, entonces continuará siendo infactible para todas las iteraciones siguientes en el problema maestro restringido (ya que éste solo añade restricciones en cada iteración). En este caso  $y_1 = 1$  no será la solución óptima del problema maestro (3.10) y por lo tanto el dual lagrangiano (3.9) para  $y_1 = 1$  no proveerá la mínima cota. Por lo tanto, podemos simplemente excluir la relajación lagrangiana número 1 de consideración.

## 3.2 ALGORITMO

Se muestra el algoritmo utilizado durante esta investigación, como se contruye la solución inicial del método, el problema maestro y subproblema así como el criterio de parada que emplea.

**Paso 0.** *Formular el problema maestro inicial.*

Resolver la relajación lineal del problema original, obteniendo el óptimo de las variables duales y el valor objetivo óptimo  $z_{lp}$ . Fijar  $u_k^0 \geq 0, k \in K$ . (Por ejemplo,  $u_k^0$  puede ser el vector de duales óptimos de la relajación lineal). Sean  $C = z_{lp}$ , y  $v$  un contador de iteraciones.

Para toda  $k \in K$  resolvemos los problemas lagrangianos:

$$z_k(u_k^0) = \max_{x \in X_k} \{cx + u_k^0(d_k - D_k x)\} \quad (3.19)$$

Denotemos por  $x_k^0$  la solución óptima del problema lagrangiano. Usando esta solución óptima formamos el problema maestro inicial.

$$\begin{aligned} & \text{mín } z \\ & z_k \geq cx_k^0 + u_k(d_k - D_k x_k^0) \\ & z_k - z \leq C(1 - y_k), 0 \leq z \leq z_k \leq C \\ & y_k \in \{0, 1\}, u_k \geq 0, k \in K \\ & \sum_k y_k = 1 \end{aligned} \quad (3.20)$$

e ir al paso 1.

**Paso 1.** *Resolver el problema maestro inicial.*

En la iteración  $v$  tenemos el problema maestro (3.10v)

$$\begin{aligned}
& \text{mín } z \\
& z_k \geq cx_{t_k} + u_k (d_k - D_k x_{t_k}), t_k \in T_k^v \subseteq T_k \\
& z_k - z \leq C (1 - y_k), 0 \leq z \leq z_k \leq C \\
& y_k \in \{0, 1\}, u_k \geq 0, k \in K \\
& \sum_k y_k = 1
\end{aligned} \tag{3.21}$$

El problema maestro (3.10<sub>v</sub>) es resuelto por enumeración completa como sigue:

Para cada  $l \in K$  hacer: Fijar  $y_l = 1, y_k = 0, k \neq l$ . Resolver el problema

$$\begin{aligned}
& \theta_l = \text{mín } z_l \\
& z_k \geq cx_{t_k} + u_k (d_k - D_k x_{t_k}), t_k \in T_k^v \subseteq T_k \\
& z_l \leq z_k, u_k \geq 0, 0 \leq z_k \leq C, k \in K
\end{aligned} \tag{3.22}$$

Si (3.22) es infactible, el conjunto  $K = K - \{l\}$  y excluimos el problema lagrangiano  $l$  de consideración ya que no provee la mejor (mínima) cota lagrangiana.

**Criterio de parada.**  $|K| = 1$

Si  $K$  contiene un único elemento entonces PARAR. Éste único elemento será el número de la mejor cota lagrangiana. Podemos continuar iterando con el problema maestro para calcular el valor de esta cota o podemos calcularla por cualquier otro método.

Si el criterio de parada no se cumple, es decir, si  $|K| \geq 2$ , entonces calcular  $\theta_{l_0} = \text{mín}_{l \in K} \theta_l$ .

Sea  $(z_k, u_k)^{l_0}$  la solución óptima del problema (3.22<sub>0</sub>), entonces la solución óptima del problema (3.10<sub>v</sub>)  $(z, z_k, y_k, u_k)^v$  es:

$$z^v = z_{l_0}, (z_k, u_k)^v = (z_k, u_k)^{l_0}, y_{l_0}^v = 1, y_l = 0, l \neq l_0 \tag{3.23}$$

e ir al paso 2.

**Paso 2.** *Resolver los problemas lagrangianos.*

Para todo  $k \in K$  resolver el problema lagrangiano

$$z_k(u_k^v) = \max_{x \in X_k} \{cx + u_k^v(d_k - D_kx)\} \quad (3.24)$$

Denotamos como  $x_k^v$  a la solución óptima del problema lagrangiano.

**Paso 3.** *Añadir restricción.*

Para cada  $k \in K$ , añadimos la restricción

$$z_k \geq cx_k^v + u_k(d_k - D_kx_k^v) \quad (3.25)$$

al problema maestro (3.10<sub>v</sub>) formando el problema (3.10<sub>v+1</sub>).

Ir al paso 1 con  $v = v + 1$ .

### 3.3 APLICACIÓN AL PROBLEMA DE MOCHILA MULTIDIMENSIONAL

El problema de mochila multidimensional (MKP) 0-1 es uno de los problemas más conocidos en la programación entera. Básicamente es un modelo de asignación de recursos. Cada  $j \in N$  requiere  $a_{ij}$  unidades de consumo de recursos en la  $i$ -ésima restricción mochila ( $i = 1, 2, \dots, m$ ) con un beneficio de  $c_{ij}$ .

El objetivo es encontrar un subconjunto de elementos que proporcionen el máximo beneficio denotado por  $z^*$ , sin exceder la capacidad de recursos  $b_i$ .

#### 3.3.1 FORMULACIÓN DEL PROBLEMA MKP

El problema de mochila binaria multidimensional consiste en maximizar una función lineal de variables 0 – 1 que satisfacen  $m$  restricciones de tipo mochila:

$$\text{máx } z = \sum_{j \in N} c_j x_j$$

s.a:

$$\begin{aligned} \sum_{j \in N} a_{ij} x_j &\leq b_i, i \in M = \{1, 2, \dots, m\} \\ x_j &\in \{0, 1\}, j \in N = \{1, 2, \dots, n\} \end{aligned} \quad (3.26)$$

Todas las entradas son no negativas, podemos suponer sin pérdida de generalidad que  $c_j > 0, b_i > 0, a_{ij} \leq b_i, \sum_{j \in N} a_{ij} > b_i \forall j \in N$  y  $\forall i \in M$ .

### 3.3.2 DEFINICIÓN DE RELAJACIONES

Se definen  $m$  relajaciones distintas para el MKP con  $M$  restricciones tipo mochila de la siguiente manera:

Para cada  $k = (1, 2, \dots, m)$ . Sea  $i \in I = \{1, 2, \dots, m\}$ , y sea  $i' \in I - \{k\}$ .

$$\text{max} \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{i=1}^m u_i \left[ b_{i'} - \sum_{j=1}^n a_{i'j} x_{i'j} \right] \quad (3.27)$$

s.a.

$$\begin{aligned} \sum_{j=1}^n a_{kj} x_{kj} &\leq b_k \\ x &\in \{0, 1\}, u_i \geq 0 \end{aligned} \quad (3.28)$$

## 3.4 APLICACIÓN AL PROBLEMA DE ASIGNACIÓN GENERALIZADA

Los problemas de asignación (AP) involucran la asignación óptima de elementos de dos o más conjuntos. Cuando tenemos solo dos conjuntos, nos podemos referir a ellos como tareas y agentes. En su versión original, el AP involucra la asignación de

cada tarea a un agente distinto, con cada agente asignado a lo más a una tarea (asignación uno a uno). En el problema de asignación generalizada (GAP) cada tarea es asignada a un agente, como en el AP original, pero permite la posibilidad de que un agente sea asignado a más de una tarea, reconociendo que una tarea puede solo usar una parte de la capacidad total del agente. Entonces, el GAP es un problema de asignación uno a varios que reconoce los límites de capacidad [64], [70]. Consideraremos una generalización del AP el cual es una asignación varias - varios reconociendo las capacidades límites de tareas y agentes. Diferentes técnicas basadas en relajaciones lagrangianas han sido propuestas para resolver el GAP [52], [66].

### 3.4.1 FORMULACIÓN DEL PROBLEMA GAP

El problema de asignación generalizada se modela como sigue:

$$\text{máx} \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

s.a.

$$\sum_{j=1}^n a_{ij} x_{ij} \leq b_i, i = 1, 2, \dots, m \quad (3.29)$$

$$\sum_{i=1}^m d_{ij} x_{ij} \leq f_j, j = 1, 2, \dots, n \quad (3.30)$$

$$x \in \{0, 1\}$$

donde  $i$  es el conjunto de agentes,  $j$  el conjunto de tareas,  $c_{ij}$  utilidad de asignar el agente  $i$  a la tarea  $j$ ,  $a_{ij}$  capacidad utilizada del agente  $i$  para realizar la tarea  $j$ ,  $b_i$  capacidad disponible del agente  $i$ ,  $d_{ij}$  capacidad (tiempo) utilizada de la tarea  $j$  cuando es realizada por el agente  $i$  y  $f_j$  capacidad (tiempo) disponible de la tarea  $j$ , y la variable binaria  $x_{ij} = 1$  si el agente  $i$  es asignado a la tarea  $j$ , y 0 en otro caso.



## 3.4.2 DEFINICIÓN DE RELAJACIONES

Se definen 2 relajaciones distintas para el GAP, la primera relajando las restricciones de tipo (3.29), y la segunda relajando las restricciones de tipo (3.30) como sigue:

Relajación 1

$$\text{máx} \sum_{i=1}^m u_i b_i + \sum_{i=1}^m \sum_{j=1}^n (c_{ij} - a_{ij}) x_{ij}$$

s.a.

$$\begin{aligned} \sum_{i=1}^m d_{ij} x_{ij} &\leq f_j \\ x_{ij} &\in \{0, 1\} \end{aligned} \tag{3.31}$$

Relajación 2

$$\text{máx} \sum_{j=1}^n u_j f_j + \sum_{i=1}^m \sum_{j=1}^n (c_{ij} - d_{ij}) x_{ij}$$

s.a.

$$\begin{aligned} \sum_{j=1}^n a_{ij} x_{ij} &\leq b_i \\ x_{ij} &\in \{0, 1\} \end{aligned} \tag{3.32}$$

## CAPÍTULO 4

# EXPERIMENTACIÓN COMPUTACIONAL

---

El objetivo de la experimentación computacional es poder comparar la relación entre la calidad de las cotas y el valor objetivo óptimo de los problemas, así los tiempos empleados para encontrar la mejor relajación contra el tiempo que tarda en calcular todas las cotas lagrangianas posibles para después elegir la mejor de ellas.

### 4.1 GENERALIDADES

Compararemos los resultados de varias instancias del problema de Mochila Multidimensional (MKP) y el problema de Asignación Generalizada (GAP). Se muestran algunas tablas de resultados, los demás resultados son similares y se pueden observar en el apéndice A de este trabajo de investigación.

Los datos utilizados son generados de manera uniforme entre un intervalo dado. Las cotas se calculan utilizando el método de descomposición de Benders.

### 4.2 RESULTADOS EXPERIMENTALES MKP

En esta sección se muestran las tablas de resultados para 3 tamaños de instancias:

- 20 variables de decisión y 10 restricciones (pequeñas).

- 40 variables de decisión y 20 restricciones (medianas).
- 60 variables de decisión y 30 restricciones (grandes).

Para el problema MKP se generó la matriz de datos con una estructura particular en la que las submatrices que se forman en la diagonal tienen coeficientes mucho más grandes que los elementos que no pertenecen a estas submatrices, además las submatrices comparten al menos una variable con coeficiente *grande* con alguna de las otras submatrices como se puede observar a continuación:

$$\begin{aligned} \text{máx } z &= \sum c_j x_j \\ \mathbf{a}_{11}\mathbf{x}_1 + \mathbf{a}_{12}\mathbf{x}_2 + \mathbf{a}_{13}\mathbf{x}_3 + a_{14}x_4 + a_{15}x_5 + a_{16}x_6 + a_{17}x_7 &\leq b_1 \\ \mathbf{a}_{21}\mathbf{x}_1 + \mathbf{a}_{22}\mathbf{x}_2 + \mathbf{a}_{23}\mathbf{x}_3 + a_{24}x_4 + a_{25}x_5 + a_{26}x_6 + a_{27}x_7 &\leq b_2 \\ a_{31}x_1 + a_{32}x_2 + \mathbf{a}_{33}\mathbf{x}_3 + \mathbf{a}_{34}\mathbf{x}_4 + \mathbf{a}_{35}\mathbf{x}_5 + a_{36}x_6 + a_{37}x_7 &\leq b_3 \\ a_{41}x_1 + a_{42}x_2 + \mathbf{a}_{43}\mathbf{x}_3 + \mathbf{a}_{44}\mathbf{x}_4 + \mathbf{a}_{45}\mathbf{x}_5 + a_{46}x_6 + a_{47}x_7 &\leq b_4 \\ a_{51}x_1 + a_{52}x_2 + a_{53}x_3 + a_{54}x_4 + \mathbf{a}_{55}\mathbf{x}_5 + \mathbf{a}_{56}\mathbf{x}_6 + \mathbf{a}_{57}\mathbf{x}_7 &\leq b_5 \\ a_{61}x_1 + a_{62}x_2 + a_{63}x_3 + a_{64}x_4 + \mathbf{a}_{65}\mathbf{x}_5 + \mathbf{a}_{66}\mathbf{x}_6 + \mathbf{a}_{67}\mathbf{x}_7 &\leq b_6 \\ x_j &\in \{0, 1\}, j \in N = \{1, 2, 3, \dots, n\} \end{aligned}$$

Los coeficientes que pertenecen a las submatrices dadas por los coeficientes en negrita son mucho más grandes que los que están fuera de ellos.

Estos datos se generan de la siguiente manera:

$a_{ij} \in U [10, 25]$  si  $a_{ij}$  pertenece a alguna de las submatrices.

$a_{ij} \in U [0, 0.07]$  si no pertenece.

$c_j \in U [100, 120]$

$b_i \in \left[ \left[ 0.6 \sum_{j=1}^n a_{ij} \right], \left[ 0.9 \sum_{j=1}^n a_{ij} \right] \right]$

donde U es una distribución uniforme dentro del intervalo dado.

Para cada instancia del problema se calcularon los siguientes valores:

<b>n</b>	<b>m</b>	$t_i$	$t_c$	$t_p$	<b>ahorro( %)</b>	$gap_1$	$gap_2$
20	10	1.82	1.86	2.95	37	13.102	19.18
20	10	1.44	1.46	2.61	45	22.84	30.09
20	10	2.39	2.42	2.47	7	9.98	19.86
20	10	1.96	2.02	3.61	44	11.08	24.19
20	10	2.2	2.23	3.25	32	17.14	21.24
20	10	1.24	1.33	2.19	39.27	19.28	26.36
20	10	1.36	1.54	3.41	54.83	18.19	24.79
20	10	1.37	1.45	2.64	45.08	24.2	32.33
20	10	1.18	1.4	2.27	38.33	6.74	12.28
20	10	1.08	1.2	2.74	56.2	6.43	11.85

Tabla 4.1: Resultados para casos pequeños MKP

- $z_{ip}$  solución óptima del problema original entero.
- $z_m$  mejor cota lagrangiana.
- $z_p$  peor cota lagrangiana.
- $z_l$  solución óptima de la relajación lineal.

Y además se midió la calidad de las cotas con respecto a la proximidad existente con la solución óptima del problema original, y fueron calculadas así:

$$\begin{aligned}
 gap_1 &= \frac{z_m - z_{ip}}{z_{ip}} 100 \% \\
 gap_2 &= \frac{z_l - z_{ip}}{z_{ip}} 100 \%
 \end{aligned}
 \tag{4.1}$$

donde  $gap_1$  es la proximidad de la solución óptima entera a la mejor cota lagrangiana, y  $gap_2$  a la cota de la relajación lineal.

En las tablas se muestran los resultado para los 3 tamaños de instancias, podemos ver el número de variables de decisión y restricciones en las columnas 1 y 2

<b>n</b>	<b>m</b>	$t_i$	$t_c$	$t_p$	<b>ahorro(%)</b>	$gap_1$	$gap_2$
40	20	11.14	11.55	21.27	46	6.75	8.54
40	20	11.32	11.4	19.68	43	9.89	11.59
40	20	8.47	8.75	14.54	40	8.68	10.77
40	20	11.96	12.09	16.39	26.24	7.032	8.41
40	20	9.25	9.58	18.2	47.36	0.935	1.23
40	20	10.99	11.26	17.27	34.8	9.43	10.95
40	20	48.98	49.04	88.17	44.38	8.64	10.5
40	20	32.44	34.63	71.41	51.5	8.6	11.1
40	20	41.56	41.97	58.21	27.9	10.38	12.62
40	20	29.24	29.97	57.05	47.48	3.3	4.52

Tabla 4.2: Resultados para casos medianos MKP

respectivamente, después se muestra el tiempo que le toma al algoritmo saber cuál es el índice de la mejor relajación lagrangiana ( $t_i$ ), en la cuarta columna se muestra el tiempo que tarda en dar el valor de la mejor cota al seguir iterando con el método de Benders ( $t_c$ ), en la quinta muestra el tiempo que tomaría calcular todas las posibles cotas lagrangianas para después elegir la mejor ( $t_p$ ), después se muestra el ahorro promedio de tiempo entre  $t_p$  y  $t_c$  y en las últimas 3 columnas se muestran los resultados del  $gap_1$ ,  $gap_2$  y  $gap_3$ .

Se muestra además, una tabla con algunos resultados para instancias más grandes del problema MKP para 100 variables de decisión y 50 restricciones.

También se muestra una tabla con el resumen de los resultados para algunas otras instancias con resultados similares, sus tablas completas pueden encontrarse en el apéndice A de este trabajo.

Esta tabla contiene el número de variables de decisión y restricciones, en la tercera columna aparece la media del tiempo que tarda el algoritmo en encontrar la mejor cota lagrangiana, después aparece la media del tiempo si calculamos todas las posibles cotas y al final nos muestra la media del ahorro promedio entre estos dos

<b>n</b>	<b>m</b>	$t_i$	$t_c$	$t_p$	<b>ahorro( %)</b>	$gap_1$	$gap_2$
60	30	10.96	11.22	22.95	51	4.12	5.4
60	30	42.54	42.93	66.62	36	11.72	13.4
60	30	24.77	25.31	99.8	74	10.5	12.41
60	30	16.32	16.75	33.32	49.73	5.44	6.94
60	30	40.49	41.37	162.85	74.59	12.36	14.3
60	30	39.68	40.78	108.18	62.3	8.99	10.96
60	30	25.79	25.79	30.88	16.5	4.9	5.45
60	30	51.19	51.82	100.15	48.26	10.92	12.23
60	30	14.96	15.23	28.9	47.3	1.86	4.9
60	30	39.01	40.34	111.11	63.69	11.37	13.3

Tabla 4.3: Resultados para casos grandes MKP

<b>n</b>	<b>m</b>	$t_i$	$t_c$	$t_p$	<b>ahorro( %)</b>	$gap_1$	$gap_2$
100	50	288.22	290.14	1721.36	83	10.84	11.93
100	50	440.62	450.25	2580.89	82	8.49	9.23
100	50	553.105	562.37	2621.6	78.55	7.49	9.16
100	50	487.66	492.14	2511.54	80.41	8.64	11.04
100	50	627.43	632.52	1259.08	49.76	9.6	10.54

Tabla 4.4: Algunos otros resultados MKP

tiempos.

Podemos mencionar que la diferencia de el tiempo para instancias del mismo tamaño se debe a que depende unicamente de los datos de cada problema, pero se puede observar que en los problemas en los que toma mucho tiempo encontrar el valor de la mejor cota lagrangiana por medio del algoritmo propuesto también es mayor el tiempo que tarda calcular todas las cotas posibles.

<b>n</b>	<b>m</b>	$m_c$	$m_p$	$m_a$
5	3	0.281	0.3583	22.12
10	5	1.049	1.57	32.24
20	10	6.087	10.409	40.91
40	20	33.9	58.7	41.92
60	30	66.29	129.79	49.77
100	50	485.484	2138.894	74.744

Tabla 4.5: Cálculo de medias para diferentes tamaños del problema MKP

### 4.2.1 ANÁLISIS ESTADÍSTICO DE RESULTADOS MKP

En esta sección se realizan las pruebas estadísticas necesarias para establecer algunas conclusiones sobre los datos que se obtuvieron en la experimentación computacional.

Los datos con los cuales se realizan estas pruebas con tomados de las tablas que se presentan en el apéndice de resultados, se cuenta con 155 instancias de diversos tamaños y lo que se busca probar es:

- Que el tiempo empleado por el algoritmo propuesto es menor al tiempo usado para calcular todas las cotas.
- Que la mejor cota lagrangiana domina a la cota lineal.

### PRUEBA DE HIPÓTESIS PARA DIFERENCIA DE MEDIAS MKP

Par empezar haremos la prueba para comparar el tiempo que tarda el algoritmo que se propone en este trabajo contra el tiempo de calcular cada relajación lagrangiana posible. Lo que queremos demostrar estadísticamente es que la media obtenida por el tiempo del algoritmo es mejor que la media obtenida por cel tiempo de calcular todas las cotas lagrangianas.

Criterios para rechazar  $H_0$ :

Hipótesis nula	Hipótesis alternativa	Rechaza si:
$H_0:\mu_1 = \mu_2$	$H_1:\mu_1 \neq \mu_2$	$ Z  \geq Z_{\alpha/2}$
$H_0:\mu_1 = \mu_2$	$H_1:\mu_1 > \mu_2$	$Z \geq Z_{\alpha}$
$H_0:\mu_1 = \mu_2$	$H_1:\mu_1 < \mu_2$	$ Z  \geq Z_{\alpha}$

Consideramos un  $\alpha = 0.05$ , es decir, un intervalo de confianza del 95 % y tenemos:

$H_0 : \mu_p = \mu_c$  donde  $\mu_p$  es la media del tiempo de calcular todas las cotas y  $\mu_c$  es la media del tiempo de encontrar la mejor cota en base al algoritmo propuesto, y

$H_1 : \mu_p > \mu_c$

- $\bar{X}_p = 106.3552$  se calculan las respectivas medias muestrales.
- $\bar{X}_c = 34.4341$
- $\sigma_p^2 = 162040.769$
- $\sigma_c^2 = 8709.777$
- $n_1 = n_2 = 155$

Se calcula el estadístico de prueba  $Z$  como:

$$Z_0 = \frac{\bar{X}_p - \bar{X}_c}{\sqrt{\frac{\sigma_p^2}{n_1} + \frac{\sigma_c^2}{n_2}}}$$

$$Z_0 = 2.1669$$

$$Z_{\alpha} = Z_{0.05} = 1.65$$

Para comprobar nuestra hipótesis debemos tener  $Z > Z_{\alpha}$ , y entonces comparamos los resultados obtenidos y tenemos que  $2.1669 > 1.65$ .



Por lo tanto se rechaza  $H_0$ , por lo que aceptamos  $H_1$ , es decir, que la media del tiempo usado por el algoritmo es menor que la media del tiempo de calcular cada relajación lagrangiana para elegir la mejor después.

Ahora se quiere probar estadísticamente que la cota lagrangiana es mejor en un que la cota lineal con respecto a su cercanía con el valor óptimo del problema con un  $\alpha = 0.01$ .

Criterios para rechazar  $H_0$ :

Hipótesis nula	Hipótesis alternativa	Rechaza si:
$H_0: \mu_d = d$	$H_1: \mu_d \neq d$	$ Z  \geq Z_{\alpha/2}$
$H_0: \mu_d = d$	$H_1: \mu_d > d$	$Z \geq Z_{\alpha}$
$H_0: \mu_d = d$	$H_1: \mu_d < d$	$ Z  \leq -Z_{\alpha}$

Se tiene entonces,  $H_0 : \mu_d = 60\%$  donde  $\mu_d$  es la media de diferencias entre la cota lagrangiana y la lineal,  $H_1 : \mu_d > 60\%$ . Entonces:

- $\bar{d} = 8.494$
- $\mu_d = 60\% = 5.097$
- $\sigma_d = 9.025$
- $n = 155$

El estadístico de prueba que se usa es:

$$Z_0 = \frac{\bar{d} - \mu_d}{\sigma_d} \sqrt{n}$$

$$Z_0 = 4.68$$

Para rechazar la hipótesis nula necesitamos que  $Z_0 > Z_{\alpha}$ , entonces comparamos los resultados que se obtuvieron.

$$Z_\alpha = Z_{0.01} = 3.09$$

Entonces  $4.68 > 3.09$ , por lo que rechazamos la hipótesis nula, por lo tanto hemos comprobado de forma estadística que el promedio de mejoría de la mejor cota lagrangiana sobre la cota lineal con respecto a la cercanía de ellas al valor objetivo óptimo es mayor a un 60 %.

### 4.3 RESULTADOS EXPERIMENTALES GAP

Se mostrarán algunos resultados del algoritmo aplicado al problema de Asignación Generalizada (GAP) de 2 tamaños distintos:

- 10 tareas y 5 agentes (pequeñas).
- 20 tareas y 5 agentes (medianas).

Los datos se generaron de la siguiente manera:

$$\begin{aligned} a_{ij} &\in U [20, 30] \\ d_{ij} &\in U [1, 10] \\ f_j &\in \left[ \left[ 0.6 \sum_{j=1}^n d_{ij} \right], \left[ 0.9 \sum_{j=1}^n d_{ij} \right] \right] \\ b_i &\in \left[ \left[ 0.6 \sum_{j=1}^n a_{ij} \right], \left[ 0.9 \sum_{j=1}^n a_{ij} \right] \right] \\ c_{i,j} &\in U [100, 120] \end{aligned}$$

donde U es una distribución uniforme dentro del intervalo dado.

Para cada instancia del problema se calcularon los siguientes valores:

- $z_{ip}$  solución óptima del problema original entero.
- $z_m$  mejor cota lagrangiana.

<b>n</b>	<b>m</b>	$t_i$	$t_c$	$t_p$	<b>ahorro( %)</b>	$gap_1$	$gap_2$
10	5	0.03	0.3	0.52	42.3	3.29	9.32
10	5	0.36	0.57	0.75	24	0.19	6.95
10	5	0.18	0.44	0.72	38.9	3.26	8.59
10	5	0.18	0.38	0.53	28.3	2.92	7.56
10	5	0.06	0.22	0.65	66.15	0.68	7.95
10	5	1.01	3.33	4.78	30.33	3.31	8.71
10	5	0.31	0.84	3.34	74.85	0.25	12.72
10	5	0.39	1.24	2.58	51.93	3.16	8.47
10	5	0.12	0.38	2.08	81.73	0.59	7.22
10	5	0.75	1.97	2.57	23.34	4.8	8.39

Tabla 4.6: Resultados para casos pequeños GAP

- $z_p$  peor cota lagrangiana.
- $z_l$  solución óptima de la relajación lineal.

Y nuevamente se midió la calidad de las cotas con respecto a la proximidad existente con la solución óptima del problema original de la siguiente manera:

$$\begin{aligned}
 gap_1 &= \frac{z_m - z_{ip}}{z_{ip}} 100 \% \\
 gap_2 &= \frac{z_l - z_{ip}}{z_{ip}} 100 \%
 \end{aligned} \tag{4.2}$$

donde  $gap_1$  es la proximidad de la solución óptima entera a la mejor cota lagrangiana,  $gap_2$  a la cota de la relajación lineal.

Se muestran los resultado para los 2 tamaños de instancias, podemos ver el número de variables de decisión y restricciones en las columnas 1 y 2 respectivamente, después se muestra el tiempo que le toma al algoritmo saber cuál es el índice de la mejor relajación lagrangiana ( $t_i$ ), en la cuarta columna se muestra el tiempo que tarda en dar el valor de la mejor cota al seguir iterando con el método de Benders

<b>n</b>	<b>m</b>	$t_i$	$t_c$	$t_p$	<b>ahorro( %)</b>	$gap_1$	$gap_2$
20	5	0.07	1.12	17.89	93.7	0.062	9.77
20	5	0.84	1.08	13.17	91.7	1.38	5.92
20	5	4.92	5.21	13.44	61.23	3.39	5.23
20	5	0.83	1.99	16.8	88.15	0.084	11.87
20	5	0.67	1.52	16.31	90.68	1.39	5.67
20	5	0.51	1.42	11.25	87.36	0.69	5.57
20	5	4.32	5.88	12.24	51.96	4.5	5.86
20	5	0.71	1.03	11.12	90.73	1.46	7.24
20	5	9.55	33.57	38.09	11.86	1.703	2.98
20	5	1.06	2.19	22.25	90.15	1.65	6.32

Tabla 4.7: Resultados para casos medianos GAP

( $t_c$ ), en la quinta muestra el tiempo que tomaría calcular todas las posibles cotas lagrangianas para después elegir la mejor ( $t_p$ ), después se muestra el ahorro promedio de tiempo entre  $t_p$  y  $t_c$  y en las últimas 3 columnas se muestran los resultados del  $gap_1$ ,  $gap_2$  y  $gap_3$ .

También se muestra la tabla con el resumen de los resultados para algunas otras instancias.

Esta tabla contiene el número de variables de decisión y restricciones, en la tercera columna aparece la media del tiempo que tarda el algoritmo en encontrar la mejor cota lagrangiana, después aparece la media del tiempo si calculamos todas las posibles cotas y al final nos muestra la media del ahorro promedio entre estos dos tiempos.

### 4.3.1 ANÁLISIS ESTADÍSTICO DE RESULTADOS GAP

Al igual que en la sección anterior se realizan las pruebas estadísticas necesarias para establecer algunas conclusiones sobre los datos que se obtuvieron en la

<b>n</b>	<b>m</b>	$m_c$	$m_p$	$m_a$
5	3	0.166	0.318	43.639
10	5	0.967	1.852	46.183
20	5	5.501	17.256	75.752

Tabla 4.8: Cálculo de medias para diferentes tamaños del problema GAP

experimentación computacional.

Los datos con los cuales se realizan estas pruebas con tomados de las tablas que se presentan en el apéndice de resultados, se cuenta con 60 instancias de diversos tamaños y lo que se busca probar es:

- Que el tiempo empleado por el algoritmo propuesto es menor al tiempo usado para calcular todas las cotas.
- Que la mejor cota lagrangiana domina a la cota lineal.

#### PRUEBA DE HIPÓTESIS PARA DIFERENCIA DE MEDIAS GAP

Primero se realizará la prueba para comparar el tiempo que tarda el algoritmo que se propone en este trabajo contra el tiempo de calcular cada relajación lagrangiana posible. Se utiliza el mismo estadístico de prueba que en las pruebas del MKP.

Criterios para rechazar  $H_0$ :

<b>Hipótesis nula</b>	<b>Hipótesis alternativa</b>	<b>Rechaza si:</b>
$H_0:\mu_1 = \mu_2$	$H_1:\mu_1 \neq \mu_2$	$ Z  \geq Z_{\alpha/2}$
$H_0:\mu_1 = \mu_2$	$H_1:\mu_1 > \mu_2$	$Z \geq Z_\alpha$
$H_0:\mu_1 = \mu_2$	$H_1:\mu_1 < \mu_2$	$ Z  \geq Z_\alpha$

Consideramos un  $\alpha = 0.05$ , es decir, un intervalo de confianza del 95% y tenemos:

$H_0 : \mu_p = \mu_c$  donde  $\mu_p$  es la media del tiempo de calcular todas las cotas y  $\mu_c$  es la media del tiempo de encontrar la mejor cota en base al algoritmo propuesto, y

$H_1 : \mu_p > \mu_c$

- $\bar{X}_p = 6.2165$  se calculan las respectivas medias muestrales.
- $\bar{X}_c = 2.2528$
- $\sigma_p^2 = 62.2415$
- $\sigma_c^2 = 25.2394$
- $n_1 = n_2 = 60$

Se calcula el estadístico de prueba  $Z$  como:

$$Z_0 = \frac{\bar{X}_p - \bar{X}_c}{\sqrt{\frac{\sigma_p^2}{n_1} + \frac{\sigma_c^2}{n_2}}}$$

$$Z_0 = 3.8916$$

$$Z_\alpha = Z_{0.05} = 1.65$$

Para comprobar nuestra hipótesis debemos tener  $Z > Z_\alpha$ , y entonces comparamos los resultados obtenidos y tenemos que  $3.8916 > 1.65$ .

Por lo tanto se rechaza  $H_0$ , por lo que aceptamos  $H_1$ , es decir, que la media del tiempo usado por el algoritmo es menor que la media del tiempo de calcular cada relajación lagrangiana para elegir la mejor después.

Ahora se quiere probar estadísticamente que la cota lagrangiana es mejor en un que la cota lineal con respecto a su cercanía con el valor óptimo del problema con un  $\alpha = 0.01$ .

Criterios para rechazar  $H_0$ :

Se tiene entonces,  $H_0 : \mu_d = 60\%$  donde  $\mu_d$  es la media de diferencias entre la cota lagrangiana y la lineal,  $H_1 : \mu_d > 60\%$ . Entonces:

Hipótesis nula	Hipótesis alternativa	Rechaza si:
$H_0: \mu_d = d$	$H_1: \mu_d \neq d$	$ Z  \geq Z_{\alpha/2}$
$H_0: \mu_d = d$	$H_1: \mu_d > d$	$Z \geq Z_{\alpha}$
$H_0: \mu_d = d$	$H_1: \mu_d < d$	$ Z  \leq -Z_{\alpha}$

- $\bar{d} = 8.7125$
- $\mu_d = 30\% = 5.2275$
- $\sigma_d = 5.8194$
- $n = 60$

El estadístico de prueba que se usa es:

$$Z_0 = \frac{\bar{d} - \mu_d}{\sigma_d} \sqrt{n}$$

$$Z_0 = 4.638$$

Para rechazar la hipótesis nula necesitamos que  $Z_0 > Z_{\alpha}$ , entonces comparamos los resultados que se obtuvieron.

$$Z_{\alpha} = Z_{0.01} = 3.09$$

Entonces  $4.638 > 3.09$ , por lo que rechazamos la hipótesis nula, por lo tanto hemos comprobado de forma estadística que el promedio de mejoría de la mejor cota lagrangiana sobre la cota lineal con respecto a la cercanía de ellas al valor objetivo óptimo es mayor a un 60 %.

## CAPÍTULO 5

# CONCLUSIONES

---

En el presente capítulo se hablará sobre las conclusiones a las que se llegó luego de la experimentación computacional y el análisis de los resultados obtenidos del algoritmo propuesto.

### 5.1 CONTRIBUCIONES

Se implementó el algoritmo propuesto al inicio del presente trabajo y se realizaron diversos experimentos computacionales para después analizar los resultados de los mismos de forma estadística.

Como se puede ver en los resultados que se obtuvieron, nuestra metodología ofrece una forma eficiente de encontrar la mejor relajación lagrangiana, en el sentido de cuál provee la mejor cota, sin tener que obtener, explícitamente, los resultados de cada relajación y después elegir la mejor, método que ha sido empleado hasta ahora.

### 5.2 TRABAJO A FUTURO

Trabajamos con algunas estructuras especiales de problemas de optimización a gran escala, se puede también, trabajar con distintas estructuras de los problemas.

Además, el cálculo de cada cota lagrangiana es un problema tipo minmax (maxmin). Queremos buscar el mejor valor entre varios valores de minmax. Esta clase



---

de problemas se encuentran no solo en la problemática de cotas lagrangianas sino también en la optimización con incertidumbre, optimización con varios escenarios, optimización paramétrica, etc. Si el número de subproblemas minmax es grande, la estrategia exhaustiva de calcular todos los valores de minmax y después elegir el mejor no es eficiente.

Como trabajo futuro se propone aplicar la metodología desarrollada aquí para los problemas mencionados anteriormente.

APÉNDICE A

RESULTADOS

---

<b>n</b>	<b>m</b>	$t_i$	$t_c$	$t_p$	<b>ahorro(%)</b>	$gap_1$	$gap_2$
5	3	0.04	0.07	0.14	50	8.36	29.13
5	3	0.06	0.06	0.08	25	7.45	33.11
5	3	0.06	0.07	0.1	30	0.95	36.97
5	3	0.14	0.15	0.16	6.25	16.05	32.65
5	3	0.1	0.12	0.14	14.29	16.45	37.6
5	3	0.09	0.11	0.12	8.33	0.57	27.51
5	3	0.07	0.07	0.1	30	17.51	34.48
5	3	0.11	0.12	0.13	7.69	26.62	42.19
5	3	0.05	0.06	0.11	45.45	41.19	82.93
5	3	0.04	0.06	0.08	25	3.32	22.15
5	3	0.3	0.35	0.47	25.531	11.03	33.98
5	3	0.16	0.21	0.3	30	0.6	28.33
5	3	0.53	0.57	0.67	14.925	13.75	29.34
5	3	0.42	0.48	0.55	12.727	14.1	28.62
5	3	0.5	0.55	0.63	12.698	17.51	37.51

<b>n</b>	<b>m</b>	$t_i$	$t_c$	$t_p$	<b>ahorro( %)</b>	$gap_1$	$gap_2$
5	3	0.26	0.31	0.36	13.888	20.68	43.2
5	3	0.21	0.26	0.47	44.680	22.7	46.18
5	3	0.32	0.37	0.58	36.206	22.79	47.2
5	3	0.32	0.38	0.43	11.627	1.66	26.49
5	3	0.36	0.42	0.53	20.754	5.83	26.006
5	3	0.3	0.3	0.34	11.764	7.29	30.87
5	3	0.29	0.35	0.39	10.256	11.99	42.34
5	3	0.5	0.54	0.61	11.475	15.4	35.45
5	3	0.47	0.51	0.63	19.047	14.84	40.05
5	3	0.16	0.22	0.36	38.888	0.47	27.93
5	3	0.29	0.34	0.38	10.526	21.98	36.32
5	3	0.3	0.35	0.39	10.256	23.79	38.43
5	3	0.28	0.36	0.64	43.75	43.33	92.66
5	3	0.25	0.3	0.36	16.666	2.69	20.98
5	3	0.3	0.37	0.5	26	11.29	30.89

Tabla A.1: Resultados MKP tamaño 5x3

<b>n</b>	<b>m</b>	$t_i$	$t_c$	$t_p$	<b>ahorro( %)</b>	$gap_1$	$gap_2$
10	5	0.21	0.21	0.39	46	9.74	25.22
10	5	0.17	0.17	0.29	41	5.27	11.49
10	5	0.29	0.31	0.38	19	10.88	15.99
10	5	0.19	0.19	0.38	50	16.33	25.88
10	5	0.14	0.15	0.24	38	5.53	11.6
10	5	0.3	0.3	0.37	18.91	19.76	28.61
10	5	0.18	0.2	0.32	37.5	10.53	16.43
10	5	0.3	0.31	0.41	24.4	13.45	18.77
10	5	0.3	0.34	0.4	15	6.86	16.44
10	5	0.22	0.23	0.33	30.3	4.94	14.12
10	5	1.35	1.38	1.87	26.203	10.51	27.39
10	5	1.57	1.6	2.6	38.461	9.88	31.49
10	5	1.37	1.37	2.35	41.702	12.39	15.41
10	5	1.22	1.28	2.3	44.347	10.83	32.19
10	5	1.02	1.17	2.74	57.299	16.73	20.56

<b>n</b>	<b>m</b>	$t_i$	$t_c$	$t_p$	<b>ahorro( %)</b>	$gap_1$	$gap_2$
10	5	1.05	1.1	1.39	20.863	17.6	8,32
10	5	1.38	1.54	1.83	15.846	19.13	12.37
10	5	0.89	0.98	1.88	47.872	8.9	26.57
10	5	1.2	1.25	1.6	21.875	13.12	21.19
10	5	1.09	1.12	1.96	42.857	15.14	31.95
10	5	1.85	1.89	2.19	13.698	12.52	33.23
10	5	1.88	2.04	2.53	19.367	17.63	20.47
10	5	1.69	1.96	2.92	32.876	11.18	19.17
10	5	1.8	1.84	2.41	23.651	19.9	7.64
10	5	1.01	1.05	1.44	27.083	22.78	14.83
10	5	1.14	1.23	2.06	40.291	14.93	22.09
10	5	1.63	1.7	2.16	21.296	15.87	12.2
10	5	1.45	1.71	3.18	46.226	16.59	27.55
10	5	0.85	1.09	1.8	39.444	21.82	29.57
10	5	1.74	1.77	2.39	25.941	8.86	20.25

Tabla A.2: Resultados MKP tamaño 10x5

<b>n</b>	<b>m</b>	$t_i$	$t_c$	$t_p$	<b>ahorro( %)</b>	$gap_1$	$gap_2$
20	10	1.82	1.86	2.95	37	13.72	19.18
20	10	1.44	1.46	2.61	45	22.84	30.09
20	10	2.39	2.42	2.47	7	9.98	19.86
20	10	1.96	2.02	3.61	44	11.08	24.19
20	10	2.2	2.23	3.25	32	17.14	21.24
20	10	1.24	1.33	2.19	39.27	19.28	26.36
20	10	1.36	1.54	3.41	54.83	18.19	24.79
20	10	1.37	1.45	2.64	45.08	24.2	32.33
20	10	1.18	1.4	2.27	38.33	6.74	12.28
20	10	1.08	1.2	2.74	56.2	6.43	11.85
20	10	8.53	8.82	15.73	43.928	19.05	14.72
20	10	11.42	11.69	17.23	32.153	21.3	12.32
20	10	7.5	8.19	14.76	44.512	7.73	17.49
20	10	7.08	7.08	14.18	50.070	21.86	15.74
20	10	5.67	6.07	12.32	50.730	9.81	23.17

<b>n</b>	<b>m</b>	$t_i$	$t_c$	$t_p$	<b>ahorro( %)</b>	$gap_1$	$gap_2$
20	10	6.85	7.12	14.63	51.332	4.23	24.74
20	10	9.8	9.97	13.22	24.583	6.92	23.28
20	10	8.93	9.33	16.49	43.420	12.38	12.4
20	10	7.73	7.84	14.97	47.628	11.04	19.55
20	10	8.28	8.32	13.16	36.778	16.47	19.65
20	10	6.4	7.22	14.18	49.083	22.72	18.6
20	10	6.14	6.56	14.44	54.570	13.32	23.99
20	10	10.12	10.34	15.98	35.294	10.09	15.9
20	10	6.73	7.17	13.58	47.201	3.5	26.73
20	10	7.71	8.04	12.79	37.138	7.31	30.02
20	10	8.12	8.41	13.55	37.933	14.96	19.78
20	10	8.64	9.17	13.63	32.721	4.06	22.32
20	10	7.73	7.98	11.77	32.200	12.96	21.95
20	10	8.12	8.6	16.52	47.941	13.94	2.57
20	10	7.79	7.79	11.02	29.310	12.93	11.95

Tabla A.3: Resultados MKP tamaño 20x10

<b>n</b>	<b>m</b>	$t_i$	$t_c$	$t_p$	<b>ahorro( %)</b>	$gap_1$	$gap_2$
40	20	11.14	11.55	21.27	46	6.75	8.54
40	20	11.32	11.4	19.68	43	9.89	11.59
40	20	8.47	8.75	14.54	40	8.68	10.77
40	20	11.96	12.09	16.39	26.24	7	8.41
40	20	9.25	9.58	18.2	47.36	0.935	1.23
40	20	10.99	11.26	17.27	34.8	9.43	10.95
40	20	48.98	49.04	88.17	44.38	8.64	10.5
40	20	32.44	34.63	71.41	51.5	8.6	11.1
40	20	41.56	41.97	58.21	27.9	10.38	12.62
40	20	29.24	29.97	57.05	47.48	3.3	4.52
40	20	41.02	41.83	75.33	44.470	5.64	7.19
40	20	34.29	34.63	60.04	42.321	9.03	11.55
40	20	24.61	25.16	64.86	61.208	7.56	10.08
40	20	22.84	23.47	38.7	39.354	4.07	5.14
40	20	35.88	36.12	56.43	35.991	9.78	11.74



<b>n</b>	<b>m</b>	$t_i$	$t_c$	$t_p$	<b>ahorro( %)</b>	$gap_1$	$gap_2$
40	20	29.97	31.1	68.41	54.538	7.35	10.14
40	20	49.72	50.71	80.39	36.920	11.8	13.97
40	20	44.8	44.94	65.68	31.577	11.05	12.46
40	20	33.96	34.35	59.92	42.673	5.45	7.92
40	20	60.31	60.59	99.63	39.184	6.32	7.58
40	20	54.26	56.24	91.7	38.669	7.11	8.14
40	20	45.93	46.14	78.97	41.572	9.56	11.09
40	20	50.51	50.59	74.22	31.837	9.92	10.95
40	20	39.89	39.93	57.49	30.544	7.43	8.51
40	20	44.06	47.04	79.81	41.060	8.96	11.67
40	20	25.72	26.45	56.51	53.194	7.58	9.91
40	20	42.61	44.02	81.22	45.801	8.61	10.31
40	20	28.17	29.13	56.35	48.305	11.38	13.9
40	20	43.39	44.03	73.22	39.866	11.08	13.3
40	20	28.73	30.39	59.97	49.324	6.76	9.13

Tabla A.4: Resultados MKP tamaño 40x20

<b>n</b>	<b>m</b>	$t_i$	$t_c$	$t_p$	<b>ahorro( %)</b>	$gap_1$	$gap_2$
60	30	10,96	11,22	22,95	51	4,12	5,4
60	30	42,54	42,93	66,62	36	11,72	13,4
60	30	24,77	25,31	99,8	74	10,5	12,41
60	30	16,32	16,75	33,32	49,73	5,44	6,94
60	30	40,49	41,37	162,85	74,59	12,36	14,3
60	30	39,68	40,78	108,18	62,3	8,99	10,96
60	30	25,79	25,79	30,88	16,5	4,9	5,45
60	30	51,19	51,82	100,15	48,26	10,92	12,23
60	30	14,96	15,23	28,9	47,3	1,86	4,9
60	30	39,01	40,34	111,11	63,69	11,37	13,3
60	30	113,18	114,12	222,09	48,615	7,07	8,47
60	30	47,51	47,9	88,83	46,076	2,66	3,52
60	30	120,88	123,26	277,29	55,548	8,92	10,68
60	30	81,62	81,86	154,93	47,16	5,08	5,6
60	30	55,14	55,98	143,02	60,858	3,82	4,53
60	30	196,22	198,64	276,78	28,231	9,15	10,45
60	30	17,21	18,09	64,39	71,905	1,24	2,27
60	30	128,39	128,63	197,75	34,953	8,26	9,38
60	30	123,68	123,68	211,8	41,605	10,19	11,76
60	30	121,94	122,19	194,21	37,083	5,99	7,08

Tabla A.5: Resultados MKP tamaño 60x30

<b>n</b>	<b>m</b>	$t_i$	$t_c$	$t_p$	<b>ahorro( %)</b>	$gap_1$	$gap_2$
100	50	288,22	290,14	1721,36	83	10,84	11,93
100	50	440,62	450,25	2580,89	82	8,49	9,23
100	50	553,75	562,37	2621,6	78,55	7,49	9,16
100	50	487,66	492,14	2511,54	80,41	8,64	11,04
100	50	627,43	632,52	1259,08	49,76	9,6	10,54

Tabla A.6: Resultados MKP tamaño 100x50

<b>n</b>	<b>m</b>	$t_i$	$t_c$	$t_p$	<b>ahorro( %)</b>	$gap_1$	$gap_2$
5	3	0.07	0.09	0.13	30.77	0.24	21.81
5	3	0.09	0.09	0.21	57.14	0.67	16.42
5	3	0.08	0.12	0.19	36.84	5.04	15.14
5	3	0.08	0.12	0.16	25	0.2	20.5
5	3	0.05	0.06	0.12	50	0.47	21.32
5	3	0.19	0.25	0.33	24.24	0.24	22.65
5	3	0.25	0.27	0.62	56.45	0.88	7.68
5	3	0.22	0.24	0.39	38.46	0.076	13.12
5	3	0.19	0.21	0.57	63.15	0.024	11.97
5	3	0.18	0.21	0.46	54.34	0.088	15.13
5	3	0.2	0.23	0.58	60.344	0.03	15.26
5	3	0.13	0.31	0.601	48.419	0.33	16.48
5	3	0.09	0.12	0.55	78.181	0.08	20.24
5	3	0.24	0.34	0.67	49.253	0.69	16.04
5	3	0.14	0.34	0.71	52.112	0.38	20.56
5	3	0.28	0.35	0.57	38.596	1.24	16.9
5	3	0.09	0.13	0.68	80.882	0.08	9.12
5	3	0.27	0.44	0.7	37.142	0.72	12.2
5	3	0.18	0.31	0.69	55.072	0.29	13.6
5	3	0.18	0.25	0.32	21.875	0.16	15.02

Tabla A.7: Resultados GAP tamaño 5x3

<b>n</b>	<b>m</b>	$t_i$	$t_c$	$t_p$	<b>ahorro(%)</b>	$gap_1$	$gap_2$
10	5	0.03	0.3	0.52	42.3	3.29	9.32
10	5	0.36	0.57	0.75	24	0.19	6.95
10	5	0.18	0.44	0.72	38.9	3.26	8.59
10	5	0.18	0.38	0.53	28.3	2.92	7.56
10	5	0.06	0.22	0.65	66.15	0.68	7.95
10	5	1.01	3.33	4.78	30.33	3.31	8.71
10	5	0.31	0.84	3.34	74.85	0.25	12.72
10	5	0.39	1.24	2.58	51.93	3.16	8.47
10	5	0.12	0.38	2.08	81.73	0.59	7.22
10	5	0.75	1.97	2.57	23.34	4.8	8.39
10	5	0.77	2.87	4.03	28.784	3.66	6.83
10	5	0.29	0.61	2.23	72.645	2.76	8.49
10	5	0.37	1.19	2.59	54.054	2.92	7.97
10	5	0.23	0.56	2.48	77.419	0.96	8.24
10	5	0.4	1.76	3.14	43.949	0.98	8.02
10	5	1.01	3.12	4.23	26.241	5.94	11.77
10	5	0.49	3.12	4.15	24.819	3.36	7.6
10	5	0.32	0.36	1.49	75.838	1.05	8.99
10	5	0.85	2.16	3.22	32.919	4.16	9.78
10	5	1.16	3.14	4.36	27.981	1.57	7.4

Tabla A.8: Resultados GAP tamaño 10x5

<b>n</b>	<b>m</b>	$t_i$	$t_c$	$t_p$	<b>ahorro( %)</b>	$gap_1$	$gap_2$
20	5	0.07	1.12	17.89	93.7	0.062	9.77
20	5	0.84	1.08	13.17	91.7	1.38	5.92
20	5	4.92	5.21	13.44	61.23	3.39	5.23
20	5	0.83	1.99	16.8	88.15	0.084	11.87
20	5	0.67	1.52	16.31	90.68	1.39	5.67
20	5	0.51	1.42	11.25	87.36	0.69	5.57
20	5	4.32	5.88	12.24	51.96	4.5	5.86
20	5	0.71	1.03	11.12	90.73	1.46	7.24
20	5	9.55	33.57	38.09	11.86	1.703	2.98
20	5	1.06	2.19	22.25	90.15	1.65	6.32
20	5	2.08	5.28	6.28	15.923	3.02	4.96
20	5	2.74	9.15	10.82	15.434	2.12	4.5
20	5	5.16	19.23	20.91	8.0344	4.57	6.6
20	5	3.08	4.17	25.01	83.326	2.86	6.19
20	5	0.78	1.67	16.41	89.823	1.62	6.5
20	5	0.82	1.66	11.81	85.944	0.93	6.16
20	5	0.53	1.14	11.05	89.683	1.14	6.1
20	5	0.33	0.38	13.23	97.127	0.06	10.24
20	5	0.35	0.51	14.85	96.565	0.09	5.9
20	5	3.19	3.93	10.37	62.102	2.38	3.88

Tabla A.9: Resultados GAP tamaño 20x5

## APÉNDICE B

# GAMS

---

## B.1 GENERALIDADES

### B.1.1 GENERALIDADES DE GAMS

Alrededor de la década de los cincuentas ha ocurrido un fulminante desarrollo de algoritmos y códigos computacionales para la solución y análisis de problemas de programación matemática. En principios de los ochentas los sistemas de modelado tuvieron una parte importante de su desarrollo. Uno de los sistemas de modelado que se desarrollaron primeramente fue el General Algebraic Modeling System (GAMS), es un sistema de alto nivel utilizado para la modelación de problemas de programación matemática y optimización, que esta compuesto por un lenguaje compilador integrado a un grupo de solvers de alto rendimiento. Principalmente fue diseñado para:

- La optimización de problemas: lineales, no lineales, enteros y cualquier combinación de ellos, entre otros.
- Brindar un lenguaje basado primordialmente en álgebra para representar de forma compacta modelos grandes y complejos.
- Admitir de forma rápida y segura cambios en las especificaciones del modelo.
- Calcular sentencias de relaciones algebraicas omitiendo ambigüedades.

- 
- Facilitar la verificación de que el modelo creado sea correcto, tratando de propagar la expansión de contextos, en un ambiente que este basado en subíndices, cierta expansibilidad, dar acceso al usuario de empezar con un conjunto pequeño de datos.
  - Permitir la expansión de los nombres de las variables, describir la función de ecuaciones, índices, definición de datos como comentarios siendo inherentemente auto documentado. GAMS esta diseñado para que la estructura del modelo, suposiciones, y cualquier proceso de cálculo usado en el reporte escrito sea documentado como un archivo auto contenido.
  - Facilitar la interfaz de interacción con otros nuevos y mejores solvers, siempre y cuando sean independientes, para que sean utilizados en la solución de cualquier problema dado.
  - Incluir en la automatización del proceso de modelado lo siguiente:
    - acceso a cálculo de datos;
    - verificar la estructura correcta de las sentencias algebraicas del modelo;
    - evitar la formulación de fallos obvios;
    - interactuar con un solver;
    - salvar y presentar una base avanzada cuando se generan soluciones relacionadas;
    - tener uso de la solución para reportes escritos.
  - Tener acceso a la formulación de un modelo entre sistemas de cómputo es decir que pueda ser usada en una amplia gama de ordenadores, también importar/exportar datos de otros paquetes computacionales.
  - Cambiar entre opciones de un solver y otro.
  - Ser utilizados por grupos de varias disciplinas.
  - Contar con una librería de modelos para dar ejemplos de ayuda a los usuarios.

## B.1.2 GENERALIDADES DE CPLEX

### PROGRAMACIÓN LINEAL

La solución que realiza Cplex de problemas lineales la lleva a cabo utilizando alternadamente varios algoritmos. Por ejemplo el algoritmo dual simplex más avanzado de Cplex es utilizado para resolver la mayoría de problemas lineales debido a que son resueltos mejor con este, así mismo otros ciertos tipos de problemas se resuelven mejor usando el algoritmo primal simplex, el optimizador para redes, el algoritmo de barrera, o el algoritmo de exploración. Existe la opción paralela si esta activada, resuelve diversos algoritmos en paralelo y da la respuesta del primero que termine.

A pesar de que para la memoria es muy intensivo resolver problemas de programación lineal Cplex la maneja eficientemente. La escasez de la memoria física es un problema común al momento de resolver problemas de dimensiones grandes; en caso de ser la memoria limitada, Cplex automáticamente hará los ajustes necesarios para no impactar negativamente el funcionamiento.

El diseño de Cplex está hecho para resolver bajo varios escenarios por defecto la mayoría de los problemas lineales. Tales escenarios brindan la mejor velocidad y confiabilidad de la optimización del problema. Sin embargo, en varias ocasiones hay razones para modificar las opciones para la mejora del funcionamiento, evitar dificultades numéricas y tener el control de la duración de la optimización, o las opciones de salida.

Existen diferencias entre el tiempo de solución de los problemas resueltos con el algoritmo primal simplex o con el algoritmo dual simplex, es poco común que un problema cuente con un funcionamiento numérico pobre en ambos algoritmos. Consecuentemente es considerable contemplar el algoritmo primal simplex si existen problemas numéricos cuando se usa el dual simplex.

Cplex cuenta con un algoritmo para modelos de red muy eficiente. Las carac-



terísticas que poseen las restricciones son las siguientes:

- los coeficientes distintos a cero son  $+1$  ó  $-1$ ;
- cada una de las columnas de las restricciones tiene exactamente 2 entradas distintas a cero, una con un coeficiente  $+1$  y la otra con  $-1$ .

Otra de las cualidades de Cplex es que puede realizar extracciones automáticamente de redes que no se adhieren a las convenciones antes mencionadas y podrían ser transformados para contar con esas características.

Una alternativa al método simplex es el algoritmo de barrera utilizado para solucionar programas lineales. Éste consiste en emplear un algoritmo de barrera primal-dual logarítmico el cual genera una sucesión de soluciones primales y duales rigurosamente positiva, usar este algoritmo resulta ventajoso para problemas grandes y dispersos.

Cplex cuenta con un algoritmo de exploración que podría ser eficaz en problemas con que cuentan con muchas más variables que ecuaciones. Lo que hace tal algoritmo es solucionar una sucesión de subproblemas lineales que a partir de los resultados obtenidos de un subproblema se seleccionan las columnas del modelo original que se incluirán en el siguiente subproblema.

Otra de las funciones que maneja GAMS/Cplex es el acceso al *Cplex Infeasibility Finder*, el cual es un buscador de infactibilidad que toma un problema lineal infactible y origina un conjunto de restricciones inconsistentes irreducibles (IIS), el cuál es un conjunto de restricciones y de cotas para las variables que lo hacen infactible, eliminado cualquier miembro del sistema para convertirlo en factible. En GAMS/Cplex el reporte del IIS está dentro en términos de las ecuaciones de GAMS y de los nombres de las variables, el reporte del IIS esta incluido en una sección del registro de la solución.

## PROGRAMACIÓN ENTERA MIXTA

Para solucionar problemas de tipo entero-puro y mixto es necesario usar métodos que realicen más cálculos matemáticos que los empleados para problemas lineales puros que cuenten con las mismas condiciones. El tiempo de solución para resolver un problema de programación entera de dimensiones relativamente pequeño es enorme.

Un algoritmo utilizado por Cplex para problemas con variables enteras, es el de ramificar y acotar, el cual consiste en solucionar una serie de subproblemas. Un problema entero mixto es muy agotador computacionalmente por la gran cantidad de subproblemas que genera y la demasiada memoria física que requiere para solucionarse.

### B.1.3 GAMS/CPLEX

Un solver de GAMS que brinda acceso a los usuarios de combinar el alto nivel de modelación de GAMS y el poder de los optimizadores de Cplex es el GAMS/Cplex. El diseño con el cuentan los ordenadores de Cplex es para solucionar problemas grandes y complicados en un corto tiempo y de ser posible una mínima mediación del usuario. Para los problemas de programación lineal, cuadrática, problemas con restricciones y enteros mixtos, da acceso (con la licencia apropiada) a los algoritmos de solución de Cplex. Además GAMS/Cplex cuenta con varias opciones de solución disponibles que para la mayoría de las opciones de problemas específicos calculan y fijan automáticamente los mejores valores GAMS/Cplex cuenta con todas las opciones de Cplex.

Si se desea tener mayor información sobre GAMS, Cplex y GAMS/Cplex ver [65] y por la red en [www.gams.com](http://www.gams.com) y [www.cplex.com](http://www.cplex.com).

## APÉNDICE C

# PROGRAMACIÓN LINEAL

---

La programación lineal es una rama de la programación matemática que estudia la optimización (maximización o minimización) de una función lineal que debe satisfacer un conjunto de restricciones lineales también de desigualdad o igualdad.

## C.1 BREVE HISTORIA

La programación lineal fue concebida por George B. Dantzig quien la desarrolló alrededor de 1947 cuando fungía como consejero de la Fuerza Aérea de los Estados Unidos de América para dar solución a un programa de entrenamiento y abastecimiento logístico durante la segunda guerra mundial. Dantzig no sólo resolvió dicho problema, sino que formuló el *problema general de programación lineal* y desarrolló el *método simplex* de solución, sin embargo su trabajo no estuvo disponible hasta 1949, año de su publicación.

El nombre de *programa* no está asociado de manera alguna con un programa de ordenador, pues de hecho para la época esto hubiera sido imposible. Dado que en la fuerza aérea los planes y proyectos a implementar son llamados programas, Dantzig solía referirse a los problemas de programación lineal como “programa en una estructura lineal”, y no fue hasta 1948 que el matemático Koopmans acuñó el término *programación lineal*.

Diversos problemas del tipo de los de programación lineal fueron formulados y

resueltos antes del primer trabajo de Dantzig. Von Neumann proporcionó su modelo de equilibrio en 1937. En 1939 Kantorovich planteó y resolvió un problema de organización y planificación. En 1941 Hitchcock formuló y resolvió el problema de transporte que fue independientemente resultado por Koopmans en 1947. En 1942 Kantorovich también formuló el problema de transporte, pero no lo resolvió. En 1944 Von Neumann y Morgenstern formularon la teoría de juegos. Stigler trabajó el problema de la dieta en 1945.

Después de la publicación del método simplex el campo de la programación lineal se ha visto ampliamente enriquecido. La programación lineal ha sido favorablemente aceptada por su inherente capacidad para modelar una gran diversidad de problemas complejos y de gran importancia en física, geometría, administración e ingeniería entre otras. Así mismo, gracias al desarrollo y cada vez mayor capacidad de los ordenadores modernos, el método simplex ha sido la herramienta de solución de los problemas de programación lineal por excelencia.

## C.2 EL PROBLEMA GENERAL DE LA PROGRAMACIÓN LINEAL

El siguiente es un problema de programación lineal en su forma general.

$$\begin{aligned}
 \text{mín} \quad & c_1x_1 + c_2x_2 + \dots + c_nx_n \\
 \text{s. a:} \quad & a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \geq b_1 \\
 & a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \geq b_2 \\
 & \vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \\
 & a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \geq b_m \\
 & x_1, x_2, \dots, x_n \geq 0
 \end{aligned}$$

Aquí  $c_1x_1 + c_2x_2 + \dots + c_nx_n$  es la *función objetivo* que debe minimizarse, los parámetros  $c_j$  son los *coeficientes de costo*, las cantidades  $x_j$  son las *variables de*

*decisión* que deben determinarse. La desigualdad  $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \geq b_i$  es la  $i$ -ésima *restricción*, los parámetros  $a_{ij}$  se denominan *coeficientes tecnológicos*, los parámetros  $b_i$  reciben el muy general nombre de *valor del lado derecho* y representa los requerimientos que deben satisfacerse. Por último,  $x_1, x_2, \dots, x_n \geq 0$  son las *restricciones de no negatividad*.

Un conjunto de valores para las variables  $x_1, x_2, \dots, x_n$  que satisfacen todas las restricciones se denomina *punto factible*. Al conjunto de todos los puntos factibles se le llama *región factible*.

### C.2.1 HIPÓTESIS DE LA PROGRAMACIÓN LINEAL

Para que la representación de un problema de optimización como uno de programación lineal sea válida se requiere cumplir con al menos las siguientes cuatro hipótesis:

1. *Proporcionalidad*. El costo de una variable es proporcional al valor que tome así como la contribución en cada restricción. Esto es, dada una variable  $x_j$ , su contribución al costo total es  $c_j x_j$  y a la  $i$ -ésima restricción es  $a_{ij} x_j$ .

2. *Aditividad*. El costo total es la suma de los costos individuales y la contribución total en cada restricción es la suma de las contribuciones de cada variable.

3. *Divisibilidad*. Las variables de decisión se pueden dividir en cualesquiera valores fraccionales, permitiendo con ello valores no enteros para las variables.

4. *Determinística*. Todos los coeficientes  $c_j$ ,  $a_{ij}$  y  $b_i$  están bien determinados. Esto es, se asume que no hay componentes estocásticas o que tales componentes pueden aproximarse bien mediante tales coeficientes.

### C.2.2 MANIPULACIONES DEL PROBLEMA GENERAL

Como ya se ha dicho, un programa lineal es un problema de maximizar o minimizar una función lineal bajo restricciones lineales de desigualdad o igualdad. Todo problema lineal puede transformarse de una forma a otra equivalente mediante manipulaciones simples.

**Desigualdades e igualdades:** Una restricción de desigualdad  $\sum_j a_{ij}x_j \leq b_i$  puede transformarse fácilmente en una de igualdad adicionando una *variable de holgura*  $s_i \geq 0$  obteniendo  $\sum_j a_{ij}x_j + s_i = b_i$ . Similarmente, la restricción  $\sum_j a_{ij}x_j \geq b_i$  se transforma en  $\sum_j a_{ij}x_j - s_i = b_i$  sustrayendo la *variable de exceso*  $s_i \geq 0$ . Por otra parte una restricción de igualdad  $\sum_j a_{ij}x_j = b_i$  siempre puede transformarse en las dos desigualdades  $\sum_j a_{ij}x_j \geq b_i$  y  $\sum_j a_{ij}x_j \leq b_i$ .

**No negatividad en las variables:** El método simplex ha sido diseñado para resolver problemas cuyas variables sean no negativas. Si una variable se requiere no positiva, i.e.  $x_j \leq 0$ , es posible la siguiente sustitución  $x'_j = -x_j \geq 0$ . Si una variable  $x_j$  no está restringida en signo, es posible reemplazarla por  $x'_j - x''_j$ , donde  $x'_j, x''_j \geq 0$ . Si alguna variable está acotada inferiormente,  $x_j \geq l_j$ , entonces la variable  $x'_j = x_j - l_j$  es claramente no negativa. También si una variable está acotada superiormente  $x_j \leq u_j$  por una cantidad no positiva, i.e.  $u_j \leq 0$ , entonces la sustitución  $x'_j = u_j - x_j$  produce una variable no negativa.

**Maximización y minimización:** Todo problema de maximización puede convertirse en uno de minimización y viceversa. A saber, es claro que

$$\text{máx } \sum_j c_j x_j = - \text{mín } \sum_j -c_j x_j.$$

Así, todo problema de maximización (minimización) puede transformarse en uno de minimización (maximización) multiplicando por  $-1$  los coeficientes de la función objetivo. Obsérvese que al completar la optimización el valor del problema original es el simétrico aditivo.

## C.3 REPRESENTACIÓN MATRICIAL DE UN PROBLEMA LINEAL

Todo problema de programación lineal puede escribirse en forma matricial. Por ejemplo el problema

$$\begin{aligned} \text{máx} \quad & \sum_j c_j x_j \\ \text{s. a:} \quad & \sum_j a_{ij} x_j = b_i \quad i = 1, \dots, m \\ & x_j \geq 0 \quad j = 1, \dots, n \end{aligned}$$

puede escribirse matricialmente declarando las siguientes matrices

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}, \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}, A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

con lo cual el problema puede quedar expresado mediante

$$\begin{aligned} \text{máx} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s. a:} \quad & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

### C.3.1 FORMAS CANÓNICA Y ESTÁNDAR

Como ya se ha visto, un problema de programación lineal puede expresarse en diversas maneras haciendo alguna manipulación. Existen dos formas particularmente útiles llamadas forma canónica y forma estándar.

Se dice que un problema de programación lineal se encuentra en forma estándar si todas las restricciones son de igualdad y todas las variables son no negativas, i.e. si es de la forma

$$\begin{aligned} & \text{máx } \mathbf{c}^T \mathbf{x} \\ & \text{s. a: } \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

Un problema está en forma canónica si todas sus variables son no negativas y todas las restricciones son de tipo  $<$  para un problema de maximizar o todas son del tipo  $>$  si es un problema de minimizar. Esto es, si es de cualquiera de las formas

$$\begin{array}{ll} \text{máx } \mathbf{c}^T \mathbf{x} & \text{mín } \mathbf{c}^T \mathbf{x} \\ \text{s. a: } \mathbf{Ax} \leq \mathbf{b} & \text{s. a: } \mathbf{Ax} \geq \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} & \mathbf{x} \geq \mathbf{0} \end{array}$$

Como antes,  $A$  es una matriz  $n \times m$ ,  $\mathbf{c}$  y  $\mathbf{x}$  son un vector  $n$ -dimensional y  $\mathbf{b}$  es un vector no negativo  $m$ -dimensional.

Es importante mencionar que el método simplex está diseñado para resolver problemas sólo una vez que han sido expresados en su forma estándar.



## APÉNDICE D

# PROGRAMACIÓN ENTERA

---

### D.1 BREVE DESCRIPCIÓN

La programación entera es la rama de la programación matemática que trata de la optimización de problemas cuyas variables deben ser discretas o enteras. Por ejemplo en muchos modelos las variables deben tomar un valor de 0 ó 1 y representan una decisión de *Si* o *No* que debe ser tomada.

Una gran variedad de los problemas de la práctica pueden ser formulados y resueltos mediante programación entera. Desafortunadamente los métodos actuales de solución, aunque son suficientemente eficientes, no suelen ser tan rápidos como los que existen para programación lineal.

### D.2 CLASIFICACIÓN Y FORMULACIONES

**Programación lineal entera:** El siguiente es un problema de programación lineal entera. Obsérvese que todas las variables deben tomar valores enteros.

$$\begin{aligned} \text{máx} \quad & \mathbf{c}^T \mathbf{x} \\ \text{s. a:} \quad & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \\ & \mathbf{x} \in \mathbb{Z} \end{aligned}$$

donde  $A$  es una matriz  $m \times n$ ,  $\mathbf{c}$  e  $\mathbf{x}$  son vectores columna  $n$ -dimensionales, y  $\mathbf{b}$  un vector columna  $m$ -dimensional.

**Programación lineal entera mixta:** En ocasiones no es necesario que todas las variables tomen valores enteros, sino sólo un grupo de ellas. El siguiente es un problema entero mixto.

$$\begin{aligned} \text{máx} \quad & \mathbf{c}^\top \mathbf{x} + \mathbf{f}^\top \mathbf{y} \\ \text{s. a:} \quad & A\mathbf{x} + D\mathbf{y} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \mathbf{y} \geq \mathbf{0} \\ & \mathbf{x} \in \mathbb{R}, \mathbf{y} \in \mathbb{Z} \end{aligned}$$

donde  $A$ ,  $\mathbf{c}$ ,  $\mathbf{x}$  e  $\mathbf{b}$  son como antes;  $D$  es una matriz  $m \times p$ ;  $\mathbf{f}$  e  $\mathbf{y}$  son vectores columna  $p$ -dimensionales.

**Programación entera binaria:** Si todos los valores de las variables están restringidas a tomar valores en el conjunto  $\{0, 1\}$  se dice que tenemos un problema de programación entera binaria o simplemente un problema binario.

$$\begin{aligned} \text{máx} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s. a:} \quad & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \\ & \mathbf{x} \in \mathbb{B} \end{aligned}$$

donde  $A$ ,  $\mathbf{c}$ ,  $\mathbf{x}$  e  $\mathbf{b}$  son como ya se han definido y  $\mathbb{B} = \{1, 0\}$ .

**Optimización combinatoria:** Para finalizar, existe un tipo de problemas usualmente más complejo. Sea  $N = \{1, \dots, n\}$ ,  $c_j \in \mathbb{R}$ , para cada  $j \in N$ , y un conjunto de subconjuntos factibles de  $N$ . Un problema de optimización combinatoria es

$$\min_{S \subseteq N} \sum_{j \in S} c_j.$$

## D.3 ALGORITMO DE RAMIFICAR Y ACOTAR

El algoritmo de ramificar y acotar (“*branch & bound*” en inglés) es quizá el algoritmo exacto más empleado para resolver problemas enteros. En la actualidad existen diversas y muy eficientes herramientas basadas en ramificar y acotar que son implementadas computacionalmente modificando o enriqueciendo el algoritmo básico. Mayores detalles de implementaciones del algoritmo pueden encontrarse en [64].

El algoritmo de ramificar y acotar se basa en dos ideas claves.

### D.3.1 DIVIDE Y VENCERÁS

Consideremos el problema  $z = \text{máx}\{\mathbf{c}\mathbf{x} \mid \mathbf{x} \in S\}$ . Se busca dividirlo en problemas más pequeños de tal forma que resulte más fácil de resolver. Este método se basa en el siguiente resultado.

**Proposición D.1** *Sea  $S = S_1 \cup \dots \cup S_K$  una descomposición de  $S$  en conjuntos más pequeños, sea  $z^k = \text{máx}\{\mathbf{c}\mathbf{x} \mid \mathbf{x} \in S_k\}$  para  $k = 1, \dots, K$ . Entonces  $z = \text{máx}_k z^k$ .*

### D.3.2 ENUMERACIÓN IMPLÍCITA

Dado que la enumeración exhaustiva es prácticamente irrealizable, esta idea consiste en emplear inteligentemente algunas cotas de los valores  $z_k$ . El siguiente resultado indica cómo.

**Proposición D.2** *Sea  $S = S_1 \cup \dots \cup S_K$  una descomposición de  $S$  en conjuntos más pequeños, sea  $z^k = \text{máx}\{\mathbf{c}\mathbf{x} \mid \mathbf{x} \in S_k\}$  para  $k = 1, \dots, K$ ,  $\bar{z}_k$  una cota superior de  $z^k$  y  $\underline{z}_k$  una cota inferior de  $z^k$ . Entonces  $\bar{z} = \text{máx}_k \bar{z}_k$  es una cota superior de  $z$  y  $\underline{z} = \text{máx}_k \underline{z}_k$  es una cota inferior de  $z$ .*

### D.3.3 ALGORITMO DE RAMIFICAR Y ACOTAR (MAXIMIZAR)

Paso 0: Inicialización.

Colocar el problema inicial  $S^0$  con su formulación  $P^0$  en la lista.

$z = -\infty$ . Solución incumbente  $x^*$  vacía.

Paso 1: Si la lista está vacía, parar: Solución incumbente  $x^*$  óptima.

Paso 2: Elegir un problema  $S^i$  con su formulación  $P^i$ .

Paso 3: Resolver una relajación sobre  $P^i$ .

Cota dual  $\bar{z}^i =$  Valor del objetivo en la relajación de  $P^i$ .

$\tilde{x}^i =$  Solución de la relajación de  $P^i$ .

Paso 4: Si  $P^i = \emptyset$ , truncar por infactibilidad e ir a 1.

Paso 5: Si  $z^i \leq z$ , truncar por acotamiento e ir a 1.

Paso 6: Si  $\tilde{x}^i$  es entera hacer:

Actualizar cota primal  $z = \bar{z}^i$ .

Solución incumbente  $x^* = \tilde{x}^i$ .

Truncar por optimalidad e ir a 1.

Paso 7: Elegir una variable no entera  $x_j$  para ramificar.

Hacer con la variable elegida dos ramas redondeándola al entero

inmediato superior y al inmediato inferior.

Agregar dos sub problemas a la lista:  $S_1^i$  y  $S_2^i$  con formulaciones  $P_1^i$

y  $P_2^i$  respectivamente. Ir a paso 1

# BIBLIOGRAFÍA

---

- [1] Alfandari L., Plateau A. y Tolla P. (2003). A path relinking algorithm for the generalized assignment problem. En M. G. C. Resende and J. P. de Sousa, editores, *Metaheuristic: Computer Decision-Making*, 1-18.
- [2] Amini M. M. y Racer M. (1994). A robust heuristic for the generalized assignment problem. *Ann Operation Research*, 50, 487-503.
- [3] Balas E. y Christofides N. (1981). A restricted Lagrangean approach to the traveling salesman problem. *Mathematical Programming* 21, 19-46.
- [4] Barahona F. y Anbil R. (2000). The volume algorithm: producing primal solutions with a subgradient method. *Mathematical Programming* 87, 385-399.
- [5] Barnhart C., Johnson E. L., Nemhauser G. L., Savelsbergh M. W. P. y Vance P. (1998). Branch-and-price: column generation for solving huge integer programs. *Operation Research* 46, 316-329.
- [6] Beasley J. E. (1993). Lagrangean relaxation. In: Reeves C. R. (ed.), *Modern heuristic techniques for combinatorial problems*. Blackwell Scientific Publications, 243-303.
- [7] Bilde y Krarup. (1977). Sharp lower bounds and efficient algorithms for the simple plant location problem. *Annals of Discrete Mathematics* 1, 79-97.
- [8] Camerini P. M., Fratta L. y Maffoli F. (1975). On improving relaxation methods by modified gradient techniques. *Mathematical Programming Study* 3, 26-34.

- 
- [9] Chajakis E., Guignard M. y Ryu C. (1994). Lagrangean bounds and heuristic for integrated resource planning in forestry. *Proceedings of Symposium on System Analysis and Forest Management Problems*, Valdivia, Chile, 1993, 350-363.
- [10] Chajakis E., Guignard M., Yan H. y Zhu S. (1996). The Lazy Lagrangean Heuristic, *Optimization Days*, Montreal, 1996.
- [11] Chen B. y Guignard M. (1998). Polyhedral analysis and descompositions for capacited plant location-type problems. *Discrete Applied Mathematics* 82, 79-91.
- [12] Cheney E. W. y Goldstein A. A. (1959). Newton 's method for convex programming and Tchebicheff approximations. *Numerisch Mathematik* 1, 253-268.
- [13] Chu P. C. y Beasley J. E. (1997). A genetic algorithm for the generalized assignment problem. *Computers Oper. Res* 24, 17-23.
- [14] Contesse L. y Guignard M. (1995). A Proximal Augmented Lagrangean Relaxation for Linear and Nonlinear Integer Programming, Report 95-03-06, *Operations and Information Management Department*, University of Pennsylvania.
- [15] Contesse L. y Guignard M. (2002). An Augmented Lagrangean relaxation for integer programming with application to nonlinear capacited facility location. Part I: theory and algorithm, Research Report, *Operations and Information Management Department*, University of Pennsylvania.
- [16] Contesse L. y Guignard M. (2002). An Augmented Lagrangean relaxation for integer programming with application to nonlinear capacited facility location. Part II: Algorithm and Computational Results, Research Report, *Operations and Information Management Department*, University of Pennsylvania.
- [17] Dantzing G. B. y Wolfe P. (1960). The descomposition principle for linear programs. *Operations Research* 8, 101-111.
- [18] Dantzing G. B. y Wolfe P. (1961). The descomposition algorithm for linear programs. *Econometrica* 29, 767-778.

- 
- [19] Daskalaki S., Birbas T., Housos E. (2004). An integer programming formulation for a case study in university timetabling, *European Journal of Operational Research* 153 (1), 117-135.
- [20] de Matta R. y Guignard M. (1994). Dynamic production scheduling for a process industry. *Operations Research* 42, 492-503.
- [21] Desrosiers J., Soumins F. y Desrochers M. (1984). Routing with time windows by column generation. *Networks* 14, 545-565.
- [22] Desrosiers J., Sauvé M. y Soumins F. (1988). Lagrangian relaxations methods for solving the minimum fleet size multiple traveling salesman problem with time windows. *Management Science* 34, 1005-1022.
- [23] du Merle O., Goffin J.-L. y Vial J-Ph. (1998). On improvements to the analytic center cutting plane method. *Computational Optimization and Applications* 11, 37-52.
- [24] du Merle O., Villeneuve D., Desrosiers J. y Hansen P. (1999). Stabilized column generation. *Discrete Mathematics* 94, 229-237.
- [25] Escudero L., Guignard, M. y Malik K. (1994). A Lagrangean Relax-and-Cut approach for the sequential ordering problem with precedence relationships. *Annals of Operations Research* 50, 219-237.
- [26] Erlenkotter D. (1978). A dual-based procedure for uncapacitated facility location. *Operations Research* 26, 992-1009.
- [27] Everett III H. (1963). Generalized Lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research* 11, 399-417.
- [28] Fisher M. L. (1981). The Lagrangian relaxation method for solving integer programming problems. *Management Science* 27, 1-18.
- [29] Fisher M. L. (1985). An applications oriented guide to Lagrangian relaxation. *Interfaces* 15, 10-21.

- 
- [30] Fisher M. L. y Hochbaum D. S. (1980). Database location in a computer network. *Journal of the ACM* 27, 718-735.
- [31] Fisher M. L., Jaikumar R. y van Wassenhove L .N. (1986). A multiplier adjustment method for the generalized assignment problem. *Management Science* 32, 1095-1103.
- [32] Fisher M. L. y Kedia P. (1990). Optimal Solution of Set Covering/Partitioning Problems Using Dual Heuristic. *Management Science* 39, 67-88.
- [33] Franz L.S., Miller J.L. (1993). Scheduling medical residents to rotations: Solving the large-scale multiperiod staff assignment problem, *Operation Research* 41 (2), 269-279
- [34] Frangioni, A. (2005). About Lagrangian methods in integer optimization, *Annals of Operation Research*. 139, 163-169
- [35] Freville, A. and Hanafi, S. (2005). The multidimensional 0-1 knapsack problem – bounds and computational aspects, *Annals of Operations Research* 139, 195-227
- [36] Foulds Jr. L. R., Fulkerson Jhonson D. G., SlotManager (2000). A microcomputer-based decision support system for university timetabling, *Decision Support System* 27(4), 367-381.
- [37] Geoffrion A. M. (1974). Lagrangean relaxation for integer programming. *Mathematical Programming Study* 2, 82-114.
- [38] Geoffrion A. M. y McBride R. (1978). Lagrangean relaxation applied to capacitated facility location problems. *AIIE Transaction* 10, 40-47.
- [39] Glover F., Klingman D. (1988). Layering strategies for creating exploitable structure in linear and integer programs. *Mathematical Programming* 40, 165-182.



- 
- [40] Guignard M. (1994). Primal relaxation in integer programming. VII CLAIO Meeting, Santiago, Chile, 1994, and *Operations and Information Management Department* Artículo en proceso 94-02-01, University of Pennsylvania, 1994.
- [41] Guignard M. y Kim S. (1987). Lagrangean decomposition: a model yielding stronger Lagrangean bounds. *Mathematical Programming* 39, 215-228.
- [42] Guignard M. y Rosenwein M. B. (1989). An application-oriented guide for designing Lagrangian dual ascent algorithms. *European Journal of Operational Research* 43, 197-205.
- [43] Guignard M., Ryu C., Qian H. y Dowlath L. (2002). Multi-item capacitated lot-sizing problem (MCLP). IV ALIO/EURO Workshop on Applied Combinatorial Optimization, 2002, <http://www-di.inf.puc-rio.br/~celso/artigos/pucon.ps>
- [44] Guignard M. y Zhu S. (1994). A hybrid algorithm for solving Lagrangean duals in mixed-integer programming, Memorias del VI CLAIO, Santiago, Chile, 1994, 339-410.
- [45] Guignard, M. (2003) Lagrangean relaxation, *TOP*, 11(2), 151-228.
- [46] Guignard M. y Rosenwein M. (1990). An application of Lagrangean Decomposition to the resource-constrained minimum weighted arborescence problem. *Networks* 20, 345-359.
- [47] Guignard M. (2003). Lagrangean Decomposition and Lagrangean Substitution for Stochastic Integer Programming, *Operations and Information Management Department*, University of Pennsylvania.
- [48] Haddadi S. y Ouzia H. (2001). An effective Lagrangian heuristic for the generalized assignment problem. *INFOR*, 39, 351-356.
- [49] Held M. y Karp R. M. (1970). The traveling salesman problem and minimum spanning trees. *Operations Research* 18, 1138-1162.

- 
- [50] Held M. y Karp R. M. (1971). The traveling salesman problem and minimum spanning trees: part II. *Mathematical Programming* 1, 6-25.
- [51] Held M., Wolfe P. y Crowder H. (1974). Validation of subgradient optimization. *Mathematical Programming* 6, 62-88.
- [52] Jeet V. , Kutanoglu E. ,Lagrangian relaxation guided problem space search heuristics for generalized assignment problems, *European J Operational Research*, 182:1039-1056,2007.
- [53] Jörnsten K. y Näsberg M. (1986). A new Lagrangian relaxation approach to the generalized assignment problem. *European Journal of Operation Research* 27, 313-323.
- [54] Kelley J. E. (1960). The cutting-plane method for solving convex programs. *Journal of the SIAM* 8, 703-712.
- [55] Kuhn.H. W. (1955). The Hungarian method for the assignment problem, *Naval Research Logistics Quartely* 2 (1&2), 83-97.
- [56] Laguna M., Kelly P., González-Velarde J.L. y Glover F. (1995). Tabu search for the multilevel generalized assignment problem. *European Journal of Operation Research*, 82, 176-189.
- [57] Lasdon, L.S.(2002). Optimization Theory for Large Scale Systems. 2nd ed. Dover, Mineola, New York.
- [58] Lemaréchal C. (1974). An algorithm for minimizing convex functions, Memorias del congreso IFIP '74. North Holland, 552-556.
- [59] Lemarèchal, C. (2001). Lagrangean relaxation, M. Junger, D. Naddef eds. *Computational Combinatorial Optimization*. Primavera Verlag, Heidelberg, 115-160
- [60] Litvinchev I. (2007). Refinement of Lagrangian bounds in optimization problems. *Computational Mathematics and Mathematical Physics*. 47(7), 1101-1108.

- 
- [61] Lourenco H. R. y Serra D. (2002). Adaptative search heuristic for the generalized assignment problem. *Mathware and Soft Computing*, 9, 209-234.
- [62] Lucena A. (1982). Steiner problems in graphs, Lagrangean relaxation and cutting planes. *COAL Bulletin* 21, 2-8.
- [63] Martin R. K. (1999). Large Scale Linear and Integer Programming: A Unified Approach. Kluwer, Boston.
- [64] Martello S. y Toth P. (1990). Knapsack Problems: Algorithms and Computer Implentation. Wiley, N. Y.
- [65] McCarl, B. A. (2004). *GAMS User Guide: 2004, Version 21.3*, Developed in cooperation with GAMS Development Corporation.
- [66] Narciso M.G., Lorena L.A.N. , Lagrangian/surrogate relaxation for generalized assignment problems, *European J Operational Research* 114(1):165-177,1999.
- [67] Näsberg M., Jörnsten K. O. y Smeds P. A. (1985). Variable Splitting - A new Lagrangean relaxation approach to some mathematical programming problems, Reporte LITH-MAT-R-85-04, Linkoping University, 1985.
- [68] Garey, M. Jhonson. (1979). *Computers and Intractability. A guide to the Theory of NP- Completeness*. W. H. Freeman and Company, New York, EUA:
- [69] Osman I. H. (1995). Heuristic for the generalized assignment problem: Simulated annealing and tabu search approaches. *OR Spektrum*, 17, 211-225.
- [70] Pentico D. W. (2007). Assignment problems: A golden anniversary survey, *European Journal of Operation Research*, 176, 774-793.
- [71] Racer M. y Amini M. M. (1995). A hybrid heuristic for the generalized assignment problem. *European Journal of Operation Research*, 87, 343-348.
- [72] Reinoso H. y Maculan N. (1992). Lagrangean descomposition in integer linear programming: a new scheme. *INFOR* 30, 1-5.

- 
- [73] Ribero C. y Minoux M. (1986). Solving hard constrained shortest path problems by Lagrangean relaxation and branch-and-bound algorithms. *Mathematics of Operation Research* 53, 303-316.
- [74] Rockafellar R. T. y Wets R. J-B. (1999). Scenario and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research* 16, 119-147.
- [75] Ross G. T. y Soland R. M. (1975). A branch-and-bound algorithm for the generalized assignment problem. *Mathematical Programming* 8, 91-103.
- [76] Ryan D. M. y Foster B. A. (1981). An Integer Programming Approach to Scheduling. En: Wren A. (ed.), *Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling*. North Holland, 269-280.
- [77] Ryu Ch. y Guignard M. (1992). An efficient algorithm for the capacited plant location problem. Report 92-11-02, *Department of Decision Science*, University of Pennsylvania.
- [78] Saucedo Martinez J. (2007). Mejorando las Cotas Lagrangianas para el Problema de Asignación Generalizada. Universidad Autónoma de Nuevo León, México.
- [79] Savelsbergh M. (1997). A branch-and-cut algorithm for the generalized assignment problem. *Operations Research* 45, 831-841.
- [80] Shapiro J. F. (1974). A survey of Lagrangean techniques for discrete optimization. *Annals of Discrete Mathematics* 5, 113-138.
- [81] Shapiro J. F. (1979). *Mathematical Programming: structure and algorithms*. Jhon Wiley.
- [82] Shepardson F. y Martsen R. E. (1980). A Lagrangean relaxation algorithm for the two-duty scheduling problem. *Management Science* 26, 274-281.

- 
- [83] Soenen R. (1977). Contribution à l'étude des systèmes de conduite en temps réel en vue de la commande d'unités de fabrication, Thèse de Doctorat d'Etat, Université de Lille, France.
- [84] Terlaky, T. ed. (1996). *Interior Point Methods of Mathematical Programming*, Kluwer, Netherlands.
- [85] Toth P. y Martello S. (1981). An algorithm for the generalized assignment problem. En J.P. Brans, editor, *Memorias del Noveno IFORS International Conference on Operational Research*, 589-603.
- [86] Wentges P. (1997). Weighted Dantzig-Wolfe decomposition for linear mixed-integer programming. *International Transactions in Operational Research* 4, 151-162.
- [87] Werra D. de (1985). An introduction to timetabling, *European Journal of Operation Research*, 19 (2), 151-162.
- [88] Zowe J. (1985). Nondifferentiable optimization. En: Schittkowski K. (ed.), *Computational Mathematical Programming*, NATO ASI Series F: Computer and System Science, 15. Verano-Verlag, 323-356.

# FICHA AUTOBIOGRÁFICA

---

Edith Lucero Ozuna Espinosa

Candidato para el grado de Maestro en Ciencias  
con especialidad en Ingeniería de Sistemas

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Tesis:

## PROBLEMAS CON MÚLTIPLES RELAJACIONES LAGRANGIANAS

Edith Lucero Ozuna Espinosa, nací el 18 de abril de 1985 en Guadalupe, Nuevo León, México. Realicé mis estudios de matemáticas en la Facultad de Ciencias Físico-Matemáticas de la Universidad Autónoma de Nuevo León, posteriormente continué con mis estudios de maestría en el Posgrado en Ingeniería de Sistemas de la Facultad de Ingeniería Mecánica y Eléctrica de la misma Universidad.