UNIVERSITAT POLITÈCNICA DE CATALUNYA
Facultat d'Informàtica

DISEÑO Y ESTUDIO COMPUTACIONAL DE ALGORITMOS HÍBRIDOS PARA PROBLEMAS DE SET PARTITIONING

Memoria presentada por Elena Fernández Aréizaga, bajo la dirección del Profesor Jaime Barceló Bugeda, para acceder al grado de Doctor en Informática por la Universitat Politècnica de Catalunya.

Barcelona

Enero, 1988

IV.1 Justificación algorítmica

En una primera aproximación a los problemas (KP1) y (KP2), resulta sugerente la utilización de la llamadas restricciones agregadas. Una restricción agregada para un problema (P) es una restricción equivalente al conjunto original de restricciones que define (P). Es decir, si el conjunto de restricciones de (P) viene dado por Ax=b, una restricción $rx=r_0$ es una restricción agregada si, por definición, $\forall x$ se cumple que $Ax=b \Leftrightarrow rx=r_0$.

Las restricciones agregadas han sido ampliamente estudiadas y su utilización resulta, a priori, atractiva puesto que permiten formular un problema equivalente que tenga una única restricción. Podemos considerar las restriciones agregadas como un caso particular, muy especial, de restricciones surrogadas, en el que la restricción surrogada tambíen implica todo el conjunto original de restricciones. En este sentido, las restricciones agregadas serán combinaciones lineales de las restricciones originales (obviamente, desamos que el problema resultante siga siendo un problema de programación lineal).

Existen distintos métodos para calcular los coeficientes de esta combinación lineal. El primero de ellos, de Elmaghraby y Wig [EmWi70], fue sucesivamente mejorado hasta llegar al método de Kendall y Zionts [KeZi72, Zio74] que es el más utilizado actualmente. En este método, el cálculo de los coeficientes de la restricción agregada se hace hace en base al siguiente

Teorema 1: Dadas dos restricciones de la forma

(1) $a_1x=b_1$
(2) $a_2x=b_2$
$x\geq 0$ y entero
$x_j \leq 1$ $j\in\{1,...,p\}$ $(p<n)$,

entonces, las ecuaciones (1) y (2) son equivalentes a la ecuación

$$\mu_1 a_1x + \mu_2 a_2x = \mu_1 b_1 + \mu_2 b_2,$$

donde los multiplicadores μ_1 y μ_2 satisfacen las siguientes condiciones:

i) $\mu_1 > b_2 - \sum_{j\in N} a_{2j}^- - \min_{a_{2j}\neq 0}\{|a_{2j}|\}$

$\mu_2 > b_1 - \sum_{j\in N} a_{1j}^- - \min_{a_{1j}\neq 0}\{|a_{1j}|\}$

ii) $\dfrac{b_2 - \sum_{j\in N} a_{2j}^-}{\mu_1}$ y $\dfrac{b_1 - \sum_{j\in N} a_{1j}^-}{\mu_2}$ no son enteros.

iii) μ_1 y μ_2 son primos entre sí (el máximo común divisor de μ_1 y μ_2 es 1)

siendo $a_{ij}^- = \min\{a_{ij}, 0\}$, $i=1,2$ y $N=\{1,...,n\}$.

El teorema anterior permite calcular de forma iterativa la restricción agregada para un conjunto de restricciones, calculando, en la iteración k la restricción agregada correspondiente a las k+1 primeras restricciones. El inconveniente de este método es que, a medida que el número de restricciones del problema original aumenta, el tamaño de los coeficientes resultantes crece excesivamente dando lugar a problemas de almacenamiento. Existe, sin embargo, un trabajo reciente de Elmaghraby [Ema86], en el que se reduce logarítmicamente la magnitud de estos coeficientes.

En el caso de los problemas (KP1) y (KP2), debido a que tienen sólamente dos restricciones, los coeficientes de la restricción agregada generada por el método de Kendall y Zionts [KeZi72] son de una magnitud reducida pudiendo, por tanto, aplicarse dicho método sin inconvenientes de ese tipo. En particular, en el caso de (KP1) puede escribirse el conjunto de restricciones en forma de igualdad resultando

$$\sum_{j\in N} n_j x_j + s_1 = m$$

$$\sum_{j\in N} -c_j x_j + s_2 = -s_0$$

donde s_1 y s_2 son variables de holgura.

Aplicando el teorema anterior, la restricción agregada será

$$\sum_{j\in N} (\mu_1 n_j - \mu_2 c_j)\, x_j + \mu_1 s_1 + \mu_2 s_2 = \mu_1 m - \mu_1 s_0$$

siendo μ_1 y μ_2 dos números no negativos tales que

i) $\mu_1 > -s_0 + \sum_{j\in N} c_j - 1$

$\mu_2 > m - 1$

ii) $\dfrac{-s_0 + \sum_{j\in N} c_j}{\mu_1}$ y $\dfrac{m}{\mu_2}$ no enteros

iii) μ_1 y μ_2 primos entre sí.

Hemos de observar que la restricción obtenida deberá utilizarse siempre con una formulación como estricta igualdad. Esto puede observarse definiendo una nueva variable $s = \mu_1 s_1 + \mu_2 s_2$ con lo que la restricción agregada queda

$$\sum_{j\in N} (\mu_1 n_j - \mu_2 c_j)\, x_j + s = \mu_1 m - \mu_2 s_0$$

Es evidente que no podemos utilizar esta restricción en forma de desigualdad puesto que

$$\sum_{j\in N} (\mu_1 n_j - \mu_2 c_j)\, x_j \leq \mu_1 m - \mu_2 s_0$$

no sería una formulación correcta, ya que el rango de valores para la variable s está limitado a aquellos que sean combinación lineal de los valores μ_1 y μ_2, no pudiendo, por tanto, considerar s como una variable de holgura.

Un análisis similar puede realizarse para el problema (KP2) y en general para aquellos problemas en los que alguna de las restricciones originales no sea de estricta igualdad. Para estos problemas será necesario, al igual que para los que tengan un conjunto inicial de restricciones de estricta igualdad, mantener la formulación de estricta igualdad de la restricción agregada. Esta formulación significa un obstáculo importante a la hora de su utilización, ya que no existen algoritmos que resuelvan de una forma especialmente eficiente estos problemas, al contrario de lo que ocurre con problemas cuya restricción es de desigualdad.

La observación anterior justifica el enfoque algorítmico seguido, ya que en este contexto resulta más adecuado diseñar algoritmos específicos para los problemas (KP1) y (KP2), que saquen partido del conocimiento de la estructura original de los mismos.

IV.2 Problemas con dos restricciones de sentido contrario

Consideremos ahora un problema de la forma

$$
\text{(KP1)} \qquad \begin{array}{l} \min dx \\ cx \geq c_0 \\ gx \leq g_0 \\ x_j \in \{0,1\},\ j \in N, \end{array}
$$

donde $c,g \in R^{n+}$, $d \in R^n$.

Notemos que un problema con esta estructura no permite, a priori, reducir sus dimensiones, fijando el valor de alguna variable. Ello se debe al hecho de que, al ser todos los coeficientes de las restricciones no negativos, no podremos asegurar que una variable tenga que fijarse a 1 en una solución óptima aunque su coeficiente de coste sea negativo, porque su coeficiente en la segunda restricción puede no resultar adecuado. Lo mismo ocurre a la hora de intentar fijar a 0 variables que tengan un coeficiente positivo en la función objetivo, ya que esa variable puede resultar necesaria para satisfacer la primera restricción.

Desde una interpretación intuitiva del problema (KP1), resulta evidente que si el conjunto de todas las variables que tienen un coeficiente de coste negativo, formase una solución posible, esta solución sería la óptima para el problema. En particular, la solución óptima del problema deberá contener 'el mayor número posible' de estas variables. Por este motivo, intentaremos obtener soluciones que, aunque no sean enteras, persigan este objetivo, dando lugar a cotas inferiores para (KP1).

Teniendo en cuenta que la restricción que limita el número de variables con coeficiente negativo que pueden incluirse en una solución es la segunda, podemos reordenar el conjunto de índices de estas variables, poniendo en primer lugar aquellas que resulten más 'prometedoras'. En concreto, una de estas variables resulta tanto más sugerente en la medida en que su coeficiente de coste es menor (mayor en valor absoluto) y su coeficiente g_j también lo es. Por lo tanto consideramos que el conjunto de estos índices est ordenado según valores crecientes de d_j/g_j . Sea $N^1=\{1,...,p\}$ el conjunto de estos índices.

Por otro lado, la segunda restricción no limita la inclusión de variables con coeficientes de coste negativo, pero puede obligar a introducir en la solución alguna variable cuyo coeficiente de coste sea positivo. Por este motivo, deberemos ordenar los índices de variables con coeficiente de coste positivo poniendo en primer lugar las que resulten menos 'perjudiciales'. En concreto, una de estas variables resulta peor candidata para formar parte de una solución en la medida en que su coeficiente de coste sea mayor y su coeficiente c_j menor. Por lo tanto consideramos que el conjunto de índices de estas variables está ordenado según valores crecientes de los cocientes d_j/c_j. Sea $N^2=\{p+1,...,n\}$ el conjunto de estos índices.

Podemos considerar, por tanto, que el conjunto N de los índices de todas las variables está ordenado de forma que primero aparecen los índices de las de coeficiente de coste negativo, según la ordenación de N^1, y después los de las variables de coeficiente de coste positivo, según la ordenación de N^2.

IV.2.1 Cálculo de cotas inferiores

A continuación se presentan dos cotas inferiores para el problema (KP1). Estas cotas se obtienen considerando los problemas relajados resultantes de eliminar cada una de las restricciones de (KP1) manteniendo la otra.

Proposición 1: Sea $N^1=\{1,...,p\}$ el conjunto de índices de variables definido anteriormente, entonces

$$z^1_{inf} = \sum_{j=1}^{s_1} d_j\, x_j^*$$

es una cota inferior para (KP1), siendo

$$s_1 = \max \{ k\in N^1 \,/ \sum_{j=1}^{k} g_j \leq g_0 \}$$

y $x^*\in R^n$ el vector de componentes

$$x_j^* = \begin{cases} 1 & \text{si } 1 \leq j \leq s_1 \\ (g_0 - \sum_{j\leq s_1} g_j)/\, g_{s1+1} & \text{si } j = s_1+1 \text{ y } s_1+1 \leq p \\ 0 & \text{en otro caso} \end{cases}$$

Demostración: Sea el problema

$$\text{(RL1)} \qquad \begin{array}{l} \min dx \\ gx\leq g_0 \\ 0\leq x_j\leq 1,\ j\in N, \end{array}$$

donde $c,g\in R^{n+}$, $d\in R^n$.

El vector x^* es una solución óptima para (RL1) y z_{inf}^1 es el valor de la función objetivo para esta solución.

Teniendo en cuenta que (RL1) es una relajación de (KP1), resulta evidente que z_{inf}^1 es una cota inferior del valor óptimo de (KP1) ♦

En el caso en el que x^* satisfaga también la segunda restricción, se tiene el siguiente corolario.

Corolario 1: Sea x^* definido como en la proposición 1. Si

$$\sum_{j=1}^{s_1} c_j x_j^* \geq c_0$$

entonces x^* es un óptimo del problema (RL)

$$\text{(RL)} \qquad \begin{aligned} &\min dx \\ &cx \geq c_0 \\ &gx \leq g_0 \\ &0 \leq x_j \leq 1,\ j \in N, \end{aligned}$$

donde $c,g \in R^{n+}$, $d \in R^n$.

Nótese que (RL) es la relajación lineal ordinaria de (KP1).

Evidentemente, en la situación del corolario anterior si todas las componentes de x^* son enteras, se tratar un óptimo de (KP1). Hay que resaltar que todas las componentes de x^* serán enteras en uno de los dos siguientes casos:

i) $s_1 = p$

ii) $s_1 \leq p$ y $\sum_{j=1}^{s_1} g_j = g_0$

Consideremos ahora el problema lineal (RL2)

$$\text{(RL2)} \qquad \begin{aligned} &\min\ dx \\ &cx \geq c_0 \\ &0 \leq x_j \leq 1,\ j \in N, \end{aligned}$$

Proposición 2: Sea $N=\{1,...,p,...,n\}$ el conjunto de índices de variables definido anteriormente, entonces

$$z_{inf}^2 = \sum_{j=1}^{s_2} d_j x_j^*$$

es una cota inferior para (KP1), siendo

$$s_2 = \max \{ p, \ \min \{k \in N / \sum_{j=1}^{k} c_j \geq c_0\} \}$$

y $x^* \in R^n$ el vector de componentes

$$x_j^* = \begin{cases} 1 & \text{si } j \leq p \\ 1 & \text{si } p < j \leq s_2-1 \\ (c_0 - \sum_{j<s_2} c_j)/c_{s2+1} & \text{si } j = s_2 \\ 0 & \text{en otro caso} \end{cases}$$

Demostración: El vector x^* es una solución óptima del problema (RL2) y z_{inf}^2 el valor de esta solución; por lo tanto se trata de una cota inferior para (KP1) ♦

Corolario 2: Sea x^* definido como en la proposición 2. Si

$$\sum_{j=1}^{s_2} g_j x_j^* \leq g_0$$

entonces x^* es un óptimo del problema (RL).

Corolario 3: $z_{inf}=\max \{z_{inf}^1, z_{inf}^2\}$ es una cota inferior para (KP1).

Una observación importante es que, en el caso que el índice s_1 asociado a z_{inf}^1 sea igual a p, si la solución x^* definida como en la proposición 1 no satisface la primera restricción de (KP1), entonces el índice s_2 asociado a z_{inf}^2 será estrictamente mayor que p, con lo que la cota inferior z_{inf}^2 obtenida por la Proposición 2 será estrictamente mayor que z_{inf}^1. No puede asegurarse que esto vaya a ser cierto en el caso en el que $s_1>p$, puesto que la suma de los coeficientes de coste de las variables s_1+1 hasta p puede ser, en valor absoluto, mayor que la suma de los coeficientes de coste de las variables desde p+1 hasta s_2, en cuyo caso la cota z_{inf}^2 será menor que z_{inf}^1.

Tanto la cota z_{inf}^1 como z_{inf}^2 pueden reforzarse posteriormente teniendo en cuenta las condiciones de integridad de las variables de forma análoga a la de Martello y Toth [MaTo79] en el caso de problemas de una restricción. En concreto, en el caso de z_{inf}^1 teniendo en cuenta que x_{s1} no puede tomar un valor fraccional, la solución óptima del problema entero asociado (RL1) puede obtenerse a partir de la solución óptima x^* de (RL1) bien eliminando el elemento s_1-ésimo (x_{s1}=0), bien insertandolo (x_{s1}=0).

En el primer caso, el valor de la solución para el problema entero asociado a (RL1) no puede superar

$$b_1 = \sum_{j=1}^{s_1-1} d_j + \lfloor (g_0 - \sum_{j \le s_1} g_j)\, d_{s_1+1}/g_{s_1+1} \rfloor$$

que corresponde al caso en el que se introduzca en la solución óptima la mejor variable de las restantes, es decir x_{s1+1}.

En el segundo caso, teniendo en cuenta que será necesario eliminar aluna de las s_1-1 primeras variables el mejor valor posibles de la solución vendrá dado por

$$b_2 = \sum_{j=1}^{s_1-1} d_j + \lfloor d_{s_1} - (g_{s_1} - (g_0 - \sum_{j \le s_1} g_j))\, d_{s_1-1}/g_{s_1-1} \rfloor$$

donde se supone que la variable que ha sido eliminada tiene exactamente el menor coeficiente necesario g_j (es decir g_{s1}- (g_0-$\sum g_j$)) y el peor valor posible de d_j/g_j (es decir d_{s1-1}/g_{s1-1}).

Por lo tanto la cota z_{inf}^1 podrá reforzarse a max $\{b_1,b_2\}$. De forma análoga se puede reforzar la cota z_{inf}^2.

Hay que resaltar que este procedimiento, a pesar de su sencillez, produce cotas inferiores de una calidad apreciable, obteniendo en ocasiones la solución óptima, como veremos posteriormente.

El procedimiento anterior permite, además, obtener dos cotas inferiores asociadas a cada una de las variables. En concreto, para cada variable x_j, estas cotas son las que se obtendrían para los problemas resultantes de fijar x_j respectivamente a 0 ó 1 en (KP1). Es decir, para cada variable x_j se definen los siguientes subproblemas asociados

$$(KP1_j) \quad \min \sum_{k\in N\setminus\{j\}} d_k x_k$$

$$\sum_{k\in N\setminus\{j\}} c_k x_k \geq c_0 - c_j$$

$$\sum_{k\in N\setminus\{j\}} g_k x_k \leq g_0 - g_j$$

$$x_k \in \{0,1\},\ k\in N\setminus\{j\}$$

donde $c_k, g_k \in R^+$, $d_k \in R$ $k\in N\setminus\{j\}$ y

$$(KP0_j) \quad \min \sum_{k\in N\setminus\{j\}} d_k x_k$$

$$\sum_{k\in N\setminus\{j\}} c_k x_k \geq c_0$$

$$\sum_{k\in N\setminus\{j\}} g_k x_k \leq g_0$$

$$x_k \in \{0,1\},\ k\in N\setminus\{j\}$$

donde $c_k, g_k \in R^+$, $d_k \in R$ $k\in N\setminus\{j\}$

Sean $z0_j$ y $z1_j$ las cotas inferiores obtenidas mediante el procedimiento anterior para $(KP0_j)$ y (KP1j) respectivamente. Nótese que si $s=\max\{s_1,s_2\}$ entonces $\forall j \leq s+1$ $z0_j=z_{inf}$ y $\forall j \geq s+1$ $z1_j=z_{inf}$.

El conocimiento de estas cotas permite establecer un criterio de fijación de variables a 0 o a 1 como veremos posteriormente.

IV.2.2 Algoritmo de enumeración implícita

A continuación se propone un algoritmo de enumeración implícita para resolver al óptimo problemas del tipo (KP1). En este algoritmo se supone que el conjunto de índices de variables está ordenado de forma que aparecen primero los índices asociados a variables con coeficiente de coste negativo, en el mismo orden que para el cálculo de las cotas inferiores, y después los correspondientes a las de coeficiente de coste positivo; ahora, estos índices están ordenados según orden creciente de coeficiente de coste. El criterio que se ha seguido a la hora de diseñar este algoritmo es el de construir soluciones a partir de subsoluciones parciales, a las que imponemos siempre cumplir la segunda restricción.

A partir de una subsolución parcial se intenta completar una solución fijando a 1 la siguiente variable, que no haya sido previamente considerada, que no haga violar la segunda restricción y cuyo coeficiente de coste, en caso de ser no negativo, no haga que el valor de la función objetivo asociado a la nueva solución parcial sea mayor que el valor de la solución incumbente. Si la nueva subsolución parcial satisface la primera restricción, se habrá obtenido una nueva incumbente, en caso contrario se siguen añadiendo variables.

En el caso que una variable k, candidata a formar parte de la solución, haga violar la segunda restricción, se elige como siguiente variable candidata una de las variables posteriores j que tenga un coeficiente g_j menor que g_k; en caso que esta no exista se hace backtracking. Para ello asociada a cada variable j se define $ming_j = \min\{k>j \,/\, d_k<g_j\}$.

Si la inclusión en la solución parcial de una variable k, cuyo coeficiente de coste es positivo, produjese que el valor de la función objetivo parcial superara el de la incumbente, también se haría backtracking, puesto que cualquiera de las variables posteriores tiene un coeficiente de coste al menos tan grande como d_k.

En el caso de las variables de coeficiente de coste negativo, éstas se van añadiendo a la solución parcial mientras no se viole la segunda restricción. Hay que tener en cuenta que, en este caso, aún cuando se llegue a satisfacer la primera restricción, podríamos mejorar el valor de la incumbente añadiendo tantas variables con d_j negativo como sea posible sin violar la segunda restricción.

En todo momento podemos saber si es posible llegar a satisfacer la primera restricción añadiendo variables adicionales, independientemente de que se vaya a violar la segunda o no. Si esto no va a ser posible también haremos backtracking. Para ello asociado a cada variable j de define $sumc_j = \sum_{k\geq j} c_k$.

Cuando sea preciso efectuar un backtracking, si éste es debido a que se va a superar el valor de la incumbente, a que no va a poder satisfacerse la primera restricción, a que no exista ninguna variable candidata para completar la solución, o bien a que se haya completado una solución posible cuya última componente tenga un coeficiente de coste positivo, podemos eliminar, de la solución parcial que se tenga, todas aquellas variables que aparezcan correlativas en orden decreciente a la última que se haya introducido. Esto es así ya que, debido a la ordenación que se tiene, en ninguno de estos casos podrá mejorarse el valor de la incumbente con alguna solución que contenga alguna colección de variables formada por, exactamente, las mismas primeras variables y alguna subcolección de las variables correlativas que aparecen al final. Nótese que esta eliminación de varias variables a la vez puede acelerar considerablemente la exploración.

La eliminación anterior no podrá realizarse cuando se haga backtracking porque que no exista ninguna variable que se pueda añadir y siga cumpliendo la segunda restricción; en este caso, se eliminará de la solución parcial únicamente la última variable que se haya introducido en la misma.

Tenemos, además el siguiente criterio de eliminación de variables:

Sea $z0_j$ el valor de la cota inferior asociada a una variable j y sea z_{sup} el valor de la solución incumbente. Entonces, si $z0_j \geq z_{sup}$ podremos fijar x_j a 1, ya que cualquier solución que no la contenga tendrá un valor de la función objetivo por lo menos tan grande como el valor de la incumbente.

Análogamente, si $z1_j \geq z_{sup}$ podremos fijar dicha variable a 0, puesto que cualquier solución que la contenga tendrá un valor de la función objetivo por lo menos tan grande como el valor de la incumbente.

Hay que resaltar que la aplicación de este criterio de eliminación, a menudo reduce sensiblemente la dimenión del problema original, siendo en cualquier caso mejor que los criterios de eliminación clásicos para problemas generales enteros con variables 0 ó 1.

Inicialmente suponemos que se conoce una solución posible para el problema original obtenida mediante una heurística y que, por lo tanto, se dispone de una cota superior, mejor que ∞, para el valor del problema. Además, el conocimiento de esta cota superior permite aplicar los tests de eliminación de variables antes mencionados. Posteriormente se presentará una heurística específica para problemas de este tipo, especialmente adecuada para aquellos que tengan las características numéricas similares a las que se dan en en el caso de relajaciónes lagrangianas del tipo (RL6u).

La exploración terminará cuando se hayan examinado implícitamente todas las posibles soluciones. Es decir, cuando no exista ninguna posible solución que mejore el valor de la incumbente, o bien cuando se encuentre una incumbente con un valor igual al de la cota inferior.

A continuación se expone un esquema de dicho algoritmo:

<u>Notación</u>:

n — Número total de variables que intervienen en el problema.

ultneg	Indice de la última variable con coeficiente de coste negativo.
$ming_j = \min\{k>j \,/\, g_k<g_j\}$	Indice de la primera variable posterior a j con un coeficiente en la segunda restricción menor que el de x_j.
$sumc_j = \sum_{k\geq j} c_k$	Suma de los coeficientes de la primera restricción correspondientes a variables con índice igual o posterior a j.
$sumd\_j = \sum_{k=1}^{ultneg} d_k$	Suma de los coeficientes de coste de aquellas variables que lo tengan negativo y con índice igual o posterior a j.
xp	Vector n-dimensional de componentes 0 ó 1, cuyas componentes fijadas a 1 indican los índices de las variables que forman parte de una subsolución parcial.
$r1 = \sum_{j=1}^{n} c_j xp_j$	Valor de la primera restricción para una subsolución parcial xp.
$r2 = \sum_{j=1}^{n} g_j xp_j$	Valor de la segunda restricción para una subsolución parcial xp.
$fp = \sum_{j=1}^{n} d_j xp_j$	Valor de la función objetivo para una subsolución parcial xp.
ampli(xp,j)	Test lógico que indica si la solución parcial xp es ampliable en la componente j. Es decir si al hacer $xp_j=1$ la solución parcial resultante cumple:

i) $r2 \leq g_0$	(No viola 2^a restricción)
ii) $fp + sumd\_j < incumb$	(Puede mejorar el valor de la incumbente).

Algoritmo

Inicialización

Calcular Incumb	*Valor de la cota superior obtenido mediante una heurística.*
Si incumb $\leq z_{inf}$ **Terminar**	*La solución asociada a la incumbente es óptima.*
Aplicar test de eliminación de variables	
Para j=1,n **Hacer**	*Se empieza con la subsolución parcial vacía.*
$xp_j=0$	
Fin Para	
$r1 \leftarrow 0$	
$r2 \leftarrow 0$	
$fp \leftarrow 0$	
$j \leftarrow 1$	
fin = falso	

Fin Inicialización

Mientras no fin **Hacer**

Ampliar subsolución parcial xp	
Si $r1 \geq c_0$ y $r2 \leq g_0$ y $fp < incumb$ **Entonces**	*La solución parcial xp es una solución posible mejor que la incumbente.*
Actualizar solución incumbente	
Fin Si	
Si fp+ Sumd_j $\geq$ Incumb **Entonces** **Hacer** Backtracking 1	*La subsolución parcial xp no puede completarse mejorando el valor de la incumbente*
En otro caso si r1+ $sumc_j<c_0$ **Entonces** **Hacer** Backtracking 1	*La subsolución parcial xp no se puede completar satisfaciendo la 1ª restricción.*
En otro caso si r2+ $ming_j>g_0$ **Entonces** **Hacer** Backtracking 2	*Cualquier ampliación de la subsolución parcial xp viola la 2ª restricción.*
Fin si	

Fin Mientras

Backtracking 1 y backtracking 2 son, respectivamente, los pasos de retroceso que se han mencionado en la explicación previa al algoritmo.

Para una mayor claridad en la exposición, las cantidades fp, r1 y r2, asociadas a una solución parcial xp, se suponen en todo momento debidamente actualizadas.

El procedimiento anterior termina, bien cuando se tenga una incumbente menor o igual que la cota inferior z_{inf}, ya que la solución asociada a dicha incumbente ser óptima, bien cuando en alguno de los pasos de eliminación de variables se detecta que no existe posibilidad de retroceder ni de encontrar ninguna variable candidata; en este caso, la solución que haya proporcionado la incumbente que se tenga en ese momento, será óptima.

El procedimiento ampliar subsolución parcial xp es el siguiente:

Ampliar subsolución parcial xp

Buscar $k \geq j$ tal que ampli(xp,k) sea cierto
Hacer $xp_k \leftarrow 1$
$j \leftarrow k+1$
Fin Hacer
Mientras ampli(xp,j) sea cierto **Hacer**
$xp_j \leftarrow 1$
$j \leftarrow j+1$
Fin Mientras

Fin

Evidentemente, en el algoritmo anterior puede incluirse la utilización del test de eliminación de variables cada vez que se actualice el valor de la incumbente. Se ha comprobado empíricamente que, en el caso de que la cota superior inicial sea suficientemente ajustada, no es rentable su utilización, ya que el número de variables que se eliminarán es bastante reducido comparativamente con la cantidad de variables que se eliminan inicialmente.

IV. 3 Problemas con dos restricciones del mismo sentido

Consideremos ahora un problema de la forma

$$
\begin{array}{ll}
\text{(KP2)} & \min dx \\
 & cx \geq c_0 \\
 & gx \geq g_0 \\
 & x_j \in \{0,1\},\ j \in N,
\end{array}
$$

donde $c, g \in R^n$, $d \in R^{n+}$ y $c_0, g_0 \in R^+$.

Hay que tener en cuenta que esta formulación se corresponde con una clase más amplia de problemas que es

$$
\begin{array}{ll}
\text{(KP2')} & \min dx \\
 & cx \geq c_0 \\
 & gx \geq g_0 \\
 & x_j \in \{0,1\},\ j \in N,
\end{array}
$$

donde $c, g \in R^n$, $d \in R^n$ y $c_0, g_0 \in R^+$,

puesto que si una variable de un problema (PK2') tiene un coeficiente de coste $d_j < 0$, basta con hacer un cambio de variable $y_j = 1 - x_j$ para obtener un problema transformado equivalente del tipo (KP2).

Además en esta transformación, si los coeficientes iniciales c_j y g_j son ambos positivos, la variable x_j puede fijarse a 1 y eliminarla del problema, actualizando los valores de los términos independientes de las restricciones. Supongamos, por lo tanto, que tenemos un problema del tipo (KP2).

Es posible hacer una eliminación previa de algunas variables, ya que si x_j es tal que tanto c_j como g_j son ≤ 0 entonces dicha variable nunca estará en una solución óptima, por lo que puede fijarse a 0.

Resulta, en este caso, difícil intuir cuáles son las variables más 'prometedoras' para formar parte de una solución óptima. Evidentemente, las que tengan un coeficiente de coste menor serán, a priori, mejores candidatas pero sólo en el caso de que sus coeficientes en las dos restricciones sean suficientemente grandes, ya que, de no ser así, su inclusión en una solución daría lugar a la incorporación de otras muchas variables con lo que se perdería su interés de cara a la función objetivo.

Resultaría, sin embargo, más fácil tener una idea clara de cuales serían las variables 'mejores' si el problema (KP2) tuviese una única restricción. Una observación trivial es que estas variables no serían las mismas para cumplir la primera restricción que para la segunda. En cualquier caso, las dos relajaciones de (KP2) obtenidas eliminando una de las dos restricciones, pueden llevarnos a obtener cotas inferiores para el problema original, así como proporcionarnos información sobre cual de las dos desigualdades resulta más restritiva para la resolución de (KP2). Esta información podrá utilizarse para definir una estrategia a la hora de diseñar de un algoritmo de enumeración implícita.

IV.3.1 Cálculo de cotas inferiores

Consideremos las siguientes relajaciones del problema (KP2)

(RL) $\min dx$

$cx \geq c_0$

$gx \geq g_0$

$0 \leq x_j \leq 1, j \in N,$

donde $c,g \in R^n$, $d \in R^{n+}$ y $c_0, g_0 \in R^+$.

(RL1) $\min dx$

$gx \geq g_0$

$0 \leq x_j \leq 1, j \in N,$

donde $g \in R^n$, $d \in R^{n+}$ y $g_0 \in R^+$.

(RL2) $\min dx$

$cx \geq c_0$

$0 \leq x_j \leq 1, j \in N,$

donde $c \in R^n$, $d \in R^{n+}$ y $c_0 \in R^+$.

Sea $N^1 = \{ 1,...,n \}$ el conjunto de índices de variables que se suponen ordenados de la siguiente forma: Primero aquellos correspondientes a variables con coeficientes $g_j \geq 0$ por orden creciente de d_j/g_j y despúes los índices correspondientes a las variables con $g_j < 0$ ordenados según valores decrecientes de d_j/g_j (creciente en valor absoluto).

Proposición 3: Sea $N^1=\{1,...,n\}$ el conjunto de índices de variables definido anteriormente, entonces

$$z^1_{inf} = \sum_{j=1}^{s_1} d_j x_j^*$$

es una cota inferior para (KP2), siendo

$$s_1 = \min \{ k \in N / \sum_{j=1}^{k} g_j \geq g_0 \}$$

y $x^* \in R^n$ el vector de componentes

$$x_j^* = \begin{cases} 1 & \text{si } j < s_1 \\ (g_0 - \sum_{j<s1} g_j) / g_{s1} & \text{si } j = s_1 \\ 0 & \text{en otro caso} \end{cases}$$

Demostración : Es evidente ya que x^* es una solución óptima para el problema (RL1) que es una relajación de (KP2) y $z_{inf}{}^1$ el valor de la función objetivo para esta solución ♦

Corolario 4: Sea x^* definido como en la Proposición 3. Si

$$\sum_{j=1}^{s_1} c_j x_j^* \geq c_0$$

entonces x^* es un óptimo del problema (RL)

Consideremos ahora el conjunto $N^2= \{ 1,...,n \}$ de índices de variables que se suponen ordenados de forma que primero aparecen los índices correspondientes a variables con c_j positivo, según el orden creciente de los cocientes d_j/c_j, y después los índices de las variables de coeficiente c_j negativo ordenadas por valor decreciente (creciente en valor absoluto) de d_j/c_j

Proposición 4: Sea $N^2=\{1,...,n\}$ el conjunto de índices de variables que se acaba de definir, entonces

$$z_{inf}^2 = \sum_{j=1}^{s_2} d_j\, x_j^*$$

es una cota inferior para (KP2), siendo

$$s_2 = \min \{ k\in N \,/\, \sum_{j=1}^{k} c_j \geq c_0 \}$$

y $x^*\in R^n$ el vector de componentes

$$x_j^* = \begin{cases} 1 & \text{si } j < s_2 \\ (c_0 - \sum_{j<s2} c_j) \,/\, c_{s2} & \text{si } j = s_2 \\ 0 & \text{en otro caso} \end{cases}$$

Demostración : Es evidente ya que x^* es una solución óptima para el problema (RL2) que es una relajación de (KP2) y z_{inf}^2 el valor de la función objetivo para esta solución ♦

Corolario 5: Sea x^* definido como en la Proposición 4. Si

$$\sum_{j=1}^{s_2} g_j\, x_j^* \geq g_0$$

entonces x^* es un óptimo del problema (RL)

Tanto la cota z_{inf}^1 como z_{inf}^2 pueden reforzarse posteriormente teniendo en cuenta las condiciones de integridad de las variables de forma análoga a la que se ha explicado anteriormente para el caso de dos restricciones de sentido contrario siguiendo el criterio de Martello y Toth [MaTo79] en el caso de problemas de una restricción.

Corolario 6: $z_{inf} = \max \{z_{inf}^1, z_{inf}^2\}$ es una cota inferior para (KP2).

De una forma similar al caso del problema (KP1), podemos obtener dos cotas inferiores, asociadas a cada variable, para el caso en que éstas se fijen a 0 ó a 1.

Para ello, dada una variable j se definen los subproblemas

$$(KP1_j) \quad \min \sum_{k\in N\setminus\{j\}} d_k x_k$$

$$\sum_{k\in N\setminus\{j\}} c_k x_k \geq c_0 - c_j$$

$$\sum_{k\in N\setminus\{j\}} g_k x_k \geq g_0 - g_j$$

$$x_k \in \{0,1\},\ k\in N\setminus\{j\}$$

donde $\forall\ k\in N\setminus\{j\}$ $c_k, g_k \in R^+$, $d_k \in R$ $k\in N\setminus\{j\}$ y c_0, $g_0 \in R^+$, y

$$(KP0_j) \quad \min \sum_{k\in N\setminus\{j\}} d_k x_k$$

$$\sum_{k\in N\setminus\{j\}} c_k x_k \geq c_0$$

$$\sum_{k\in N\setminus\{j\}} g_k x_k \geq g_0$$

$$x_k \in \{0,1\},\ k\in N\setminus\{j\}$$

donde $\forall k\in N\setminus\{j\}$ $c_k, g_k \in R^+$, $d_k \in R$ $k\in N\setminus\{j\}$ y $c_0, g_0 \in R^+$,

y se aplica el procedimiento anterior para la obtención de una cota inferior a cada subproblema.

De esta forma obtendremos dos cotas inferiores $z1_j$ y $z0_j$ que son, respectivamente, la cota inferior que obtendríamos si en el problema original (KP2) se hubiese fijado xj a 0 ó a 1. Posteriormente podremos utilizar estas cotas en un procedimiento de enumeración implícita para eliminar variables, fijándolas a 0 ó a 1, reduciendo, de esta forma, las dimensiones del problema original.

IV.3.2 Algoritmo de enumeración implícita

A continuación se propone un algoritmo de enumeración implícita para el problema (KP2). Se trata de un algoritmo que utiliza criterios semejantes al de Martello y Toth [MaTo79] para problemas con una única restricción, imponiendo además, a las subsoluciones parciales que se obtengan, poder satisfacer la otra restricción. Es importante en el diseño de este algoritmo conocer a priori cual es la desigualdad más 'restrictiva'. Esta será la restricción que haya proporcionado la mayor de las dos cotas inferiores, ya que en principio parece ser la restricción más difícil de satisfacer y permitirá eliminar rápidamente partes importantes del espacio de búsqueda. Será a esta restricción a la que se apliquen fundamentalmente los criterios de poda en el algoritmo. Por lo tanto, existirán dos versiones diferentes del algoritmo que se propone, en función de que la restricción más vinculante sea la primera o la segunda.

Sin pérdida de generalidad y para una mayor claridad en la exposición, supondremos a partir de ahora que el subproblema que ha proporcionado la mejor cota inferior es (RL1) por lo que se toma la primera restricción como la más vinculante. Sea $N^1 = \{1,...,n\}$ el conjunto de índices de las variables ordenados según el mismo criterio que en la Proposición 3.

En este procedimiento se construyen subsoluciones parciales para el problema (KP2), que no satisfacen la primera restricción. Estas se completan, añadiendo el menor número posible de elementos, hasta obtener una solución posible. Para completar una subsolución parcial, se incluye en la misma, mientras no se satisfaga la primera restricción, la primera variable que no haya sido previamente considerada, que pueda mejorar el valor de la incumbente y que permita satisfacer las dos restricciones. Cuando la nueva subsolución parcial satisfaga ambas restricciones habrá que comprobar si el valor de la función objetivo asociado a la misma mejora el valor de la incumbente para actualizarlo.

Dada una subsolución parcial, Martello y Toth [MaTo79] proponen un criterio para saber a priori si ésta no va a poder proporcionar, en caso de que llegue a completarse, un valor de la función objetivo mejor que el de la incumbente que se tenga. En particular, dada una solución parcial xp el valor

$$zxp\_ = \sum_{j=1}^{n} d_j . xp_j + \lfloor (c_0 - \sum_{j=1}^{n} c_j) . d_{r+1}/c_{r+1} \rfloor ,$$

donde $r = \max \{ j \in N \,/\, xp_j = 1 \}$ es una cota inferior del valor de la función objetivo que puede llegar a obtenerse completando esta solución parcial de forma que cumpla la primera restricción. Por lo tanto, si zxp es mayor o igual que el valor de la incumbente habrá que hacer backtracking. Para saber si puede cumplirse la segunda restricción se define, para cada variable j, $sumg_j^+ = \sum_{k>j} \{ g_k / g_k > 0 \}$.

En caso de que ésta no pueda llegar a satisfacerse, habrá que hacer backtracking eliminando de la solución parcial variables, empezando por la última, hasta encontrar una cuya eliminación permita satisfacer la segunda restricción.

Cuando tenga una solución posible cuyo último elemento no sea la última variable x_n, para continuar la exploración se deberá eliminar de la solución únicamente la última variable introducida ya que nada asegura por la ordenación que si se añade una variable posterior no pueda mejorarse el valor de la incumbente. Si, por el contrario, el último elemento de la solución es la variable x_n habrá que eliminar todas las variables de índices sucesivamente correlativos (en orden decreciente) con la variable x_n.

En el caso que el backtracking se realice por la condición de poda de Martello y Toth [MaTo79] se procederá exactamente igual. Se deberá tambien efectuar el mismo retroceso cuando se tenga una subsolución parcial que no cumpla la primera restricción y la siguiente variable candidata x_j tenga un coeficiente c_j negativo. En este caso la inclusión de una o varias de las variables posteriores no llevará a satisfacer esta restricción.

Teniendo en cuenta que se conocen los valores de las cotas $z1_j$ y $z0_j$ asociadas a cada variable, éstas podrán utilizarse en un procedimiento de eliminación de variables, como en el caso de (KP1), fijándolas a o ó a 1 respectivamente cuando el valor de dichas cotas supere el valor de la incumbente. También en este caso, este criterio llevará a reducciones sensibles de las dimensiones del problema que se esté resolviendo, mejorando los criterios clásicos de eliminación.

Suponemos también en este caso que, inicialmente, se dispone de una solución posible obtenida a partir de una heurística. Posteriormente se expondrá una heurística específica para problemas de este tipo especialmente adecuada para problemas que tengan características numéricas similares a las que se dan en las relajaciones lagrangianas del tipo (RL7).

Además, al comenzar la exploración se obtiene una primera solución parcial que será solución posible para el problema (KP2). Se comparará el valor de la función objetivo asociado a la misma con el de la incumbente obtenida mediante la heurística, ya que en el caso en que esta sea mejor se actualizará el valor de la incumbente.

Notación:

n	Número total de variables que intervien en el problema.
$sumg_j^+ = \sum_{k>j} \{g_k / g_k>0\}$	Suma de los coeficientes positivos de la restricción correspondientes a variables con índice posterior a j.
$zxp\_ = \sum_{k=1}^{n} d_k\, xp_k + \lfloor(c_0 - \sum_{l=1}^{n} c_l)\, d_{r+1}/c_{r+1}\rfloor$	Cota inferior del valor de la función objetivo para una subsolución parcial obtenida a partir de xp que satisfaga la primera restricción, siendo r el mayor de los índices de las variables fijadas a 1 en xp.
zxp_j	Cota inferior análoga a zxp_ que se obtendría si se incluyese la variable j en la subsolución parcial xp.
xp	Vector n-dimensional de componentes 0 ó 1, cuyas componentes a 1 indican los índices de las variables que intervienen en la subsolución parcial.
$r1 = \sum_{j=1}^{n} c_j xp_j$	Valor de la primera restricción asociado a una subsolución parcial xp.
$r2 = \sum_{j=1}^{n} g_j xp_j$	Valor de la segunda restricción asociado a una solución parcial xp.
$fp = \sum_{j=1}^{n} d_j xp_j$	Valor de la función objetivo asociado a una solución parcial xp.

Algoritmo

Inicialización

Calcular Incumb — *Valor de la cota superior obtenido mediante una heurística.*

Si incumb $\leq$ zinf **Terminar** — *La solución asociada a la incumbente es óptima.*

Aplicar test de eliminación de variables

Para j=1,n **Hacer** — *Se empieza con la subsolución parcial vacía.*

$xp_j=0$

Fin Para

$r1 \leftarrow 0$

$r2 \leftarrow 0$

$fp \leftarrow 0$

$j \leftarrow 1$

Fin Inicialización

Mientras no fin **Hacer**

Ampliar subsolución parcial xp

Si $r1 \geq c_0$ y — *La solución parcial xp es una subsolución posible mejor que la incumbente.*

$r2 \leq g_0$ y

fp < incumb

Entonces

Actualizar solución incumbente

Hacer backtracking 1

En otro caso si zxp_j $\geq$ Incumb **Entonces** — *La subsolución parcial xp no puede completarse mejorando el valor de la incumbente.*

Hacer backtracking 2

En otro caso si $c_j < 0$ **Entonces** — *No se puede satisfacer la primera restricción.*

Hacer backtracking 2

En otro caso si $r2+g_j+sumg_j^+<g_0$ **Entonces** — *No se puede satisfacer la segunda restricción.*

Hacer backtracking 3

Fin Si

Fin Mientras

Siendo,

Ampliar subsolución parcial xp

Mientras zxp_j < Incumb y

$r1+c_j < c_0$ y

$c_j > 0$ **Hacer**

Añadir elemento j

$j \leftarrow j+1$

Fin Mientras

Fin

Backtracking 1 consiste en eliminar de la solución xp únicamente la última variable añadida a la misma en el caso en que ésta no sea x_n, mientras que, en el caso en que se trate de x_n, deberán eliminarse los últimos elementos correlativos de la solución parcial y la variable inmediatamente anterior a la última correlativa.

Backtracking 2 consiste en eliminar los últimos elementos correlativos añadidos a la subsolución parcial xp y la variable inmediatamente anterior a la última correlativa, y backtracking 3 consiste en eliminar todos los elementos de la subsolución xp con un índice igual o posterior a r, siendo

$$r = \max \{ j \in N \,/\, xp_j = 1 \text{ y } \sum_{k=1}^{r} g_k xp_k + sumg_r^+ \geq g_0 \}$$

el índice de la última variable a partir de la cual puede completarse la subsolución parcial xp satisfaciendo la 2a. restricción.

De nuevo, para mayor claridad en la exposición, en el esquema del algoritmo anterior se supone que en cada paso las cantidades fp,r1 y r2 asociadas a cualquier subsolución parcial xp están debidamente actualizadas.

El algoritmo anterior finalizará, bien cuando se obtenga una incumbente que sea menor o igual que la cota inferior, la solución asociada a la misma será la óptima, bien cuando en alguno de los pasos de retroceso se llegue a una subsolución parcial vacía para la que no exista ninguna variable candidata para completarla.

El test de eliminación de variables es exactamente igual que en el caso del algoritmo para (KP1). Asimismo, este test podría aplicarse cada vez que se actualizase la incumbente, aunque, al igual que para (KP1), no parece que sea rentable en el caso en el que la solución generada por la heurística proporcione una cota superior suficientemente ajustada.

El esquema anterior es igualmente válido para el caso en el que la restricción más vinculante fuese la segunda. Para adaptarlo a esa situación habría que considerar el conjunto N^2= {1,...,n} de índices de las variables ordenados según el criterio de la Proposición 4, así como sustituir todas las condiciones referentes a la primera restricción por una condición análoga respecto a la segunda y viceversa.

IV.4 Casos Particulares: (RL6u) y (RL7u)

En esta sección se estudian los problemas (RL6u) y (RL7u) como casos particulares de los problemas (KP1) y (KP2) estudiados anteriormente.

Sean los problemas

$$(RL6u)\quad \min_{x\in X}\ cx + u(e-Ax)$$
$$nx \leq m$$
$$cx \geq s_0$$
$$x_j \in \{0,1\},\ j\in N$$

y

$$(RL7u)\quad \min_{x\in X}\ cx + u(e-Ax)$$
$$gx \geq g_0$$
$$cx \geq s_0$$
$$x_j \in \{0,1\},\ j\in N$$

Estos problemas pueden formularse como

$$(RL6u)\quad ue + \min_{x\in X}\ (c-uA)x$$
$$nx \leq m$$
$$cx \geq s_0$$
$$x_j \in \{0,1\},\ j\in N$$

y

$$(RL7u)\quad ue + \min_{x\in X}\ (c-uA)x$$
$$gx \geq g_0$$
$$cx \geq s_0$$
$$x_j \in \{0,1\},\ j\in N$$

IV.4.1 Problemas (RL6u)

En (RL6u) los coeficientes de la función objetivo (c-uA) no están restringidos en signo, pero los coeficientes y los términos independientes de las dos restricciones son no negativos. En particular, en el caso de (RL6u), los coeficientes d_j de la función objetivo son los costes reducidos del problema (P) asociados al vector de multiplicadores u; el término independiente de la primera restricción s_0 se corresponde con el número entero inmediatamente superior a la mejor cota inferior conocida para el problema original (P), y los coeficientes de esta restricción son los coeficientes de coste del problema original.

En la segunda restricción, el término independiente indica el número de restricciones del problema original (P), mientras que los coeficientes que intervienen son el número de elementos de cada una de las columnas de la matriz A de restricciones.

En concreto, en la primera restricción los coeficientes serán números naturales y en la segunda números reales no negativos. Los términos independientes son números naturales.

Es evidente que (RL6u) es un caso particular de (KP1), donde, para una mayor generalidad, todos los coeficientes se consideran números reales. Evidentemente, la dificultad en obtener una solución factible para (RL6u) está directamente relacionada con la proporción que exista entre los coeficientes y los términos independientes de estas dos restricciones.

En el caso de la segunda restricción esta proporción vendrá determinada por la densidad de los problemas originales que se intenten resolver, ya que si la densidad del problema (P) es de $d/100 \Rightarrow d/100 \cdot n \cdot m \equiv \sum g_j$ con lo que el valor medio de dichos coeficientes viene a ser aproximadamente $d/100 \cdot m$.

Esto nos indica que en principio es relativamente fácil mantener la factibilidad de la segunda restricción, incluso si estamos interesados en añadir variables que tengan coeficiente de coste d_j negativo en (RL6u), siempre que la densidad de los problemas no sea muy alta, cosa que en la práctica siempre suele suceder.

Con respecto a la primera restricción, para conseguir su factibilidad deberemos añadir un número suficiente de variables a las subsoluciones parciales que se consideren, ya que se trata de una desigualdad de tipo "≥". En este caso, la proporción existente entre los coeficientes de la restricción y el término independiente es a priori más dificil de conocer, debido a que el término independiente no es un dato del problema original.

Resulta, de todos modos, evidente que a medida que aumente la calidad de la mejor cota inferior que se conozca, esta primera restricción será más difícil de satisfacer, sobre todo manteniendo la factibilidad de la segunda restricción, especialmente si se tiene en cuenta que los coeficientes de coste d_j no son sino los costes reducidos asociados al vector de multiplicadores u, con lo que a medida que la calidad de la solución dual u aumente, estos coeficientes tenderán a disminuir para acercarse a las condiciones de holgura complementaria.

A continuación se expone una heurística para resolver problemas del tipo (KP1). Esta heurística está diseñada teniendo en cuenta las características numéricas que aparecen, como se acaba de comentar, en las relajaciones lagrangianas del tipo (RL6u). Por lo tanto, a pesar de que su aplicación es posible para problemas (KP1) más generales, resulta más adecuada para problemas con esta estructura específica que son los que estamos interesados en resolver en este contexto.

Heurística

La heurística procede en cuatro etapas. En la primera de ellas se obtiene una solución posible. Pueden construirse ejemplos en los que ésta fase falle, pero raramente esto será así. De hecho, la calidad de esta primera solución no será normalmente buena, dado que en su obtención no se tienen en cuenta los coeficientes de coste d_j, y la única justificación para utilizarla es asegurar, en la gran mayoría de los casos, la obtención de una solución posible a partir de la cual se irán obteniendo mejoras sucesivas. Además, esta solución tampoco será en general minimal, entendiendo por minimal aquella solución tal que si se eliminase alguna de sus componentes dejaría de serlo.

La segunda etapa de la heurística proporciona una solución minimal contenida en la primera. En particular, la subsolución obtenida será, a menudo, la mejor posible, debido al orden en el que se irán considerando las variables.

En las dos etapas posteriores se mantiene la estructura obtenida en las dos primeras fases, a saber: un conjunto de índices asociado a una solución posible no minimal y una solución minimal (a menudo la mejor) contenida en ella.

Las dos últimas etapas consisten en mejoras que se intentan a base de intercambios. Ello se debe a que el conjunto de variables a partir del cual se ha obtenido la solución no minimal, y por tanto del que se han elegido las variables que forman parte de la solución minimal, no tiene, en absoluto, en cuenta los coeficientes de coste. Por ese motivo, es probable que alguna de las variables con mejor coeficiente de coste, haya quedado fuera del conjunto de variables elegibles y que ahora sea posible intercambiarla por otra que pertenezca a la solución minimal obtenida.

En la primera mejora se intentan intercambios de variables que estén en la solución minimal por variables que estén fuera de la solución no minimal, siempre que el intercambio proporcione una solución posible cuya subsolución minimal asociada mejore el valor de la función objetivo.

Finalmente, en la segunda mejora se intenta substituir variables de la solución minimal por conjuntos de variables no pertenecientes a la solución no minimal. De nuevo, el intercambio se realizará siempre que sea posible obtener una nueva solución posible cuya subsolución minimal asociada mejore el valor de la función objetivo.

La heurística es como sigue:

1-. Construir una primera solución posible.

Sea $N^1 = \{ 1,...,n \}$ el conjunto de índices de variables ordenados por orden creciente de n_j/c_j.

Sean $s_1 = \max \{ j \in N^1 / \sum_{k=1}^{j} g_k \leq g_0 \}$ y $S^1 = \{ j \in N^1 / j \leq s_1 \}$

Se construye el vector x, definido por: $x_j=1, \forall j \in S^1, x_j=0 \ \forall j \in N^1 \backslash S^1$.
Si x no es solución posible para (KP1) Final. La heurística falla.

Teniendo en cuenta que en N^1 aparecen primero aquellas variables cuya relación entre los coeficientes de las dos restricciones es mejor, la heurística fallará en muy pocas ocasiones. De hecho, este procedimiento asegura la obtención de una solución posible no entera para la relajación lineal de (KP1) cuando este problema sea factible.

2-. Encontrar una solución minimal contenida en la anterior.

Considerar el problema (KP1) restringido únicamente a aquellas variables que pertenecen a S^1.

Sea (KP1') $\min \sum_{j\in S^1} d_j x_j$

$$\sum_{j\in S^1} c_j x_j \geq c_0$$

$$\sum_{j\in S^1} g_j x_j \leq g_0$$

$$x_j \in \{0,1\},\ j\in S^1,$$

donde $\forall\, j\in S^1$ $c_j, g_j \in R^+$, $d_j \in R$.

Sea el conjunto $N^2 = \{ 1,\dots,s_1 \}$ de índices de variables pertenecientes a S^1 ordenadas según valores crecientes de d_j/c_j.

Sean $s_2 = \min \{ j\in N^2 / \sum_{k=1}^{j} c_k \geq c_0 \}$ y $S2 = \{ j\in N^2 / j \leq s_2 \}$.

El vector x, definido por $x_j=1$, $\forall j\in S^2$, $x_j=0$ $\forall j\in S^1 \backslash S^2$, es una solución minimal contenida en S^1.

Nótese que un procedimiento análogo proporcionaría el óptimo de la relajación lineal (KP1'_) asociada a (KP1').

3-. Intentar intercambios uno a uno.

Consideremos los siguientes conjuntos de índices:

$S^2 = \{1,\dots,s_2\}$, que ahora suponemos ordenado por orden decreciente de coeficientes de coste d_j,

$S^1 \backslash S^2 = \{ s_2+1,\dots, s_1 \}$ ordenado según orden creciente de coeficientes de coste d_j y

$S^3 = \{ s_1+1,\dots,n \}$ ordenado según valor creciente de los coeficientes de coste d_j.

Mientras no fin **Hacer**

Sean k el siguiente elemento de S^3 y j el siguiente elemento de S^2

Si $d_j \leq d_k$ Entonces

Fin = cierto

En otro caso

Sean $S^{2'} = S^2 \backslash \{j\} \cup \{k\}$ y $S^{1'} = S^1 \backslash \{j\} \cup \{k\}$

Mientras $S^{1'}$ viole la 2ª restricción **Hacer**

$S^{1'} \leftarrow S^{1'} \backslash \{r\}$ siendo r el último elemento de $S^{1'}$

Fin Mientras

Mientras $S^{2'}$ viole la 1ª restricción **Hacer**

$S^{2'} \leftarrow S^{2'} \cup \{r\}$ siendo r el primer elemento de $S^{1'}$

$S^{1'} \leftarrow S^{1'} \backslash \{r\}$

Fin Mientras

Si $S^{2'}$ es el soporte de una solución cuyo

cuyo coste es menor que el de S^2 **Entonces**

$S^2 \leftarrow S^{2'}$

$S1 \leftarrow S^{1'}$

Fin Si

Fin si

Fin Mientras

Este tercer paso favorece la inclusión en la solución de variables que tengan un coeficiente de coste pequeño, y cuya relación entre los coeficientes de las dos restricciones sea adecuada, aún sin ser tan buena como para las variables que forman el soporte S^1.

4-. Intentar intercambios con varias variables a la vez.

Consideremos los siguientes conjuntos de índices:

$S^2 = \{1,...,s_2\}$, que ahora suponemos ordenado por orden decreciente de coeficientes de coste d_j,

$S^1 \backslash S^2 = \{ s_2+1,..., s_1 \}$ ordenado según orden creciente de coeficientes de coste d_j y

$S^3 = \{ s_1+1,...,n \}$ ordenado según valor creciente de los coeficientes de coste d_j.

Mientras no fin **Hacer**

Sean k el siguiente elemento de S^3 y j el siguiente elemento de S^2

Si $d_j \leq dk$ **Entonces**

Fin = cierto

En otro caso

Sea $S^{2'} = S^2 \setminus \{j\} \cup \{k\}$

Mientras $S^{2'}$ viole la 1ª restricción **Hacer**

$S^{2'} \leftarrow S^{2'} \cup \{r\}$ siendo r el siguiente elemento de S^3

$S^3 \leftarrow S^3 \setminus \{r\}$

Fin Mientras

Si $S^{2'}$ es el soporte de una solución cuyo coste es menor que el de S^2 **Entonces**

$S^2 \leftarrow S^{2'}$

Fin Si

Fin si

Fin Mientras

En este cuarto paso, la heurística favorece la inclusión en la solución de conjuntos de variables para las que sean pequeños tanto los coeficientes de coste d_j como los de la primera restricción c_j. Este tipo de variables es, individualmente, poco adecuado para formar parte de alguna solución, debido a que en la ordenación que se hace en el primer paso, aunque el coeficiente g_j sea bastante pequeño, es poco probable que, en el caso en que c_j sea pequeño, el cociente g_j/c_j resulte de los menores independientemente del coste d_j.

Análogamente, en el tercer paso la inclusión de este tipo de variables puede resultar difícil, ya que para ello será necesario añadir alguna de las que, a pesar de tener un coeficiente adecuado en la 1ª restricción, se sabe que tienen un coeficiente de coste d_j poco prometedor.

Este no será el caso en este cuarto paso en el que, al considerar a la vez un conjunto de variables de este tipo, el intercambio con alguna variable cuyo coeficiente de coste no sea suficientemente pequeño puede realizarse fácilmente.

Finalmente y mientras se produzca alguna mejora, se añadirán a la solución todas aquellas variables que, no perteneciendo a la misma, tengan coeficiente de coste negativo y cuya inclusión en la solución no haga violar la segunda restricción, reduciendo posteriormente la nueva solución obtenida a una solución minimal.

La complejidad del peor caso de la heurística que se ha expuesto es de $O(n^3)$. En concreto, esta cota se corresponde con el coste del tercer paso en el que, en el caso peor, se intentarán intercambiar todas las variables de S^2 por todas las de S^3, y cada uno de estos intercambios puede dar lugar a ajustes en S^1, S^2 y S^3 de coste O(n). En los dos primeros pasos el coste será como máximo O(nlogn), ya que se realiza una ordenación y un recorrido secuencial. En el cuarto paso se realiza un recorrido de orden $O(n^2)$ ya que se estudian los intercambios entre S^2 y S^3 (sin analizar posibles reajustes).

IV.4.2 Problemas (RL7u)

En el caso de (RL7u) los coeficientes d_j de la función objetivo no están restringidos en signo ya que, de nuevo, se trata de los costes reducidos para el problema (P) asociados al vector de multiplicadores u. Tampoco están restringidos en signo los coeficientes de la segunda restricción, siendo, en cambio, los coeficientes de la primera números reales no negativos. En particular, la interpretación de la primera restricción es es exactamente la misma que en el caso de (RL6u).

La segunda restricción es, como ya se ha visto en el capítulo anterior, la subrogada obtenida por la aplicación del procedimiento de Dyer. Lo único que puede asegurarse sobre esta segunda restricción es que, debido al procedimiento utilizado para su obtención, tanto el término independiente como los coeficientes que en ella aparecen serán de una magnitud bastante reducida, estando además todos los coeficientes normalizados respecto a la norma euclídea.

Se trata, por tanto, de un problema que tiene una estructura similar a la de los problemas del tipo (KP2') que ya se ha visto pueden transformarse en los de la clase (KP2) mediante un cambio de variable apropiado. En este caso haciendo $y_j = 1\text{-}x_j$ $\forall j$ tal que el coeficiente de coste $(c\text{-}uA)_j$ sea negativo, se convierte (RL7u) en un problema equivalente en el que los coeficientes de la función objetivo son todos no negativos. Resulta por tanto evidente que (RL7u) es un caso particular de (KP2).

Debido a la estructura de estos problemas, es difícil conocer a priori cual de las dos restricciones resulta más difícil de satisfacer, pudiendo únicamente asegurar que, como en el caso de (RL6u) a medida que aumente la calidad de la mejor cota inferior conocida para el problema (P) la primera restricción de (RL7u) resultará más difícil de satisfacer. En cualquier caso, teniendo en cuenta la diferencia de magnitud que aparece en los coeficientes y términos independientes de la primera restricción respecto de los de la segunda, cualquier procedimiento para encontrar una solución posible deberá tener este hecho en cuenta y considerar algún tipo de ordenación que refleje la magnitud de los coeficientes relativamente a la influencia de la restricción en la que intervienen.

Para satisfacer la 1ª restricción interesaría ordenar las variables según orden creciente de d_j/c_j. Análogamente, para la 2ª restricción habría que hacerlo por orden creciente de d_j/g_j. Ahora bién, para que las dos restricciones tengan una ponderación similar habría que normalizar ambos términos independientes con lo que las ordenaciones resultarían en función de los cocientes d_jc_0/c_j y d_jg_0/g_j respectivamente. Por lo tanto, una posible ordenación para resolver el problema con dos restricciones es aquella que resulta de la suma de los dos cocientes así definidos; de esta forma se estarían potenciando de igual manera las dos restricciones.

Ahora bien, teniendo en cuenta que las dos restricciones no resultan igualmente 'vinculantes' en el sentido de que una de ellas es más fácil de satisfacer que la otra, resulta razonable sugerir una ordenación que potencie conseguir una solución que satisfaga la restricción más 'difícil' sin llegar a penalizar la que resulte más 'fácil'. Esto puede conseguirse multiplicando cada uno de los cocientes anteriores por un factor μ_i que sea tanto mayor en la medida en que la restricción a la que va asociado resulte más 'difícil'. Teniendo en cuenta que las cotas inferiores obtenidas para los problemas del tipo (KP2) reflejan la dificultad para satisfacer cada una de las restrciciones, ya que estas cotas aumentan a medida que la restricción a la que están asociadas resulta más difícil de satisfacer, parece adecuado que los factores μ_i guarden proporción con dichas cotas.

A continuación se expone una heurística para problemas (KP2) que tiene en cuenta las consideraciones anteriores. Debe comentarse, una vez más, que se trata de un procedimiento que puede aplicarse a problemas más generales que las relajaciones lagrangianas (RL7u), aunque su diseño se ha hecho en función de las características numéricas de este tipo de problemas.

Heurística

Sea N ={ 1,...,n } el conjunto de los índices de las variables ordenados según valores crecientes de $\mu_1 d_j.c_0/c_j + \mu_2 d_j g_0/g_j$, siendo μ_1 y μ_2 tales que $\mu_1+\mu_2=1$ y $\mu_1/\mu_2=z_{inf}^1/z_{inf}^2$.

Sea $s = \min \{ j \in N \,/\, \sum_{k=1}^{j} c_j \geq c_0$ y $\sum_{k=1}^{j} g_j \geq g_0 \}$

Sea el vector x, definido por: $x_j=1, \forall j \leq s$, $x_j= 0\ \forall j>s$. Si x no es solución posible para (KP2), la heurística falla.

Hay que tener en cuenta que la heurística anterior puede, teóricamente, fallar, ya que al existir coeficientes negativos, tanto en la 1ª cómo en la 2ª restricción, el procedimiento anterior no asegura la obtención de una solución posible. De todos modos, la ordenación que se utiliza favorece la obtención de una solución posible, ya que aparecen, casi seguramente, primero las variables que tienen coeficiente positivo en las dos restricciones y después, las que tengan un coeficiente positivo y uno negativo. Hay que resaltar que esta heurística no ha fallado en ninguno de los problemas en los que se ha utilizado.

La complejidad del peor caso para esta heurística es de orden O(nlogn) que se corresponde a la ordenación previa al recorrido secuencial que se realiza en la misma.

CAPÍTULO V

UNA CLASE DE ALGORITMOS HÍBRIDOS PARA PROBLEMAS DE (SP) Y EXPERIENCIAS COMPUTACIONALES

En el comentario final del primer capítulo ya se ha indicado que la tendencia actual en el diseño de algoritmos para los problemas enteros de carácter combinatorio apunta a la construcción de algoritmos híbridos que utilicen de forma conjunta distintos procedimientos aplicables a un mismo problema. De esta forma se potencia la obtención del mejor rendimiento de cada uno de los métodos utilizados y la inclusión de otros procedimientos proporciona herramientas que reducen los inconvenientes individuales de cada uno de ellos.

En los capítulos anteriores se han estudiado procedimientos ya existentes y se ha propuesto la utilización de algunos nuevos para la resolución de los problemas de set partitioning. Teniendo en cuenta que este trabajo se enmarca dentro de una línea de diseño algorítmico de carácter híbrido, resulta natural recoger y sintetizar los distintos procedimientos que se han propuesto para combinarlos en un único algoritmo enfocado a la resolución de los problemas de set partitioning.

V.1 Algoritmo híbrido para problemas de (SP)

Los métodos que se han analizado pueden clasificarse en los siguientes tipos: métodos heurísticos de obtención de soluciones posibles, obtención de desigualdades válidas y métodos refuerzo dual. Cada uno de éstos métodos resulta en sí mismo de gran interés ya que parcialmente contribuye a la resolución del problema original. En primer lugar, los métodos heurísticos producen soluciones posibles primales que proporcionan cotas superiores del valor del óptimo; los planos secantes reducen el conjunto de soluciones posibles y los métodos duales ayudan a reducir el gap de dualidad. Ahora bien, cada uno de ellos presenta asimismo algún inconveniente que deseamos superar: cómo mejorar la calidad de una solución posible obtenida mediante una heurística, cómo utilizar la información proporcionada por una desigualdad válida o cómo conocer que la calidad de la cota inferior que se tiene es suficientemente buena si no se dispone de una cota superior para compararla.

Evidentemente, para poder mejorar la calidad de una solución primal obtenida mediante una heurística puede resultar fundamental disponer de un procedimiento que permita reducir el espacio de búsqueda de soluciones posibles; de ahí el interés en un procedimiento de generación de planos secantes que pueda combinarse con la heurística. Recíprocamente para obtener un mejor rendimiento de la información proporcionada por un plano secante resulta básico disponer de algún procecimiento heurístico que proporcione nuevas soluciones obtenidas en el espacio de búsqueda más restringido que define la desigualdad. Por lo tanto será necesario combinar el procedimiento de generación de desigualdades con alguna heurística que proporcione soluciones posibles para el problema resultante. Asimismo la mejor solución obtenida en la combinación de los métodos anteriores puede resultar un buen punto de referencia para evaluar la calidad de las cotas inferiores obtenidas mediante procedimientos duales.

Además un esquema algorítmico para resolver (SP), en el que se utilicen heurísticas para obtener soluciones posibles y a partir de ellas se deriven desigualdades básicas, elimina uno de los principales inconvenientes de los métodos clásicos de generación de planos secantes a partir de soluciones fraccionales obtenidas de tablas del Simplex; a saber: la utilización de la relajación lineal ordinaria que, como ya se ha comentado en I.5, supone un obstáculo en la resolución de los problemas de (SP) ya que éstos a menudo presentan problemas de degeneración. Además, en términos de eficiencia, los métodos heurísticos que se proponen resultan computacionalmente rentables debido a su sencillez.

Teniendo en cuenta las consideraciones anteriores, la clase de algoritmos que se proponen se ajustan al siguiente esquema iterativo básico:

Inicialización k=0

Mientras no fin **Hacer**

Sea (SPA_k) el problema actual

Obtener una solución posible (u,v) para el problema lineal dual asociado a (SPA_k).

Obtener una solución primal x asociada a u para el problema (SPA_k).

Aplicar test de eliminación de variables

Generar una desigualdad válida $\pi x \geq 1$

Incorporar la desigualdad válida (SPA_k).

$k \leftarrow k+1$

Fin mientras

El algoritmo anterior terminará cuando se demuestre que se ha encontrado el óptimo del problema o bien cuando falle el procedimiento para encontrar una solución posible primal. Podremos asegurar que se ha resuelto el problema al óptimo cuando el gap de dualidad que exista entre el valor de la mejor solución posible primal y la mejor cota inferior obtenida mediante procedimientos duales sea suficientemente reducido.

Siguiendo la notación introducida en el capítulo II, en una iteración k el problema (SPA_k) será de la forma

$$(SPA_k) \quad \begin{array}{l} \min cx \\ Ax=e_m \\ Gx \geq e_k \\ x_j \in \{0,1\}, \ \forall j \in N, \end{array}$$

donde $c \in Z^{n+}$, A es una matriz de dimensión mxn y G es una matriz de dimensión rxn cuyos elementos son únicamente 0 ó 1, y e_m y e_k son , respectivamente, los vectores m-dimensional y k-dimensional formados por todo 1.

Su problema lineal dual asociado es por lo tanto:

$$(DA_k) \quad \begin{array}{l} \max ue_m+ve_k \\ uA+vG \leq c \\ v \geq 0. \end{array}$$

Por consiguiente en la iteración inicial, k=0, el problema (SPA_k) es un problema de minimización de set partitioning puro, puesto que el número de filas de la matriz G (número de desigualdades válidas que se han generado) es 0. En las iteraciones posteriores, al incorporar la desigualdad válida del tipo $\pi x \geq 1$, la estructura de (SPA_k) es la de los problemas de minimización de set partitioning ampliados que se han estudiado en II.4.

V.1.1 Obtención de un par soluciones posibles primal y dual

La obtención de una solución posible para el problema lineal dual (DA_k) asociado a (SPA_k) puede realizarse mediante dos tipos de procedimientos diferentes. Bien por procedimientos heurísticos (la heurística dual propuesta en II.2 en la primera iteración y la heurística dual propuesta en II.4 en las posteriores), o bien mediante un procedimiento de tipo subgradiente aplicado a una relajación lagrangiana. La utilización de los procedimientos heurísticos presenta la ventaja de que computacionalmente resultan muy poco costosos, pero, sin embargo, la calidad de la solución dual obtenida no está garantizada. La utilización de un procedimiento de tipo subgradiente resulta computacionalmente más costosa, pero garantiza la calidad de la solución dual que se obtiene. Sin embargo, si la relajación lagrangiana que se utiliza en el segundo caso es un problema que tiene una estructura sencilla de resolver, el esfuerzo computacional resulta sin lugar a dudas rentable. La versión del algoritmo con la que se han realizado las experiencias computacionales cuyos resultados se expondrán posteriormente utiliza relajación lagrangiana en la primera iteración y, posteriormente, en cualquier iteración inmediatamente posterior a haber mejorado la calidad de la cota superior. Para justificar la decisión anterior se expondrán resultados comparativos sobre la calidad de las cotas inferiores obtenidas con los distintos métodos.

En la primera iteración se utiliza la relajación lagrangiana ordinaria dada por

$$(RL1u) \qquad \min_{x \in X} cx + u(e-Ax)$$

donde $X = \{x \in R^n \, / \, x_j \in \{0,1\}, \, j \in N \}$.

En las iteraciones posteriores, teniendo en cuenta que el criterio para su utilización es el de haber mejorado el valor de la mejor cota superior, mantendremos como restricción explícita la última desigualdad válida generada antes de la obtención de la solución posible primal que ha proporcionado la mejora; es decir utilizaremos la relajación lagrangiana dada por:

$$(RL9u,v) \qquad \min_{x \in X} cx + u(e_m-Ax)+ v(e_k-Gx)$$
$$\pi x \geq 1$$

donde $\pi x \geq 1$ es la desigualdad que se acaba de mencionar y, cometiendo un abuso de notación, estamos suponiendo que la matriz G está formada por todas las desigualdades obtenidas anteriormente a la misma así como por la última desigualdad válida generarda inmediatamente después de haber mejorado la cota superior.

Cabe resaltar que la estructura de las dos relajaciones lagrangianas que se proponen resulta muy sencilla; las soluciones para (RL1u) son inmediatas y para (RL9uv) se obtienen muy fácilmente como se ha expuesto en III.2.2. Por ese motivo es reducido el esfuerzo computacional para resolver los problemas duales asociados a dichas relajaciones utilizando un procedimiento de tipo subgradiente y su utilización resulta rentable.

Una vez que se disponga de una solución dual para el problema actual (SPA_k), se obtendrá una solución posible primal asociada a la misma. Para la obtención de esta nueva solución primal se utilizarán los procedimientos heurísticos propuestos en el capítulo II: la heurística primal de II.2 en la primera iteración y la heurística primal de II.4 en las iteraciones posteriores. Como ya se ha comentado repetidas veces en el capítulo II, el criterio en que que se basan estos procedimientos es el sugrerido por Fisher y Kedia [FiKe86] en su heurística para problemas de maximización de set partitioning; es decir, alcanzar en la medida de lo posible las condiciones de holgura complementaria para el par de soluciones primal-dual.

V.1.2 Test de eliminación de variables y obtención de desigualdades válidas.

Posteriormente el algoritmo aplica un test de eliminación de variables y genera una desigualdad válida del tipo $\pi x \geq 1$. El test de eliminación de variables se deriva del caso particular que resulta del Corolario 1 en II.3 cuando la disyunción obtenida consta de un único término. Las desigualdades $\pi x \geq 1$ se obtienen gracias a la aplicación del Teorema 2 en II.3 y su validez para los problemas (SPA_k) ha quedado probada.

En ambos casos, las condiciones necesarias para asegurar la validez de los resultados es necesario disponer de un par de soluciones posibles primal y dual respectivamente tales que la suma de los costes reducidos de las variables que forman parte del soporte de la solución sea mayor que el gap de dualidad existente entre el valor de la mejor solución posible primal conocida y el de la solución dual actual. En concreto, si z_{sup} es el valor de la mejor solución posible primal conocida y $(\hat{x},(u,v))$ es el par de soluciones posibles actuales para los problemas primal y dual respectivamente, la condición necesaria para la aplicación del Corolario 1 y del Teorema 2 es

(1) $\sum_{j\in S(x)} s_j \geq z_{sup} - (ue_m+ve_k)$, siendo $S(x) = \{j\in N \,/\, x_j=1\}$ y $s = c-(uA-vG)$.

El Teorema 4 de II.3 proporciona una condición necesaria para que se cumpla la condición anterior. Es decir, este teorema asegura bajo qué condiciones es posible derivar disyunciones válidas así como obtener desigualdades a partir de cotas condicionales. La respuesta al mismo viene dada en términos de condiciones adicionales para el par de soluciones primal-dual. En particular, demuestra que si se cumple

(2) $u(e_m-Ax)+v(e_k-Gx)=0$

entonces también se cumple la condición (1).

La condición (2) siempre se cumple en el caso de un problema de set partitioning puro (k=0) puesto que se convierte en $u(e_m-Ax)=0$, que debe satisfacerse para cualquier solución posible primal independientemente del vector u. No ocurre lo mismo, sin embargo, en el caso de un problema ampliado (SPA_k) con $k\neq 0$, aunque si puede lograrse esta condición modificando ligeramente la solución dual que se tenga.

Efectivamente, sean (u,v) la solución dual actual para (DA_k) y x la solución primal asociada. Sean $K=\{1,...,k\}$ el conjunto de índices de la matriz G y $K(x)=\{l\in K \,/\, Gx>1\}$. Teniendo en cuenta que

$$u(e_m-Ax) + v(e_k-Gx) = \sum_{l\in K(x)} v_l\,(e_k-Gx)^l$$

Si la condición (2) no se cumple, haciendo 0 sucesivamente las componentes v_l, $l\in K(x)$, como máximo será necesario hacer todas ellas 0 para conseguir que se satisfaga esta condición.

Por lo tanto dado un par de soluciones primal y dual respectivamente, para obtener disyunciones y desigualdades válidas será necesario verificar si se cumple la condición (1). En caso que ésta no se cumpla, se irán anulando sucesivamente las componentes v_l, $l\in K(x)$ hasta satisfacer la condición (2).

En estas condiciones, el Corolario 1 asegura la validez de disyunciones del tipo

$$\bigvee_{i=1}^{p} (x_j=0,\ j\in Q_i)$$

En nuestro contexto algorítmico, el caso particular que se obtiene cuando la disyunción obtenida consta de un único término resulta de gran interés. En este caso será posible fijar a 0 todas las variables que intervienen en el único término de la disyunción.

Recordando de nuevo este caso particular, cuando la disyunción consta de un único término ésta se convierte en

$(x_j=0, j\in Q_1)$, donde el conjunto $Q_1 = \{j\in N \,/\, s_j \geq zsup-(ue_m+ue_k)\}$.

Por consiguiente, dado un par de soluciones posibles primal y dual que satisfagan las condiciones de aplicación del Corolario 1 si $Q_1\neq\emptyset$, podremos fijar a 0 (y por tanto eliminar del problema todas las variables de dicho conjunto. Hay que resaltar que las variables que intervienen en Q_1, son aquellas para las que el coste reducido es mayor o igual que el gap de dualidad existente entre el valor de la mejor solución posible primal conocida y el valor de la solución dual actual.

Posteriormente, y teniendo en cuenta que ya está verificada la condición (1) el algoritmo generará una desigualdad válida del tipo $\pi x\geq 1$ mediante el procedimiento expuesto en II.3.

V.1.3 Tests lógicos de eliminación

La eficiencia del esquema básico algorítmico expuesto al principio de esta sección puede mejorarse incluyendo en él la utilización de los tests lógicos de eliminación de variables que se han expuesto en I.4. La utilización de estos procedimientos es una fase previa a la resolución de los problemas puede incidir favorablemente en el desarrollo posterior del algoritmo y el esfuerzo computacional que requiere su utilización resulta reducido. Ello resulta evidente, puesto que estos tests, al eliminar variables del problema fijándolas a 0 o 1, pueden proporcionar nuevos problemas de dimensiones más reducidas en los que la aplicación de los procedimientos antes descritos resulte más eficaz.

Asimismo, cuando el test de eliminación de variables que se deriva de las disyunciones a partir de cotas condicionales expuesto en el apartado anterior fije alguna de las variables a 0, la aplicación de estos tests lógicos de eliminación puede producir eliminaciones adicionales que se deriven de la estructura del nuevo problema resultante.

V.1.4 Procedimientos de reducción del gap de dualidad

Uno de los criterios de terminación del algoritmo básico propuesto es el fallo del procedimiento de búsqueda de soluciones posibles primales. Si esto ocurre y si la calidad de la mejor cota inferior que se tiene no resulta suficientemente buena, no disponemos de ninguna herramienta que permita evaluar la calidad de la mejor solución posible primal obtenida. En esta situación , estamos especialmente interesados en reforzar la calidad de la cota inferior y poder reducir el gap de dualidad al máximo.

Hay que resaltar que en algoritmo propuesto esto resulta de especial importancia debido a que el fallo de los procedimientos primales puede deberse a que el problema (SPA_k) actual no tenga soluciones posibles. Recordemos, una vez más, el significado de las desigualdades válidas $\pi x \geq 1$ obtenidas: dada una cota superior z_{sup} cualquier solución posible primal mejor que la que ha proporcionado dicha cota deberá satisfacer la desigualdad. Ahora bien, como ya se ha comentado anteriormente, en el caso en el que la solución que haya proporcionado z_{sup} sea óptima es posible que al obtener una desigualdad válida e incorporarla al problema (SPA_k), el nuevo problema obtenido no tenga soluciones posibles, puesto que ya no existen soluciones mejores que la que ha proporcionado z_{sup}.

Por lo tanto, cuando la heurística primal falle intentaremos reducir el gap de dualidad existente mediante alguno de los procedimientos duales estudiados en el capítulo III. En concreto los distintos procedimientos que se han utilizado para las experiencias computacionales son los siguientes:

1- Procedimiento para la obtención de una restricción subrogada asociada al conjunto actual de restricciones de (SPA_k) aplicando el método de Dyer [Dye80] como se ha expuesto en III.4.2.

2- Utilización de la variante propuesta del método BISA de refuerzo dual a la relajación lagrangiana de (SPA_k) dada por

$$(RL7u) \qquad \min_{x \in X} cx + u(e_m - Ax) + v(e_k - Gx)$$
$$gx \geq g_0$$
$$cx \geq s_0$$

donde $gx \geq g_0$ es la restricción subrogada obtenida en la aplicación del procedimiento de Dyer. La restricción $cx \geq s_0$ impone que las soluciones posibles tengan un valor de la función objetivo original mayor o igual que la mejor cota inferior conocida, puesto que s_0 es el entero inmediatamente superior al valor de la mejor cota inferior conocida.

3- Utilización de la variante propuesta del método BISA [Barci85a, Barci85b] de refuerzo dual a la relajación lagrangiana de (SPA_k) dada por

$$(RL6u) \qquad \min_{x \in X} cx + u(e_m - Ax) + v(e_k - Gx)$$
$$nx \leq m$$
$$cx \geq s_0$$

donde $nx \leq m$ es una restricción implicada por el conjunto original de restricciones $Ax=e$ cuyo significado se ha analizado en III.2.2, en la que las componentes del vector n indican el número de elementos de la j-ésima columna de la matriz A y m es el número de filas de la matriz A y $cx \geq s_0$ tiene el mismo significado que en el caso de (RL7u).

Para la resolución de los problemas duales asociados a las relajaciones lagrangianas (RL6u) y (RL7u) utilizaremos optimización subgradiente y para resolver los problemas en las iteraciones internas los procedimientos descritos en el capítulo IV para los problemas de knapsack con dos restricciones ya sean del mismo sentido o de sentido contrario.

Estos procedimientos pueden también aplicarse al problema (SP) original antes de comenzar el proceso iterativo del algoritmo híbrido. Ello nos permitirá, eventualmente, la obtención de la solución óptima y, en cualquier caso, la cota inferior que proporcionen será un buen sistema de referencia para evaluar la calidad de las soluciones primales que se obtengan posteriormente.

V.1.5 Una clase de algoritmos híbridos para problemas de (SP)

Las observaciones y comentarios apuntados en los apartados anteriores han definido de una forma más precisa los distintos procedimientos que debe incluir la clase de algoritmos híbridos que se propone para resolver los problemas de (SP).

El esquema algorítmico al que deben ajustarse es, por lo tanto, el siguiente:

Inicialización
  $k=0$
  Fin = falso
  Aplicar tests lógicos de eliminación de variables a (SPA_k)
  Aplicar algún procedimiento dual a (SPA_k)
  Si se ha obtenido una solución posible primal **Entonces**
    Fin = cierto
  En otro caso
    Sea z_{inf} el valor de la cota inferior obtenida
    $zsup = \infty$
  Fin si
Fin Inicialización

Mientras no fin **Hacer**

Sea (SPA_k) el problema actual

Obtener una solución posible (u,v) para el problema lineal dual asociado a (SPA_k).

Sea z_l el valor de la cota inferior asociado a (u,v)

Si $z_l > z_{inf}$ **Entonces** zinf $\leftarrow$ zl

Obtener una solución primal x asociada a u para el problema (SPA_k).

Si la heurística falla **Entonces**

Aplicar algún procedimiento dual a (SPA_k)

Fin = cierto

Fin si

Sean z_u el valor de la cota superior asociado a x

S(x) el soporte de la solución posible x

s = c-uA-vG el vector de costes reducidos asociado a (u,v)

Si $z_u < z_{sup}$ **Entonces** $z_{sup} \leftarrow z_u$

Si $z_{sup}-z_{inf} \leq 1$ **Entonces** Fin = cierto

Si $\sum_{j\in S(x)} s_j < z_{sup}-z_l$ **Entonces**

Modificar la solución dual (u,v) hasta cumplir $u(e_m-Ax)+ v(e_k-Gx) = 0$

Actualizar el vector s de costes reducidos

Actualizar el valor z_l asociado a la solución dual

Fin si

Para j=1,n **Hacer**

Si $s_j \geq z_{sup} - z_l$ **Entonces**

Hacer x_j=0

Aplicar Tests lógicos de eliminación de variables

Fin Si

Fin Para

Generar una desigualdad válida $\pi x \geq 1$

Incorporar la desigualdad válida (SPA_k).

$k \leftarrow k+1$

Fin mientras

V.2. Procedimientos de tipo subgradiente y Resolución de Problemas de Knapsack

En esta sección se exponen las técnicas que se han utilizado en la resolución de los problemas duales que resultan de las distintas relajaciones lagrangianas formuladas en alguno de los pasos del algoritmo, así como los métodos utilizados en la resolución de los problemas clásicos de knapsack que aparecen tanto en dichas relajaciones lagrangianas como en la aplicación del método de Dyer [Dye80] para la obtención de restricciones subrogadas.

V.2.1. Procedimientos de tipo subgradiente.

Existen distintos trabajos sobre las técnicas que pueden utilizarse para la obtención de los multiplicadores duales óptimos en el contexto de dualidad lagrangiana [HeKa71, HeWoCr74, Mar75, MaHoBl75, BaGo79, Fis81]. Actualmente, sin lugar a dudas, los procedimientos de tipo subgradiente son los que se utilizan de forma generalizada.

La aplicación de estos procedimientos que, son de tipo iterativo, a los problemas duales asociados a las relajaciones lagrangianas proporciona soluciones cuasi-óptimas para dichos problemas con un coste computacional que depende de la estrucura de los subproblemas que aparecen en las iteraciones del procedimiento.

Inicialmente fueron Held y Karp [HeKa71] quienes utilizaron un procedimiento de tipo subgradiente aplicado a problemas de travelling salesman y, porteriormente, Held,Wolfe y Crowder [HeWoCr74] estudiaron la validez de procedimientos análogos aplicados a distintos tipos de problemas.

En el contexto de los problemas ampliados de set partitioning (SPAk) se desea resolver o aproximar el problema dual asociado a una relajación lagrangiana dado por

$$\max_{u\in F}\ \min_{x\in G}\ L(x,u,v) = \max_{u\in G}\ \{ue_m+ve_k + \min_{x\in G}\ (c\text{-}uA\text{-}vG)\,x\} \qquad (1)$$

donde F y G son las distintas relajaciones de los conjuntos de soluciones posibles para (DAk) y (SPk) respectivamente. Es decir:

$F \supseteq \{\ (u,v)\ /\ v\geq 0,\ uA+vG\leq c\ \}$ y $G \supseteq \{\ x\ /\ Ax=e_m,\ Gx\geq e_k\}$

Dado u_0, el algoritmo de Held Wolfe y Crowder [HeWoCr74] genera una sucesión finita de soluciones de (1) $\{(u^t,v^t)\}$, tal que para todo t se satisface:

. $(u^t,v^t) \in F$
. $(u^{t+1}, v^{t+1}) = P_F((u^t,v^t) + p^t d^t)$
. $L(x, (u^{t+1},v^{t+1})) > L(x, (u^t,v^{t+1}))$

donde d_t es un vector subgradiente (vector de dirección posible de ascenso) en (u^t,v^t), p^t es la longitud de paso y $P_F(.)$ indica la proyección en el conjunto F.

En cada iteración se resuelve el problema

$$\min_{x \in G} L(x,(u^t,v^t)) = u_t e_m + v_t e_k + \min_{x \in G} (c-u^tA-v^tG)x \qquad (2)$$

Sea $x^t=x^t(u^t,v^t)$ una solución óptima para (2), entonces el vector subgradiente d^t viene dado por $d^t = (u^t-Ax^t, v^t-Gx^t)$ y se toma como longitud de paso

$$p_t = \frac{\lambda_t [w^* - L(x^t, (u^t,v^t))]}{\| d_t \|^2}$$

siendo $\forall t$ los escalares λ_t tales que $\varepsilon \leq \lambda_t \leq 2$, para un ε dado y w^* una cota superior del valor de la función objetivo del problema original.

La convergencia de los procedimientos de tipo subgradiente está ampliamente estudiada en un trabajo de Goffin [Gof77] y está demostrada [Pol67,Pol69] para cualquier sucesión $\{\lambda_t\}$ tal que:

. $\lambda_t \rightarrow 0$
. $\Sigma \lambda_t = \infty$

Por lo tanto, en las versiones de este procedimiento que se han implementado, la elección de estos parámetros se ha tomado [BaHo80] como $\lambda_0=2$ y posteriormente haciendo $\lambda_t=\lambda_{t-1}/2$ después de cada T iteraciones, siendo T=max{T/2,5}. Inicialmente T=m donde m es el número de restricciones del problema original.

Uno de los principales inconvenientes que presentan los procedimientos de tipo subgradiente es el comportamiento errático aparece, en ocasiones, en las primeras iteraciones. Cuando esto ocurre, es necesario realizar un mayor número de iteraciones para alcanzar la convergencia del algoritmo. Para intentar solucionar este obstáculo, se ha estudiado la utilización de vectores obtenidos como modificación de los subgradientes d^t antes definidos.

Uno de los trabajos en los que se presentan alternativas en este sentido se debe a Camerini, Fratta y Maffioli [CaFrMa74] y en él se propone la utilización como posible dirección de ascenso en cada iteración de un vector s^t de la forma $s^t=d^t+\beta_t s^{t-1}$ donde d^t es el vector subgradiente definido anteriormente y β_t son unos escalares definidos por

$$\beta_t = \begin{cases} -\gamma \dfrac{s^{t-1}d^t}{\|s^{t-1}\|^2} & \text{si } s^{t-1}d^t < 0, \\ 0 & \text{en otro caso.} \end{cases}$$

Inicialmente se elige $s^{t-1}=0$ para $t=0$.

La elección del escalar g es arbitraria, aunque los autores sugieren la utilización de $\gamma=1.5$ que es una aproximación del valor $\sqrt{2}$ que, cuando $s^{t-1}d^{t-1}<0$, proporciona el valor máximo del cociente $\cos \delta_s / \cos \delta_d$ donde δ_s y δ_d son , respectivamente, los ángulos que forman los vectores s^t y d^t con la dirección óptima $w^*-L(x,(u^t,v^t))$. La elección de $\gamma=0$ supondría la elección de la dirección subgradiente d^t, mientras que $\gamma=1$ supondría elegir una dirección ortogonal a s^{t-1}.

Tomando, en cada iteración la longitud de paso como

$$p^t = \frac{w^*-L(x^t,(u^t,v^t))}{\|s^t\|^2}$$

los autores demuestran la convergencia de este algoritmo.

Los problemas de set partitioning no son una excepción respecto de los problemas que presentan un comportamiento errático en las primeras iteraciones. Por ese motivo, para resolver los problemas duales asociados a las distintas relajaciones lagrangianas, también se ha implementado una versión en la que se utiliza un procedimiento de subgradiente modificado como el propuesto en [CaFrMa74] que se acaba de exponer. Las experiencia computacionales realizadas con estas nuevas versiones no han presentado en general una convergencia más rápida y menos errática que las que se han realizado con la versión clásica de optimización subgradiente. Sin embargo, sí se ha detectado que en casi todos los casos, las cotas inferiores obtenidas al final de los procedimientos eran, la mayoría de las veces ligeramente peores que las obtenidas mediante los procedimientos clásicos. Estos resultados nos han llevado a utilizar el método clásico en la versión definitiva que con la que se han realizado las experiencias computacionales que se expondrán posteriormente.

V.2.2. Resolución de Problemas de Knapsack

En la aplicación del procedimiento de Dyer [Dye80] para la obtencipón de restricciones subrogadas y en algunas de las relajaciones lagrangianas (RL3u), (RL4u), (RL5u) (cf. 3.4.2), que se utilizan para la obtención de cotas inferiores para los problemas, aparecen problemas de minimización de tipo knapsack.

Estos problemas han sido ampliamente estudiados [MaTo79, GiGo61] y existen implementaciones muy eficientes de distintos algoritmos que los resuelven.

La formulación general que consideran los algoritmos que se acaban de mencionar es de la forma

$$
\text{(KP)} \quad \begin{array}{l} \min\ px \\ wx \geq w_0 \\ x_j \in \{0,1\},\ j \in N. \end{array}
$$

donde p, $w \in Z^{n+}$ y $w_0 \in Z^+$.

Los problemas de este tipo que aparecen en los distintos procedimientos que ya se han mencionado, se ajustan con ligeras variantes a esta formulación general, puesto que son de la forma:

$$
\text{(KPR)} \quad \begin{array}{l} \min\ px \\ wx \geq w_0 \\ x_j \in \{0,1\},\ j \in N. \end{array}
$$

donde p, $w \in R^n$ y $w_0 \in R$.

La aparición de coeficientes con distintos signos, tanto en la función objetivo como en la restricción no supone ningún obstáculo para la utilización de los algoritmos clásicos [MaTo79,GiGo61] que se han mencionado puesto que podemos fácilmente obtener un problema equivalente en el que todos los coeficientes sean no negativos. En particular, supongamos que

$\exists j \in N$ t.q. $p_j<0$; entonces,

Si $w_j>0$	Fijar $x_j=1$,	(cualquier solución óptima deberá contener esta variable)
	Hacer $w_0 \leftarrow w_0 - w_j$	(actualizar el término independiente de la restricción)
	Eliminar la variable j	
Si $w_j<0$	Hacer $\overline{x}_j \leftarrow 1-x_j$	(La nueva variable tendrá coeficiente positivo tanto en la función objetivo como en la restricción)

Si $\exists j \in N$ t.q. $p_j>0$ y $w_j<0$ podemos fijar $x_j=0$ puesto que las soluciones óptimas del problema no contendrán esta variable.

La otra diferencia que aparece entre las formulaciones de los problemas de knapsack que se presentan en nuestro contexto y la de los problemas (KP) es que los coeficientes son números reales. En este caso, podemos realizar una consideraciones similares a las que se han realizado para justificar la conveniencia de algoritmos que consideren el carácter real de los distintos coeficientes en el caso de problemas de knapsack con 2 restricciones. Por un lado, a pesar de que siempre pueden obtenerse coeficientes enteros multiplicando por una potencia de 10 suficientemente grande, en procesos iterativos en los que las precisión de los coeficientes es de varios decimales esto puede producir unos nuevos coeficientes de una magnitud excesiva dando problemas numéricos. Por otro lado, si se limita la magnitud de la potencia de 10 por la que se multiplican dichos coeficientes esto puede dar lugar a problemas que no resulten equivalentes a los problemas originales.

Por los motivos anteriores, hemos considerado conveniente disponer de implementaciones de algoritmos con las que se pudiensen trabajar directamente con coeficientes reales. Para ello, se han realizado ligeras modificaciones en el algoritmo de Martello y Toth [MaTo79] que han permitido resolver los problemas del tipo (KP') sin problemas numéricos. La eficiencia computacional de las versiones obtenidas es ligeramente inferior a la del algoritmo original para números enteros, puesto que algunas de las consideraciones que permiten reforzar las distintas cotas que utiliza el algoritmo de Martello y Toth [MaTo79] se basan en el carácter entero de los coeficientes. Por ese motivo las cotas que se obtienen en la versión que se ha implementado resultan ligeramente inferiores en cuanto a calidad, lo cual, desde un punto de vista de eficiencia computacional, incide en la exploración de un mayor número de nodos en el árbol de exploración.

En cualquier caso, consideramos que la versión obtenida resulta suficientemente eficiente en el contexto en el que se ha utilizado, especialmente teniendo en cuenta que ello nos ha permitido obviar problemas de carácter numérico y trabajar con problemas que aparecen en lugar de utilizar problemas que pueden resultar pseudo-equivalentes.

V.3. RESULTADOS COMPUTACIONALES

En esta sección se presentan, finalmente, los resultados computacionales obtenidos con la versión final de los algoritmos que se han implementado. Las pruebas se han realizado, en un VAX/VMS8600, sobre una batería de 56 problemas generados aleatoriamente similar a la que se describe en [FiKe86].

Para exponer los resultados se han agrupado los problemas según sus dimensiones. Cada uno de los cinco grupos que se han considerado está formado por unos 10 problemas. Los 10 problemas del primer grupo tienen, aproximadamente, 20 filas y 40 columnas, los 13 del segundo 20 filas y 60 columnas, los 11 del tercero 40 filas y 90 columnas los 11del cuarto 50 filas y 100 columnas y los 10 del quinto 100 filas y 200 columnas. Las densidades de los distintos problemas son, en los dos primeros grupos, próximas al 11%; en el tercero algo inferiores al 6%, para los 8 primeros problemas, y superiores al 15%, para los cuatro últimos; en el cuarto y quinto grupo , aproximadamente, del 10 y 8%, respectivamente.

A continuación, en las Tablas 1-5, se exponen los primeros resultados, en los que se comparan las dimensiones y densidades originales de los problemas con las que resultan después de aplicar los tests lógicos de eliminación.

Las cuatro primeras columnas indican, respectivamente, el número de filas, columnas, elementos no nulos de la matriz y densidades de los problemas generados, mientras que en las columnas 5-8 se indican los mismos datos referentes a los problemas después de haber aplicado los tests lógicos de eliminación. En la columna 9, fparc indica el valor de la función objetivo asociado a las variables que se han fijado a 1 en estos tests y cpu indica el tiempo en min:seg.cent. empleado en estos procedimientos previos.

Los resultados expuestos en las Tablas1-5 confirman la validez de estos procedimientos, puesto que, como se puede observar especialmente en los dos primeros grupos de problemas, en muchos casos se han obtenido reducciones importantes en las dimensiones de los mismos, llegando en algún caso a resolver directamente los problemas. En los tres últimos grupos, se han conseguido, excepto en un caso, muy pocas eliminaciones, siendo muchos los problemas para los que no se ha logrado ninguna reducción.

	m	n	ntot	dens	m	n	ntot	dens	fparc	cpu
p1	20	45	101	11.22	18	37	86	12.91	24	0:00.05
p2	20	42	89	10.59	16	27	57	13.19	34	0:00.07
p3	20	42	95	11.30	0	0	0	--	532*	0:00.02
p4	20	42	98	11.66	18	37	85	12.76	83	0:00.05
p5	20	44	120	13.63	20	43	119	13.83	0	0:00.02
p6	20	45	105	11.66	18	40	94	13.05	0	0:00.02
p7	20	40	80	10.00	2	3	4	33.33	0	0:00.01
p8	20	45	113	12.55	20	42	109	12.97	0	0:00.02
p9	20	45	108	12.00	20	42	105	12.80	0	0:00.01
p10	20	46	112	12.17	20	41	107	13.04	0	0:00.02

Tabla 1

	m	n	ntot	dens	m	n	ntot	dens	fparc	cpu
p11	20	56	125	11.16	20	48	114	11.87	0	0:00.07
p12	20	59	136	11.52	20	52	129	12.40	0	0:00.08
p13	20	63	152	12.66	20	62	151	12.17	0	0:00.07
p14	20	59	133	11.27	20	52	125	12.01	0	0:00.04
p15	20	63	146	11.58	20	57	138	12.10	0	0:00.10
p16	20	64	167	13.04	20	57	158	13.85	0	0:00.09
p17	20	61	138	11.40	13	23	48	16.05	0	0:00.08
p18	20	64	148	11.56	20	55	136	12.36	0	0:00.04
p19	20	62	149	12.01	19	53	140	13.90	13	0:00.03
p20	20	61	129	10.66	20	51	115	11.27	0	0:00.03
p21	20	60	141	11.75	20	55	134	12.18	0	0:00.03
p22	20	60	144	12.00	20	52	136	13.07	0	0:00.02
p23	20	65	160	12.30	20	60	154	12.83	0	0:00.03

Tabla 2

	m	n	ntot	dens	m	n	ntot	dens	fparc	cpu
p24	40	87	186	5.34	35	67	148	6.31	19	0:00.06
p25	40	87	193	5.54	23	38	68	7.78	320	0:00.22
p26	40	89	198	5.56	37	74	170	6.20	22	0:00.30
p27	40	91	210	5.76	21	33	57	8.22	425	0:00.19
p28	40	84	184	5.47	37	67	151	6.09	99	0:00.18
p29	40	82	190	5.79	20	29	54	9.31	234	0:00.17
p30	40	90	196	5.44	12	19	31	6.09	485	0:00.26
p31	40	98	251	6.40	40	94	247	6.56	0	0:00.46
p32	40	91	591	16.23	40	91	591	16.23	0	0:00.04
p33	40	93	590	15.86	40	93	590	15.86	0	0:00.04
p34	40	89	585	16.43	40	89	585	16.43	0	0:00.04
p35	40	90	597	16.58	40	90	597	16.58	0	0:00.04

Tabla 3

	m	n	ntot	dens	m	n	ntot	dens	fparc	cpu
p36	50	109	517	9.48	50	104	507	9.75	0	0:00.12
p37	50	100	502	10.04	50	99	498	10.06	0	0:00.12
p38	50	100	494	9.88	0	0	0	--	428*	0:00.25
p39	50	100	374	7.48	50	99	373	7.53	0	0:00.11
p40	50	100	497	9.94	50	98	495	10.10	0	0:00.13
p41	50	100	504	10.08	50	100	504	10.08	0	0:00.05
p42	50	100	502	10.04	50	97	499	10.28	0	0:00.11
p43	50	100	493	9.86	50	100	493	9.86	0	0:00.13
p44	50	100	496	9.92	50	98	494	10.08	0	0:00.06
p45	50	100	509	10.18	50	99	508	10.26	0	0:00.06
p46	50	100	452	9.04	50	100	452	9.04	0	0:00.23

Tabla 4

	m	n	ntot	dens	m	n	ntot	dens	fparc	cpu
p47	100	200	1599	7.99	100	200	1599	7.99	0	0:00.51
p48	100	200	1528	7.64	100	200	1528	7.64	0	0:00.49
p49	100	200	1542	7.71	100	200	1542	7.71	0	0:00.20
p50	100	200	1312	6.56	100	200	1312	6.56	0	0:00.19
p51	100	200	1320	6.60	100	200	1320	6.60	0	0:00.85
p52	100	200	1567	7.83	100	200	1567	7.83	0	0:00.20
p53	100	200	1545	7.72	100	200	1545	7.72	0	0:00.50
p54	100	200	1604	8.02	100	200	1604	8.02	0	0:01.11
p55	100	200	1524	7.62	100	200	1524	7.62	0	0:00.20
p56	100	200	1501	7.50	100	200	1501	7.50	0	0:00.52

Tabla 5

Para los problemas en los que se han eliminado tanto filas como columnas, se puede observar un ligero aumento en sus densidades. No consideramos que este incremento, cuando éste se produce, incida de forma negativa en la dificultad inherente a los problemas (a diferencia de los problemas de set packing y de set partitioning). En ese sentido, debemos resaltar que si un problema tuviera una densidad del 100%, su solución sería inmediata y, si las densidades fuesen muy altas, sería relativamente sencillo detectar la no factibilidad del mismo.

El esfuerzo computacional requerido para realizar estos tests resulta, como puede apreciarse, reducido. En cualquier caso, pensamos que las reducciones que se manifiestan son suficientemente importantes y que el esfuerzo computacional requerido es suficientemente limitado como para considerar que, en conjunto, la utilización de los procedimientos de eliminación de variables está plenamente junstificada.

A continuación, en las Tablas 6-10, se presentan los datos referentes al comportamiento del algoritmo básico para la resolución de los problemas de (SP). Los resultados que se exponen se refieren a los problemas obtenidos por la aplicación de los tests lógicos de eliminación a los problemas originales generados. En concreto, no se exponen resutados referentes tanto a P3 como a P38 puesto que estos dos problemas han sido resueltos óptimamente por los tests lógicos.

En las dos primeras columnas, se indica el valor de la función objetivo para la primera solución dual obtenida, en el primer caso (Du), mediante la heurística dual (c.f. II.2) para problemas de set partitioning puros y, en el segundo (Subg), mediante aplicación de optimización subgradiente a la relajación lagrangiana ordinaria (RL1u) de los problemas de set partitioning generados.

Para poder apreciar la importancia del procedimiento dual utilizado, en la calidad de la solución primal obtenida mediante la heurística propuesta en II.2, se incluyen los resultados de la aplicación de esta heurística a partir de las dos soluciones duales obtenidas por los procedimientos anteriores. Estos resultados aparecen en las columnas 3-4, en las que z1 y z2, indican, respectivamente, los valores de la función objetivo para la primera solución primal obtenida por aplicación de la heurística, a partir de las soluciones duales obtenidas mediante la heurística dual y optimización subgradiente.

En las columnas 5-9, ncort indica el número de desigualdades generadas antes de terminar el algoritmo, zsup y zinf los valores de la mejor cota superior e inferior, respectivamente, al terminar el algoritmo y it-zsup el número de desigualdades generadas antes de obtener la solución que ha proporcionado la mejor cota superior.

Nelim indica al número de variables eliminadas del problema (fijadas a 0) por aplicación de las condiciones lógicas, que se derivan cuando las disyunciones obtenidas constan de un único elemento, y fparc indica el valor de la función objetivo para las variables que se han fijado a 1 (si es que hay alguna) en la aplicación de los tests lógicos subsiguiente a dicha eliminación.

Los tiempos de cpu requeridos para la obtención de las dos soluciones duales, así como para los de la cota superior z2 (los de z1 son totalmente análogos ya que el procedimiento utilizado es el mismo), aparecen en las tres primeras columnas de las Tablas 11-15. En estas tablas, también se incluyen los tiempos (Corte) referentes a la generación de la primera desigualdad obtenida (los de las posteriores son totalmente similares), así como los requeridos (Heur2 y Heur2f) por la heurística primal (c.f.II.4) para la obtención de soluciones posibles para los problemas ampliados (SPA), tanto para el problema obtenido después de incorporar al problema original la primera desigualdad generada, como en la última iteración del algoritmo, en la que dicho procedimiento ha fallado determinando, por tanto, la terminación del mismo.

Como puede apreciarse por los resultados expuestos, la calidad de la solución dual obtenida mediante la aplicación de optimización subgradiente a los problemas duales asociados a la relajación lagrangiana es, sin lugar a dudas, mucho mejor que la obtenida mediante la heurística. Además, de forma algo sorprendente, el esfuerzo computacional requerido por el procedimiento heurístico resulta, en bastantes ocasiones, ligeramente superior al de la utilización de optimización subgradiente. La explicación a este hecho puede venir dada por la complejidad del procedimiento de mejora incluido en la heurística dual frente a la sencillez en la estructura de la relajación lagrangiana. En cualquier caso, tanto por la calidad de los resultados obtenidos como por el tiempo computacional requerido, es indudable que la utilización de los procedimientos de tipo subgradiente sugeridos resulta mucho más beneficiosa.

	Du	Subg	z1	z2	ncort	zsup	zinf	it-zsup	n-elim	fparc
p1	254.05	339.57	360*	360*	0	360*	355	0	20	360*
p2	276	292*	292*	292*	0	292*	292*	0	20	292*
p3	---	---	---	---	---	---	---	---	---	---
p4	215	324.37	478	392	3	334	325	2	5+1+11	
p5	118.70	143.09	274	163	0	163	144	0	22	
p6	142.31	254.70	584	584	3	530	273	2	0	
p7	87*	87*	87*	87*	0	87*	87*	0	0	
p8	146.75	233*	233*	233*	0	233*	233*	0	34	233*
p9	128.50	199*	283	199*	0	199*	199*	0	32	199*
p10	117.66	217.41	387	387	2	252	227.84	1	12	

Tabla 6

	Du	subg	z1	z2	ncort	zsup	zinf	it-zsup	n-elim	fparc
p11	133.33	221*	546	221*	0	221*	221*	0	7	
p12	169.00	206.95	382	303	2	216	208.71	1	1+30	
p13	213.22	281.08	448	380	6	319	281.56	5	3+5+16	
p14	153.66	166*	168	166*	0	166*	166*	0	43	166*
p15	103.42	140.78	329	153	1	153	141.09	0	27	
p16	196.33	217.24	340	312	20	272	221.52	14	1+4+1+4	
p17	101.00	148*	389	148*	0	148*	148*	0	17	148*
p18	65.80	122.8	146	146	1	146	122.98	0	29	
p19	136.66	226.45	305	292	3	262	226.45	2	14+5+9	
p20	95.00	174.49	289	252	6	213	178.67	3	2+14+3+1	
p21	213.33	311.10	499	492	8	312	311.87	7	7+7	
p22	214.27	284.70	389	361	3	305	285.63	2	2+16+10	
p23	116.17	153.01	164	283	2	164	153.18	1	6+31	

Tabla 7

	Du	Subg	z1	z2	ncort	zsup	zinf	it-zsup	nelim	fparc
p24	304.73	417.32	675	557	5	421*	421*	4	1+51	421*
p25	395.74	466*	854	466*	0	466*	466*	0	27	466*
p26	287.29	639.14	1015	892	4	655	639.87	3	14+1+17	
p27	391.50	495*	921	495*	0	495*	495*	0	22	495*
p28	325.77	472.50	637	561	4	538	479.59	2	5+2+3+2	
p29	332.00	378*	643	378*	0	378*	378*	0	20	378*
p30	229.00	243.46	776	291	1	291	243.46	0	2	
p31	344.21	587.18	931	934	3	702	602.38	1	4+3+3	
p32	53.46	148.61	174	297	2	174	151.88	1	35	
p33	71.95	133*	133*	133*	0	133*	133*	0	87	133*
p34	39.96	134.48	227	227	3	188	140.37	2	5+12	
p35	77.38	159.39	181	277	2	181	165.21	1	44	

Tabla 8

	Du	Subg	z1	z2	ncort	zsup	zinf	it-zsup	nelim	fparc
p36	208.27	549.29	570*	570*	1	570*	549.29	0	52	570*
p37	226.61	478*	478*	478*	0	478*	478*	0	89	468*
p38	---	---	---	---	---	---	---	---	---	---
p39	179.23	399.14	477	718	2	477	403.48	1	8	
p40	100.81	268.28	485	332	2	332	273.58	0	10+1	
p41	209.43	294.15	674	511	4	388	299.46	2	5	
p42	87.05	190.77	331	318	4	250	198.03	2	1+9+4	
p43	156.02	296.33	443	429	6	368	304.44	3	7+1+1	
p44	164.26	293.39	423	313	1	313	293.39	0	38	
p45	133.29	235.96	570	273	1	273	238.29	0	26	
p46	223.44	393.25	662	423	1	423	393.25	0	29	

Tabla 9

	Du	Subg	z1	z2	ncort	zsup	zinf	it-zsup	nelim	fparc
p47	187.76	397.27	631	614	2	521	413.13	1	2	
p48	208.02	434.45	658	599	9	582	448.82	3	2	
p49	171.97	329*	660	329*	0	329*	329*	0	97	329*
p50	224.29	432.24	658	613	9	486	441.92	8	21	
p51	172.08	460.49	811	566	2	566	473.32	0	3+1	
p52	239.79	431.51	763	664	12	584	445.89	7		
p53	205.66	409.30	547	707	7	516	424.86	2	5+3+1	
p54	209.36	379.11	480	691	4	480	418.84	3	17+7	
p55	116.22	333.66	648	421	2	421	347.02	0	10+6	
p56	143.57	404.39	670	545	3	441	412.44	3	1+1+51	

Tabla 10

	Du	Subg.	Heur1	Corte	Heur2	Heur2f	Dyer	Knp2r2	Knp2r1
p1	0:00.43	0:00.19	0:00.02	---	---	---	0:13.59	0:08.52	9:57.98
p2	0:00.08	0:00.15	0:00.02	---	---	---	0:00.54	---	0:07.16
p 3	---	---	---	---	---	---	---	---	---
p4	0:00.08	0:00.28	0:00.04	0:00.01	0:00.59	0:00.15	0:48:31	0:05.88	32:26.94
p5	0:00.32	0:00.01	0:00.01	---	---	---	0:06.47	0:25.81	0:51.48
p6	0:00.01	0:00.15	0:00.06	<0:00.01	0:01.02	0:00.06	0:15.26	0:10.74	15.27.80
p7	<0:00.01	0:00.01	0:00.01	---	---	---	0:00.01	---	0:00.06
p8	0:00.02	0:00.17	0:00.03	---	---	---	0:13.54	---	0:13.27
p9	0:00.01	0:00.07	0:00.02	---	---	---	0:00.98	---	0:10.76
p10	0:00.14	0:00.15	0:00.07	<0:00.01	0:00.16	0:00.11	0:27.48	3:20.68	0:00.43

Tabla 11

	Du	Subg.	Heur1	Corte	Heur2	Heur2f	Dyer	Knp2r1	Knp2r2
p11	0:00.81	0:00.43	0:00.11	---	---	---	0:01.29	---	0:29.87
p12	0:00.52	0:00.44	0:00.19	<0:00.01	0:00.54	0:00.18	1:18.38	0:11.86	5.31.68
p13	0:00.10	0:00.61	0:00.08	<0:00.01	0:03.99	0:01.57	2:05.27	25:21:82	0:14.24
p14	0:00.03	0:00.12	---	---	---	---	0:05.07	---	0:00.61
p15	0:00.03	0:00.55	0:00.70	<0:00.01	0:00.29	0:00.29	0:44.81	0:15.95	0:34.07
p16	0:01.10	0:00.53	0:00.06	0:00.01	0:01.37	0:07.66	2:27.91	0:25.90	0:12.37
p17	0:00.08	0:06.01	0:00.02	---	---	---	0:00.25	---	0:01.22
p18	0:00.25	0:00.20	0:00.02	<0:00.01	---	---	0:16.01	---	0:17.13
p19	0:00.27	0:00.18	<0:00.01	<0:00.01	0:00.57	0:00.24	0:20.52	0:04.72	0:17.88
p20	0:00.12	0:00.16	0:00.05	<0:00.01	0:00.21	0:00.43	0:21.11	0:03.20	0:51.80
p21	0:00.11	0:00.20	0:00.08	0:00.01	0:01.92	0:01.73	0:47.68	0:11.95	7:31.98
p22	0:00.16	0:00.19	0:00.03	<0:00.01	0:00.40	0:00.07	0:22.88	---	22:34.17
p23	0:00.19	0:00.25	0:00.04	<0:00.01	0:00.31	0.00.05	0:31.88	0:12.81	0:17.53

Tabla 12

	Du	Subg.	Heur1	Corte	Heur2	Heur2f	Dyer	Knp2r2	Knp2r1
p24	0:01.50	0:00.28	0:00.12	<0:00.01	0:00.45	0:03.65	1:05.55	0:16.42	1:15.02
p25	0:01.48	0:00.27	0:00.18	---	---	---	0:01.85	---	>60:00
p26	0:00.11	0:01.35	0:09.06	0:00.01	0:01.91	0.04.41	2:42.53	8:07.69	>60:00
p27	0:00.66	0:01.74	0:00.11	<0:00.01	0:00.02	0:00.02	0:02.34	---	>60:00
p28	0:02.08	0:00.69	0:00.38	0:00.01	0:00.87	0:09.44	2:59.97	>60:00	>60:00
p29	0:00.45	0:01.32	0:00.06	---	---	---	0:00.73	---	0:15.27
p30	0:00.09	0:00.16	0:00.05	<0:00.01	---	---	0:00.32	---	0:00.98
p31	0:03.48	0:01.97	0:11.81	0:00.01	0:02.56	1:53.78	16:32.16	0:49.67	>60:00
p32	0:00.42	0:00.94	0:00.04	0:00.01	0:00.94	0:00.39	0:34.44	9:31.78	7:37.17
p33	0:00.19	0:02.03	0:00.03	---	---	---	0:20.32	---	12:22.28
p34	0:00.10	0:00.94	0:00.03	0:00.01	0:00.81	0:00.87	0:55.89	0:32.89	32:19.27
p35	0:00.87	0:00.97	0:00.14	<0:00.01	0:02.12	0:00.42	1:04.88	>60:00	14:30.98

Tabla 13

	Du	Subg.	Heur1	Corte	Heur2	Heur2f	Dyer	Knp2r2	Knp2r1
p36	0:00.39	0:00.99	0:00.05	0:00.01	0:00.54	0:00.54	2:22.22	>60:00	15:43.98
p37	0:00.80	0:01.39	0:00.03	---	---	---	0:54.81	>60:00	
p38	---	---	---	---	---	---	---	---	---
p39	0:00.55	0:00.74	0:01.50	0:00.01	0:42.84	0:05.63	1:39.54	1:45.71	0:32.66
p40	0:00.80	0:02.13	0:02.18	0:00.01	0:32.16	0:02.42	2:17.01	>60:00	37:10.9
p41	0:00.45	0:00.97	0:00.05	<0:00.01	0:16.84	0:01.77	3:55.48	4:08.83	6:41.55
p42	0:00.33	0:00.95	0:00.11	<0:00.01	0:13.45	0:08.67	2:41.95	>60:00	>60:00
p43	0:00.46	0:02.29	0:00.44	0:00.02	0:08.53	0:12.60	6.10.11	1:57.31	21:20.50
p44	0:06.27	0:00.96	0:00.17	<0:00.01	0:03.29	0:03.29	2:49.26	0:50.79	4:02.95
p45	0:00.19	0:00.94	0:00.07	<0:00.01	0:03.46	0:03.46	2.33.56	>60:00	12:46.55
p46	0:00.64	0:03.30	0:00.35	0:00.02	0:26.34	0:26.34	7:56.77	32:06.36	18.40.97

Tabla 14

	Du	Subg.	Heur1	Corte	Heur2	Heur2f	Dyer	Knp2r2	Knp2r1
p47	0:20.84	0:04.90	0:00.98	0:00.02	2:24.27	1:58.44	5:32.62	>60:00	>60:00
p48	0:28.50	0:11.13	0:00.64	0:00.04	9.29.51	4:34.87	9:03.23	46:25.05	>60:00
p49	0:43.05	0:18.38	0:08.37	0:00.06	17:43.53	1:56.17	32: 03.92	>60:00	>60:00
p50	0:02.02	0:04.63	0:02.03	0:00.01	1:44.48	2:18.76	2:28.99	16:52.66	>60:00
p51	0:38.25	0:10.53	0:01.66	0:00.03	10:53.57	0:08.52	10:19.20	10:15.75	>60:00
p52	0:06.48	0:11.06	0:11.01	0:00.04	5:04.88	1:09.83	7:24.93	30:10.53	>60:00
p53	0:05.50	0:10.75	0:09.17	0:00.03	5:36.27	5:40.69	17:15.08	>60:00	>60:00
p54	0:04.72	0:07.23	0:00.26	---	---	----	1:49:35	1:43.44	>60:00
p55	0:12.45	0:04.15	0:08.39	0:00.02	6:28.93	15.16.95	4:48.03	34:04.27	>60:00
p56	1:07.76	0!16.20	0:13.02	0:00.05	[illegible]8:45.63	12:32.04	14:43.33	>60:00	>60:00

Tabla 15

Puede también observarse que la calidad de la solución primal obtenida depende en gran medida de la solución dual a partir de la cual se construye. En particular, la solución obtenida después de utilizar los procedimientos de tipo subgradiente es a menudo sensiblemente mejor, y sólamente en 7 casos es ligeramente peor, que la que se obtiene después de haber utilizado las heurísticas. Estos resultados ratifican la importancia de utilizar un procedimiento dual que proporcione soluciones de la mejor calidad posible para la obtención de las mejores soluciones en las heurísticas primales.

La heurística para obtener soluciones posibles primales para los problemas ampliados, ha requerido, como puede observarse, un esfuerzo computacional que en algunos problemas, sobre todo en los de dimensiones grandes, puede resultar excesivo. Sin embargo, de forma algo inesperada, el tiempo requerido para su aplicación parece disminuir a medida que se incorpora un mayor número de desigualdades a los problemas. Ello puede, tal vez, explicarse por el hecho de que a medida que se generan desigualdades, éstas resultan cada vez más potentes (menor número de variables que intervienen en las mismas). Por ese motivo, la capacidad de elección de dicho procedimiento resulta mucho más limitada, terminando, en general, de forma más rápida.

Hay que resaltar que, para intentar obtener un procedimiento que computacionalmente resultase más rentable, se han implementado versiones diferentes de esta heurística primal menos exigentes en la obtención de la solución del subproblema de set covering contenido en el problema ampliado, pero estos procedimientos han proporcionado un número altísimo de fallos al intentar completar dichas subsoluciones en soluciones posibles del problema ampliado. En ese sentido, consideramos que el procedimiento utilizado resulta válido en la medida que permite obtener soluciones posibles de una buena calidad para los problemas ampliados. Sobre todo, teniendo en cuenta que, en casi todos los casos, la mejor solución encontrada lo ha sido mediante este procedimiento. Además, este procedimiento es el que ha permitido plasmar de forma satisfactoria la información contenida en las desigualdades válidas que se han generado.

Con respecto a las desigualdades válidas obtenidas, los resultados presentados confirman el buen rendimiento de las mismas. En primer lugar, en casi todos los problemas, después de un número reducido de desigualdades, se ha mejorado la calidad de la solución posible inicial.

Es importante señalar también, que estas desigualdades han permitido mejorar la calidad de las cotas inferiores obtenidas. Como se desprende de las Tablas 6-10, la cota inferior que se tiene en la terminación del algoritmo es, prácticamente en todos los casos, mejor que la obtenida en la primera iteración del mismo, cuando todavía no se había generado ninguna de estas desigualdades. Desgraciadamente, dichas mejoras no han resultado todo lo satisfactorias que se esperaba, siendo éste, en nuestra opinión, el mayor inconveniente del algoritmo que se ha utilizado.

Además, en muchas ocasiones ha sido posible eliminar variables de los mismos, con lo que los problemas obtenidos han resultado mucho más sencillos de resolver, tanto para obtener buenas soluciones primales, como para acercarse a la terminación del algoritmo. Hay que resaltar que, en la mayoría de los problemas en los que se han eliminado muchas variables, la aplicación posterior de los tests de eliminación ha proporcionado, directamente, la solución óptima. En otros problemas, en los que lo anterior no ha ocurrido, el gap final, entre la mejor solución primal y la mejor solución dual, es más pequeño que en los problemas en los que no se han eliminado tantas variables.

En definitiva, consideramos que el rendimiento del procedimiento de obtención de desigualdades válidas es altamente satisfactorio, tanto por los resultados que se derivan de su utilización, como por el excelente tiempo computacional que requiere su utilización.

A continuación, se presentan los resultados referentes a los distintos métodos de refuerzo dual que se han utilizado para intentar mejorar la calidad de la cota inferior, que, a menudo, no permite demostrar que la mejor solución primal obtenida es óptima, en los casos en los que esto es así, ni permite evaluar de una forma precisa la calidad de la misma cuando lo anterior no ocurre.

En particular, para cada uno de los problemas, se presentan:

1) La cota inferior obtenida mediante la aplicación del procedimiento de Dyer, en la obtención de una restricción subrogada $gx \geq g_0$ para los problemas ampliados obtenidos en el algoritmo básico.

2) La cota inferior que se obtiene con la aplicación del método de optimización subgradiente, al problema dual asociado la relajación lagrangiana (RL5u), que consiste en incorporar a la función objetivo las restricciones del problema original, manteniendo como restricción la subgogada obtenida en el procedimiento anterior.

3) Las cotas inferiores obtenidas por la aplicación de las variantes propuestas del método BISA a los problemas duales asociados a las relajaciones lagrangianas (RL6u) y (RL7u), en las que se incorporan a la función objetivo todo el conjunto de restricciones de los problemas originales, manteniendo dos restriciones: en el primer caso $cx \geq s_0$ y $nx \leq m$ y, en el segundo, $cs \geq s_0$ y la restricción subrogada $gx \geq g_0$ obtenida por el procedimiento de Dyer.

La restricción $cx \geq s_0$ significa, como ya se ha comentado repetidas veces, que las soluciones obtenidas tengan un valor de la función objetivo asociada al problema original mayor o igual que el de la mejor cota inferior conocida hasta el momento, mientras que $nx \leq m$ implica, en cierta manera, que las soluciones obtenidas se aproximan, en la medida de lo posible, a las condiciones de ortogonalidad impuestas por el conjunto original de restricciones de los problemas de set partitioning.

Estos resultados se exponen en las Tablas 16-20. La primera columna, knp2r1, indica el valor de la cota inferior obtenida mediante la variante utilizada del método BISA en la que se considera la relajación lagrangiana (RL6u). El número de iteraciones internas del algoritmo, se ha limitado a 200, en los casos en los que el algoritmo no ha alcanzado previamente la convergencia. Teniendo en cuenta que las dos restricciones que considera esta relajación lagrangiana no utilizan la información proporcionada por el algoritmo básico, este procedimiento se ha utilizado de forma independiente al mismo. En particular, inicialmente se ha considerado que no se disponía de ninguna cota inferior y por lo tanto el valor s_0, que interviene en la segunda restricción, se ha tomado como 0.

En la columna 2, Dyer indica el valor de la cota inferior obtenida mediante el procedimiento de Dyer para la obtención de una restricción subrogada a los problemas ampliados. En este caso, el número de iteraciones internas del procedimiento se ha limitado a 500, en los casos en los que el mismo no ha proporcionado la solución óptima. En la columna 3, (RL5) indica el valor de la cota inferior para la relajación lagrangiana (RL5u). En la columna 4, knp2r2 indica la cota inferior obtenida por la aplicación de la variante del algoritmo BISA a la relajación (RL7u). En este caso, inicialmente, se toma como cota inferior inicial s_0, el valor de la cota inferior obtenida por el procedimiento de Dyer. Para este último algoritmo, no se ha limitado el número de iteraciones internas (iter) que se expone en la columna 5.

En las columnas 4 y 5, teniendo en cuenta que las formulaciones de los procedimientos utilizados utilizan la información obtenida por el procedimiento de Dyer, sólamente se exponen resultados referentes a problemas para los que dicho procedimiento no ha encontrado el óptimo.

Para los problemas en los que el algoritmo básico había eliminado todas sus variables, en la aplicación de los test lógicos subsiguiente al procedimiento que se aplica cuando las disyunciones obtenidas constan de un único elemento, los resultados que se exponen en las Tablas 16-20 se refieren a la aplicación de los métodos de refuerzo dual, que se acaban de comentar, a los problemas originales de set partitioning.

Los tiempos requeridos por los procedimientos anteriores se encuentran en las últimas columnas de las Tablas 11-15.

Las dos últimas columnas de las Tablas 16-20 indican, respectivamente, los gaps relativos, entre la mejor cota superior y la mejor cota inferior, existentes al final del algoritmo básico y después de la aplicación de los tres métodos de refuerzo dual, a fin de poder comparar las mejoras obtenidas.

	knp2r2	Dyer	RL5	knp2r1	gap1	gap2
p1	354	355	360*	360*	0	0
p2	292*	292*	---	---	0	0
p3	---	---	---	---	---	---
p4	334*	334*	---	---	2.69	0
p5	162	149	163*	163*	0	0
p6	308	303	314.64	315	48.49	40.56
p7	87*	87*	---	---	0	0
p8	233*	233*	---	---	0	0
p9	199*	199*	---	---	0	0
p10	222	225	227.81	236	9.52	6.43

Tabla 16

	knp2r1	Dyer	RL5	knp2r2	gap1	gap2
p11	221*	221*	---	---	0	0
p12	216*	≥216*	---	---	3.2	0
p13	282	281	281.61	286	11.5	10.34
p14	166*	166*	---	---	0	0
p15	187*	187*	---	---	0	0
p16	218	238	237.62	239	18.38	---
p17	148*	148*	---	---	0	0
p18	128	≥ 146*	---	---	15.75	0
p19	180	≥ 213*	---	---	15.96	0
p21	312*	307	307.42	312*	0	0
p22	294*	284	292.71	293	6.22	0
p23	161	≥ 164*	---	---	6.09	0

Tabla 17

	knp2r2	Dyer	RL5	knp2r1	gap1	gap2
p24	416	412	418.44	413	0	0
p25	430	466*	---	---	0	0
p26	460	610	654.37	632	2.29	0
p27	445	495*	---	---	0	0
p28	471	467	477.11	467	10.78	10.78
p29	378*	378*	---	---	0	0
p30	245*	245*	---	---	16.15	0
p31	421	567	625.33	576	14.10	10.82
p32	138	123	148.65	157	12.64	9.77
p33	127	133*	---	---	0	0
p34	123	117	136.57	137	25.00	25.00
p35	146	128	164.83	≥ 181*	8.28	0

Tabla 18

	knp2r2	Dyer	RL5	knp2r1	gap1	gap2
p36	400	421	570*	421	0	0
p37	451	403	478*	403	0	0
p38	---	---	---	---	---	---
p39	217	224	268.32	224	17.46	17.46
p40	224	278	297.70	278	22.68	22.68
p41	100	159	191.99	187	20.04	20.04
p42	237	280	315.54	294	17.11	14.13
p43	227	310	295.16	310	6.07	5.43
p44	207	213	240.93	223	12.45	11.72
p45	335	346	395.53	344	6.85	6.38

Tabla 19

	knp2r1	Dyer	RL5	knp2r2	gap1	gap2
p47	206	317	≤ 317	317	20.53	20.53
p48	273	339	≤ 339	339	22.85	22.85
p49	173	301	329*	309	0	0
p50	129	363	462.68	363	9.25	4.73
p51	282	398	≤ 398	398	14.48	14.48
p52	217	370	443.68	449	23.63	23.11
p53	225	312	315.43	312	17.63	17.63
p54	210	326	331.12	326	12.70	12.70
p55	174	254	347.39	254	17.33	17.33
p56	175	289	408.27	289	6.34	6.34

Tabla 20

Los resultados presentados en las Tablas 16-20 indican un comportamiento muy desigual de los métodos utilizados para los distintos problemas. En general, los resultados obtenidos para problemas de dimensiones reducidas resultan, en casi todos los casos y con todos los métodos, bastante mejores que los obtenidos para los de dimensiones mayores.

Los resultados obtenidos por la aplicación de la variante del método BISA a la relajación lagrangiana (RL6u) resultan bastante satisfactorios para los problemas de los dos primeros grupos. Hay que resaltar que se han resuelto al óptimo todos los problemas de estos dos grupos que se habían resuelto al óptimo con la relajación lagrangiana ordinaria, que se ha utilizado en el algoritmo básico, y que para los restantes problemas de estos dos grupos las cotas obtenidas resultan de una calidad apreciable y el esfuerzo computacional requerido para la convergencia, en casi todos los casos, satisfactorio.

Desgraciadamente, los resultados obtenidos con problemas de dimensiones mayores han sido bastante pobres. Para éstos problemas, ni la calidad de las cotas obtenidas, ni los tiempos utilizados por el procedimiento han resultado aceptables. Pensamos, sin embargo, que la mala calidad de las cotas obtenidas se debe al altísimo esfuerzo computacional requerido, puesto que en 5 problemas de los grupos 3 y 4 y en los 10 problemas del grupo 5, éste ha sido superior al límite máximo de 60 minutos. Por lo tanto, para estos problemas resulta imposible conocer cual hubiera sido el valor de la cota obtenida si el procedimiento hubiera continuado hasta su convergencia o hasta el límite máximo de iteraciones.

El procedimiento de Dyer proporciona, en general para los problemas de los dos primeros grupos, unos resultados bastante satisfactorios. Como se puede apreciar, en bastantes de estos problemas se ha obtenido una cota inferior que ha permitido demostrar que la mejor solución primal que se tiene es óptima. Para los restantes problemas de estos dos grupos se han obtenido unos resultados similares en calidad a los obtenidos con el algoritmo básico.

No ha ocurrido lo mismo, sin embargo con los problemas de dimensiones mayores, para los que las cotas inferiores obtenidas han resultado ser peores que las proporcionadas por el algoritmo básico. Este hecho puede entenderse, intuitivamente, al considerar que, para los problemas con un mayor número de restricciones resulta difícil sintetizar la información contenida en todas ellas en una única desigualdad. Analíticamente, lo anterior se refleja en una mayor cantidad de coeficientes a evaluar en cada una de las iteraciones internas del algoritmo y en la obtención de combinaciones lineales de gran número de restricciones.

Los tiempos de cálculo requeridos por este procedimiento son, en todos los casos sensiblemente mayores que los de la utilización de optimización subgradiente aplicada a la relajación lagrangiana ordinaria, y ello se debe al gran número de iteraciones internas que realiza el algoritmo.

La aplicación de la variante del procedimiento BISA a la relajación lagrangiana (RL7u) ha proporcionado mejoras apreciables con respecto a la cota obtenida por el procedimiento de Dyer. En algunos casos, sobre todo con problemas de dimensiones reducidas, la calidad de las cotas obtenidas con este procedimiento ha permitido reducir sensiblemente el gap que se tenía.

Los tiempos requeridos para la teminación de este algoritmo resultan bastante satisfactorios, teniendo en cuenta la estructura de los problemas de knapsack con dos restricciones que se resuelven en cada iteración, para los problemas de dimensiones reducidas, pero son extremadamente desiguales, para los problemas de dimensiones más altas. Ello puede justificarse, parcialmente, por la diferencia en el número de iteraciones internas del procedimiento, ya que en algunos casos la convergencia ha sido rápida, mientras que en otros se ha alcanzado el límite máximo de iteraciones establecido. Sin embargo, pueden también apreciarse diferencias sensibles en los tiempos requeridos para problemas en los que han sido necesarias un número similar de iteraciones y hay 10 problemas, a nuestro juicio un número excesivo, para los que no se ha alcanzado ni la convergencia ni el número máximo de iteraciones en el tiempo límite de 60 minutos.

La utilización de optimización subgradiente para resolver la relajación lagrangiana (RL5u) ha proporcionado mejoras equiparables a la obtenidas mediante el procedimiento anterior. Los tiempos requeridos en este caso son bastante reducidos, lo cual no resulta sorprendente, teniendo en cuenta que en cada iteración interna se resuelve un problema clásico de knapsack con una restricción, cuya estructura es bastante sencilla.

En definitiva, consideramos que la información proporcionada por una "buena" restricción surrogada puede contribuir, de forma sensible a la mejora de la calidad de las cotas inferiores. Ahora bien, en cuanto los requerimientos computacionales de los procedimientos que consideran dichas restricciones, la experiencia obtenida parece indicar que, si la estructura de los problemas en los que se incluye esta restricción resulta algo complicada, éstos serán excesivos cuando los problemas sean de dimensiones grandes.

La utilización de la variante del método BISA que incluye una restricción subrogada para reforzar la formulación de la relajación lagrangiana utilizada, sólo parece recomendable, por tanto cuando el gap de dualidad que se tenga inicialmente resulte exageradamente grande.

Los resultados obtenidos por la aplicación de los métodos de refuerzo dual, que se han presentado en las Tablas 16-20 han permitido, sin embargo, reducir de forma apreciable el gap entre la mejor cota superior y la mejor cota inferior en gran cantidad de casos en los que el algoritmo básico no había conseguido resolver los problemas al óptimo.

Además, en algunas ocasiones, se ha obtenido la solución óptima para problemas que el algoritmo básico no había resuelto óptimamente, demostrando así, que en esos casos, la mejor solución encontrada no era óptima.

En definitiva, consideramos que desde una perspectiva de mejora de la calidad de la cota inferior, los procedimientos de refuerzo dual resultan válidos, a pesar de que el esfuerzo computacional requerido ha sido, en bastantes casos excesivo.

Finalmente, en las Tablas 21-24, se presentan con detalle los resultados obtenidos por la aplicación de los algoritmos (c.f. IV.2.2 y IV.3.2), utilizados, respectivamente, para resolver las dos versiones de problemas de knapsack con dos restricciones, que aparecen en las relajaciones lagrangianas (RL6u) y (RL7u) utilizadas en las variantes del algoritmo BISA. Los resultados que se exponen corresponden a los problemas para los que por lo menos una de las variantes utilizadas ha terminado en el tiempo máximo de 60 minutos.

Teniendo en cuenta que ambos algoritmos se han utilizado en procedimientos iterativos, los resultados que se exponen se corresponden al comportamiento medio de los mismos. En concreto, en las columnas 1 y 4, dif1 y dif2 indican las diferencias medias relativas entre el valor de la solución de la heurística utilizada y el óptimo del problema. En las columnas 2 y 5, nelim1 y nelim2 indican el número medio de variables eliminadas por iteración y en las columnas 3 y 6 se indican los tiempos medios por iteración requeridos por ambos algoritmos.

	dif1	nelim1	t1	dif2	nelim2	t2
p1	0.771	28.12	0:02.98	16.017	2.24	0:00.04
p2	0.678	21.25	0:00.23	---	---	---
p3	---	---	---	---	---	---
p4	0.922	8.43	0:09.73	5.63	1.37	0:00.42
p5	0.442	14.97	0:00.25	12.425	2.14	0:00.06
p6	1.016	28.16	0:04.63	14.79	5.38	0:00.76
p7	0.026	0.45	0:00.03	---	---	---
p8	0.272	2.34	0:00.13	---	---	---
p9	0.171	4.15	0:00.98	---	---	---
p10	0.293	21.35	0:00.14	5.01	11.51	0:00.18

Tabla 21

	dif1	nelim1	t1	dif2	nelim2	t2
p11	0.131	33.22	0:00.63	---	---	----
p12	0.122	21.57	0:01.65	5.56		0:08.54
p13	0.319	33.24	0:00.07	4.53		0:00.01
p14	0.172	47.31	0:00.02	---	---	---
p15	2.480	51.24	0:00.17	5.48		0:00.08
p16	0.697	37.11	0:00.11	3.711		0:00.14
p17	0.473	1.23	0:00.04	---	---	---
p18	0.361	19.25	0:00.08	---	---	---
p19	0.516	7.64	0:00.08	3.51		0:00.02
p20	0.213	24.86	0:00.25	3.132		0:00.01
p21	0.370	30.36	0:02.25	2.719		0:00.06
p22	2.230	10.18	0:06.67	---		---
p23	0.760	15.41	0:00.08	3.494		0:00.06

Tabla 22

	dif1	nelim1	t1	dif2	nelim2	t2
p24	1.050	12.69	0:00.37	1.822	2.35	0:00.07
p26	---	---	---	1.944	5.46	0:00.40
p29	1.230	2.17	0:00.56	---	---	---
p30	0.850	11.73	0:00.05	---	---	---
p31	---	---	---	2.012	45.37	0:00.19
p32	1.935	42.57	0:02.28	7.500	23.12	0:00.59
p33	1.309	0.36	0:03.71	---	---	---
p34	2.254	46.89	0:09.69	6.957	31.40	0:00.42
p35	1.763	21.35	0:04.31	---	---	---

Tabla 23

	dif1	nelim1	t1	dif2	nelim2	t2
p36	0.130	34.47	0:04.71	---	---	---
p39	0.730	75.43	0:00.16	1.866	54.11	0:00.38
p40	1.740	42.21	0:11.40	---	---	---
p41	1.480	71.73	0:02.00	2.757	46.29	0:00.83
p43	1.630	54.11	0:06.40	2.473	58.31	0:00.58
p44	1.460	43.52	0:01.21	3.029	36.92	0:00.18
p45	1.010	41.91	0:03.83	---	---	---
p46	2.030	36.44	0:05.61	1.195	61.47	0:00.01
p47	---	---	---	3.576	145.42	0:00.03
p49	---	---	---	4.305	68.16	0:00.02
p50	---	---	---	2.601	127.74	0:00.04
p53	---	---	---	3.518	130.69	0:00.05
p55	---	---	---	3.354	142.45	0:00.02
p56	---	---	---	3.345	139.52	0:00.01

Tabla 24

De los resultados que aparecen en las Tablas 21-24 se deduce el buen comportamiento general de los dos algoritmos utilizados. Como se puede apreciar, la calidad media de las soluciones posibles obtenidas por las heurísticas (c.f. IV.4.1 y IV.4.2) utilizadas en ambos casos es altamente satisfactoria. Hay que resaltar, asimismo, el gran número de variables que, en media, se eliminan en cada iteración.

Hay que señalar, sin embargo que, el comportamiento de los algoritmos utilizados decae ligeramente, a medida que aumenta el valor del término independiente de la restricción $cx \geq s_0$. Ello no resulta sorprendente, si consideramos que el aumento en dicho término independiente supone una mayor dificultad para satisfacer esta restricción y, por tanto, para encontrar soluciones posibles que cumplan las dos restricciones de los problemas formulados. Desde un punto de vista intuitivo, lo anterior está plenamente justificado, puesto que a medida que aumenta el valor de la cota inferior para los problemas y, por tanto, nos acercamos al valor óptimo de los mismos, los conjuntos de soluciones posibles para las relajaciones que se han formulado, son cada vez más restringidos, con lo que aumenta la dificultad para encontrar soluciones posibles.

Este descenso en el rendimiento de ambos algoritmos se refleja tanto en la calidad de las soluciones obtenidas mediante las dos heurísticas como en el número de variables eliminadas.

Tanto por la calidad de los resultados obtenidos como por los requerimientos computacionales de tiempo de los dos algoritmos que se han utilizado, consideramos que, en ambos casos, su rendimiento ha sido satisfactorio.

Como comentario final a esta sección referente a los resultados obtenidos en las experiencias computacionales realizadas con los distintos procedimientos que se han implementado, hay que señalar que el comportamiento del algoritmo básico utilizado resulta satisfactorio. Se ha podido probar la eficacia tanto de las heurísticas primales, como de las desigualdades válidas que se han utilizado. Estas desigualdades han permitido, en pocas iteraciones, obtener mejoras importantes en la calidad de las soluciones posibles primales. Además, la incorporación de todas las desigualdades obtenidas permite, en general, mejorar de forma apreciable la calidad de la cota inferior obtenida.

La interpretación de los resultados de los métodos de refuerzo dual propuestos resulta más conflictiva. El procedimiento de Dyer para la obtención de una restricción surrogada resulta rentable, tanto respecto a la calidad de la cota inferior que proporciona, como respecto a la calidad de la restricción subrogada obtenida, sobre todo en problemas de dimensiones reducidas. Esto último se ha podido apreciar por las mejoras obtenidas por los procedimientos que utilizan dicha restricción.

Ahora bien, la estructura de los problemas de knapsack con dos restricciones resulta excesivamente rígida para que el esfuerzo computacional requerido para la convergencia de las variantes propuestas del algoritmo BISA resulte rentable ., para problemas de grandes dimensiones, cuando el gap de dualidad que se tiene inicialmente no sea muy grande.

CAPÍTULO 6

CONCLUSIONES

En este último capítulo haremos un análisis de las principales conclusiones que se derivan del trabajo realizado, así como de las líneas futuras de investigación que pueden seguirse a partir de dichas conclusiones.

Teniendo en cuenta el contexto general del trabajo que se ha realizado, se plantean distintos elementos a analizar en estas conclusiones. Por un lado, los referentes a los diversos componentes que, eventualmente, pueden formar parte de la clase de algoritmos propuestos y, por otro lado, los referentes a la propia familia de algoritmos.

Una de las componentes fundamentales que se ha propuesto para la resolución de los problemas puros de set partitioning es la heurística primal presentada en II.2. Con respecto a la misma, cabe subrayar el buen funcionamiento que ha tenido en la mayoría de los problemas. Ello se ha manifestado especialmente en dos aspectos: el número de casos para los que la solución obtenida por dicha heurística ha sido óptima y el reducidísimo porcentaje de fallos de la misma.

Hay que resaltar, además, que una de las principales virtudes de dicho procedimiento es el de encontrar soluciones posibles incluso para problemas de los que se tiene la sospecha que el número de sus soluciones posibles es muy reducido. Ya se ha comentado repetidas veces la dificultad intrínseca que existe en la obtención de soluciones posibles para problemas de set partitioning. En ese sentido, cabe conjeturar que para problemas que tengan una única solución, incluso los de grandes dimensiones, este procedimiento encontraría el óptimo en la mayoría de los casos.

Con respecto de la heurística propuesta en II.4 para obtener soluciones primales para problemas de set partitioning ampliados con restricciones de set covering, consideramos que su eficiencia y utilidad ha sido, también suficientemente probada. Este procedimiento nos ha permitido considerar problemas con una estructura que hasta ahora no había sido estudiada y obtener soluciones posibles para los mismos, que, a su vez, son soluciones posibles para los problemas de set partitioning originales.

Exceptuando los problemas resueltos al óptimo en la primera iteración, esto es, aquellos para los que el procedimiento dual utilizado en la primera iteración ha proporcionado una cota inferior igual a la cota superior proporcionada por el procedimiento primal, en casi todos los demás casos, la mejor cota superior obtenida lo ha sido gracias a la obtención de soluciones posibles para los problemas ampliados. En ese sentido, la utilización del procedimiento propuesto ha resultado fundamental.

Sin embargo, la eficiencia, en términos de coste computacional, de dicho procedimiento no ha resultado, en general, equiparable al del procedimiento para problemas de set partitioning puros. En ese sentido, cabe cuestionarse la posiblilidad de establecer alguna otra heurística para obtener soluciones posibles para problemas de set partitioning ampliados con un mejor rendimiento computacional. Ello vendría dado en términos de algún procedimiento similar al propuesto, que no resultase tan exigente en la construcción de la subsolución del subproblema de set covering contenido en el problema ampliado. Las experiencias computacionales realizadas con algunas heurísticas de ese tipo han proporcionado un número altísimo de fallos. Ahora bien, queda abierta la cuestión de encontrar un procedimiento para encontrar soluciones posibles para los problemas ampliados que suponga un compromiso entre el esfuerzo computacional requerido y su capacidad para proporcionar soluciones posibles en la mayoría de los casos.

Los procedimientos de búsqueda de soluciones posibles propuestos para problemas de set partitioning, tanto puros como ampliados, presentan, sin embargo, un problema claro. Se trata de la dificultad para demostrar que la solución obtenida es óptima en los casos en los que el gap de dualidad que se produce entre el par de soluciones dual-primal es grande. Como ya se ha visto por los resultados de las experiencias computacionales, la calidad de la mejor cota inferior obtenida por los procedimientos duales utilizados en el algoritmo básico resulta, a menudo, insuficiente para demostrar que la mejor solución posible primal que se tiene es óptima. En este sentido, la intuición de que en muchos casos esto sea así resulta un argumento totalmente desechable.

La utilización de la relajación lagrangiana (tanto la relajación lagrangiana ordinaria (RL1u) como aquella en la que se mantiene como restricción explícita la última desigualdad válida obtenida (RL9u)) como método de refuerzo de las heurísticas duales, propuestas en el algoritmo básico, ha proporcionado resultados satisfactorios. Ello se ha reflejado, por un lado, en la mejora sensible de la calidad de las cotas inferiores respecto a las obtenidas mediante las heurísticas duales y, por otro lado, en la mejora, casi sistemática, de la calidad de las soluciones primales que se han obtenido al tomar el vector de costes reducidos asociado a las soluciones duales, proporcionadas por estas relajaciones lagrangianas, como base de partida para la construcción de las distintas soluciones primales. El esfuerzo computacional requerido por la utilización de optimización subgradiente para resolver los problemas duales asociados a ambas relajaciones ha sido poco superior al de la utilización de las heurísticas duales propuestas, con lo que, sin lugar a dudas, pensamos que dichas relajaciones lagrangianas deben considerarse como un elemento básico en el procedimiento general.

Con respecto al procedimiento de generación de desigualdades válidas para los problemas de set partitioning propuesto en II.3, consideramos que, junto con los procedimientos de obtención de soluciones posibles primales, forma el nucleo básico para el buen funcionamiento general del algoritmo. Hay que resaltar que la reducción del espacio de búsqueda de soluciones posibles, que resulta de la incorporación a los problemas originales de las desigualdades obtenidas, se ha plasmado, casi sistemáticamente, en la mejora de la calidad de la solución primal obtenida. En ese sentido, si consideramos que las heurísticas primales son valiosas, puesto que proporcionan soluciones posibles de una calidad suficientemente buena, los planos secantes generados refuerzan la utilidad de los procedimientos anteriores, puesto que permiten mejorar la calidad de dichas soluciones.

Además, la capacidad de eliminación de variables que se deriva de la utilización de las disyunciones, a partir de las cuales se obtienen dichas desigualdades, resulta especialmente sugerente al reducir, a menudo de forma determinante, las dimensiones de los problemas que se están resolviendo.

Por lo tanto, la validez del procedimiento de generación de desigualdades válidas resulta incuestionable, teniendo en cuenta el escasísimo esfuerzo computacional que requiere su utilización y la incorporación de estas desigualdades a los problemas originales.

Las conclusiones que pueden obtenerse sobre la utilización de los distintos métodos de refuerzo dual propuestos resulta, en general, mucho más contradictoria. Como comentario general, antes de analizar cada uno de ellos con más detalle, cabe resaltar que el rendimiento de estos métodos ha sido muy desigual. Existe, sin embargo, una característica común a los tres tipos de procedimientos utilizados; es el comportamiento relativamente satisfactorio de los métodos para problemas de dimensiones reducidas y el descenso espectacular en los resultados obtenidos mediante los mismos para problemas de dimensiones mayores.

La utilización del procedimiento de Dyer para la obtención de una buena restricción subrogada ha resultado bastante eficiente para problemas de dimensiones reducidas. Hay que señalar que, para este tipo de problemas, se ha conseguido alcanzar el óptimo para algunos de ellos y que, en otros casos, las cotas obtenidas han resultado ser mucho mejores que las obtenidas por las relajaciones lagrangianas incorporadas al algoritmo básico propuesto. En ese sentido, la información proporcionada por las desigualdades válidas, incorporadas a los problemas originales, ha sido básica, puesto que a partir de ella hemos podido deducir, en bastantes casos, que si todas las desigualdades incorporadas tuviesen que satisfacerse se obtendría una restricción subrogada, y con ello una cota inferior, que sería mayor que la mejor cota superior conocida y, con ello, demostrar que la mejor solución obtenida es óptima.

Sin embargo, el comportamiento de este procedimiento para los problemas de dimensiones más grandes no ha resultado satisfactorio ni en la calidad de las cotas proporcionadas ni en el esfuerzo computacional requerido.

Los resultados proporcionados por las variantes propuestas del método BISA [Barci85a, Barci85b] no han proporcionado, en general, los resultados esperados, especialmente para problemas de grandes dimensiones.

Inicialmente, se implementó la versión original de dicho método en la que, en la relajación lagrangiana que se utiliza, únicamente se considera como restricción explícita $cx \geq s_0$; es decir, la restricción que impone que las soluciones obtenidas tengan un valor de la función objetivo del problema original por lo menos tan grande como el de la mejor cota inferior que se conoce. Los resultados obtenidos para dicha versión original resultaron ser poco satisfactorios, en contraste con los resultados mencionados por Barcia en sus trabajos.

Este hecho se interpretó como una dificultad intrínseca en la resolución de los problemas de set partitioning (para los que en los distintos trabajos no se reportaban resultados computacionales) y de ahí surgió la idea de reforzar la formulación de la relajación lagrangiana con algún tipo de restricción subrogada, que incorporase información sobre la estructura del problema original. De esta forma, se esperaba, además, recoger la sugerencia de Geoffrion [Geo69], en cuyo trabajo seminal sobre subrogación se incorporaban este tipo de restricciones a los subproblemas para intentar acercarse a la factibilidad, dentro de un esquema de enumeración implícita.

Como consecuencia, y con la esperanza de poder evaluar la influencia que puede tener la "potencia" de distintas restricciones subrogadas, se formularon distintas propuestas, en las que en cada una de ellas se incorporaba una restricción subrogada diferente: una, que era inmediata de obtener y que recogía información obvia sobre la estructura de los problemas, y otra, más costosa de generar, pero que incorporaba información más selecta sobre la estructura de los mismos.

Cabe resaltar que, en general, existe muy poca diferencia en la calidad de las cotas obtenidas por las dos variantes del algoritmo BISA, tanto la que considera como segunda restricción $nx \leq m$ como la que considera la restricción subrogada obtenida por la aplicación del método de Dyer [Dye80]. En ese sentido y, considerando que para la segunda de las versiones hay que añadir el esfuerzo computacional previo para la obtención de dicha restricción subrogada, la primera de las versiones parece más adecuada en un contexto algorítmico.

A pesar de que para cierta cantidad de problemas las dos variantes del algoritmo BISA han proporcionado la solución óptima y para otros problemas se ha mejorado de forma sensible la calidad de la mejor cota inferior, para los problemas "difíciles" y, especialmente, para aquellos de dimensiones mayores, las mejoras obtenidas en las cotas inferiores no han resultado ser lo satisfactorias que se esperaba.

En ese sentido, cabe resaltar, en ambos casos, la aparición de dos tipos de dificultades. Por un lado, la dificultad en la convergencia del algoritmo en sí y en que, de esta convergencia, se deduzca una mejor cota inferior, y, por otro lado, la dificultad incremental que se presenta en la estructura de los distintos problemas de knapsack con dos restricciones, tanto aquellos en los que las dos restricciones son del mismo sentido, como los que tienen las dos restricciones de sentido contrario. Ambas dificultades de deben al hecho de que, a medida de que la aumenta la cota inferior, el conjunto de soluciones posibles para los problemas resultantes resulta mucho más reducido. Por ese motivo, y como ya se ha comentado anteriormente, respecto a los dos tipos de problemas de knapsack que se resuelven, a medida de que la cota inferior aumenta resulta más difícil satisfacer la restricción $cx \geq s_0$ sin violar la segunda de las restricciones.

Respecto a los algoritmos propuestos para resolver los dos tipos de problemas de knapsack con dos restricciones, que han aparecido en las variantes del procedimiento BISA, consideramos que, desde una perspectiva general, su funcionamiento ha sido altamente satisfactorio. En media, ambas versiones resuelven los problemas con un esfuerzo computacional bastante reducido. Las heurísticas propuestas en los dos casos, proporcionan soluciones posibles que, cuando no se trata del óptimo, están muy próximas al mismo y los procedimientos de eliminación utilizados han permitido eliminar gran número de variables.

Por todo lo anterior, la utilización de ambos algoritmos parece muy prometedora, especialmente, si se tiene en cuenta que los dos tipos de problemas de knapsack con dos restricciones que se han estudiado aparecen, también, en distintas formulaciones de relajaciones lagrangianas asociadas a otros tipos de problemas diferentes de los de set partitioning.

En particular, como ya se ha comentado anteriormente, estos tipos de problemas pueden aparecer en distintas relajaciones lagrangianas asociadas a los problemas de localización de plantas con restricciones de capacidad. En concreto, algunas de las relajaciones lagrangianas que pueden formularse y que resultan sugerentes para estos problemas, a partir del "variable splitting" [BaFeJo86], presentan una estructura de problemas de knapsack con dos restricciones. Lo mismo ocurre, en el caso de algunas relajaciones lagrangianas asociadas a problemas de spanning trees minimales sometidos a restricciones complementarias [BaJoMi87].

Por ese motivo, una de las líneas de investigación más sugerentes que se deriva del trabajo realizado consiste en estudiar las posibles mejoras en las heurísticas y de los métodos de eliminación de variables así como la aplicación de los algoritmos obtenidos a relajaciones lagrangianas que se deriven de los problemas que se acaban de mencionar o a otros tipos diferentes de problemas.

En relación con el comportamiento de la clase de algoritmos propuestos para resolver los problemas de set partitioning se plantean diversas cuestiones.

En primer lugar, consideramos que el funcionamiento del algoritmo básico propuesto resulta suficientemente satisfactorio, tanto por la calidad de las cotas superiores obtenidas, como por el número de iteraciones requeridas para su terminación y el esfuerzo computacional necesario. Queda abierto, sin embargo el problema de la identificación del óptimo cuando la calidad de la cota inferior obtenida no es suficientemente buena. En este punto habría que seguir buscando algún tipo de formulación dual utilizada con ese objetivo como complemento del algoritmo básico cuyos requerimientos, en términos de esfuerzo computacional resulten reducidos.

Resulta importante, además, poder utilizar la experiencia obtenida para definir tácticas que permitan sistematizar el diseño de un algoritmo híbrido para la resolución de problemas de set partitioning. Las distintas pruebas realizadas, apuntan claramente a la necesidad de establecer criterios que permitan realizar esta sistematización. En definitiva, habría que establecer algún tipo de criterio de decisión en virtud del cual el algoritmo utilizase uno u otro de los distintos procedimientos posibles de refuerzo dual.

Finalmente, a partir del trabajo realizado, aparecen distintas líneas futuras de investigación, algunas de las cuales están relacionadas con los problemas de set partitioning y otras con otros tipos de problemas.

Con respecto a los problemas de set partitioning, y a pesar del comportamiento satisfactorio del procedimiento de obtención de desigualdades que se ha utilizado, resulta evidente que, si se dispusiese de algún método eficiente para conocer la dimensión del polítopo de set partitioning, y fuese posible obtener facetas para el mismo, ello incidiría muy positivamente en cualquier algoritmo que las utilizase. Teniendo en cuenta que lo anterior parece, actualmente, poco probable, es importante continuar buscando otro tipo de desigualdades válidas. En ese sentido, es interesante profundizar en la relación existente entre los problemas de set partitioning y los de set packing, para poder estudiar la "potencia" de algunas de las facetas de los polítopos de set packing al ser utilizadas como desigualdades válidas para problemas de set partitioning. La identificación, en tiempo polinómico, de ciertos tipos de facetas para los politopos de algunos casos particulares de problemas de set packing [BaYu86], hace esta aproximación especialemente sugerente.

Con respecto a otras familias de problemas, queda por estudiar la sistematización y generalización de toda una clase de heurísticas que pueden diseñarse dentro del marco propuesto en la heurística de Fisher y Kedia [FiKe86], que es en el que se han diseñado las heurísticas propuestas en este trabajo. Pensamos que esta línea de trabajo es fácilmente generalizable y puede proporcionar resultados interesantes en su aplicación a una clase de problemas más general.

Como ya se ha comentado anteriormente en este mismo capítulo, los algoritmos para la resolución de problemas de knapsack con dos restricciones que se han utilizado, proporcionan una herramienta muy valiosa para el estudio de otro tipo de relajaciones lagrangianas que pueden aparecer en la resolución de de problemas diferentes de los de set partitioning. La utilización más inmediata de dichos algoritmos se presenta en el contexto de problemas de localización de plantas con restricciones de capacidad y problemas de spanning trees minimales con restricciones complementarias.

Finalmente, y a pesar del relativo fracaso obtenido con la utilización de las variantes del método BISA para los problemas de set partitioning, pensamos que la idea de reforzar la formulación de la relajación lagrangiana, con alguna restricción subrogada que aporte información sobre la estructura de los problemas originales, no sólo es generalizable, sino que también puede dar lugar a resultados satisfactorios para otros tipos de problemas, especialmente, si consideramos que los algoritmos para resolver problemas de knapsack con dos restricciones que se han utilizado suponen una herramienta más que aceptable en este contexto.

BIBLIOGRAFÍA

[AlTh87] Ali A.I.; Thiagarajan H., A Network Relaxation Based Enumeration Algorithm for Set Partitioning. Working Paper 86/87-3-6. University of Texas at Austin, 1987.

[Bal74] Balas E., Disjunctive Programming: Properties of the Convex Hull of Feasible Points. MSRR 348, Carnegie-Mellon University, Julio 1974.

[Bal75] Balas E., Disjunctive Programming: Cutting Planes from Logical Conditions. O.L. Mangasarian, R.R. Meyer, S.M. Robinson (Eds), *Nonlinear Programming* 2, Academic Press, 1975, pp. 279-312.

[Bal76] Balas E., A Disjunctive Cut for Set Partitioning. W.P.57-75-76, Carnegie Mellon University, Enero 1976.

[Bal77] Balas E., Some Valid Inequalities for the Set Partitioning Problem. *Annals of Discrete Mathematics* **1**, North Holland, 1977, pp. 13-47.

[Bal79] Balas E., Disjunctive Programming. *Annals of Discrete Mathematics* **5**, North Holland, 1979, pp. 3-51.

[Bal80] Balas E., Cutting Planes from Conditional Bounds: A New Approach to Set Covering. *Mathematical Programming* **12**, 1980, pp. 19-36.

[BaCh81] Balas E.; Christofides N., A Restricted Lagrangean Approach to the Travelling Salesman Problem. *Mathematical Programming* **21**, 1981, pp. 19-46.

[BaGePa74] Balas E.; Gerritsen R.; Padberg M.W., An All-Binary Column Generating Algorithm for Set Partitioning. Paper presented at ORSA-TIMS, Boston. Abril 1974.

[BaHo80] Balas E.; Ho A., Set Covering Algorithms Using Cutting Planes Heuristics and Subgradient Optimization: A Computational Study. *Mathematical Programming* **12**, 1980, pp.37-60.

[BaJe75] Balas E.; Jeroslow R.G., Strengthening Cuts for Mixed Integer Programs. MSRR **359**, Carnegie-Mellon University, Febrero 1975.

[BaNg85] Balas E.; Ng S.H., On the Set Covering Polytope: All the facets with coefficients in {0,1,2}. MSRR **522**, 1985.

[BaSa74] Balas E.; Samuelsson H., A Symmetric Subgradient Cutting Plane Method for Set Partioning. WP 5-74-75, Carnegie-Mellon University, Agosto 1974.

[BaPa72] Balas E.; Padberg M.W., On the Set Covering Problem. *Operations Research* **20**, 1972, pp. 1152-1161.

[BaPa73] Balas E.; Padberg M.W., Adjacent Vertices of the Convex Hull of Feasible 0-1 Points. MSRR **298**, Carnegie-Mellon University, Noviembre 1973.

[BaPa79] Balas E.; Padberg M., Set Partitioning. A Survey. *Combinatorial Optimization*, Capítulo 7. N. Christofides (Ed.), John Wiley, 1979, pp. 151-210.

[BaYu86] Balas E.; Yu C.S., Finding a Maximum Clique in an Arbitrary Graph. *SIAM Journal in Computer* **15** (4), Noviembre 1986, pp.1054-1068.

[BaZe76] Balas E.; Zemel E., Graph Substitution and Set Packing Polytopes. MSRR **384**, Carnegie-Mellon University, Enero 1976.

[Barce82] Barceló J., Disyunciones, Planos Secantes y Lagrangianos Restringidos: Tendencias Actuales de la Programación Discreta. *Actas del Seminario sobre Programación Matemática,* J. Barceló y L.F. Escudero (Eds.), Marzo 1982.

[Barce85] Barceló J., Experiencias Computacionales con Procedimientos de Identificación de Restricciones para Algunos Tipos de Programas Enteros. *Qüestió* 9 (2), 1985, pp. 121-146.

[BaFeJö86] Barceló J., Fernández E, Jörnsten K., Computational results from a new lagrangian Relaxation Algorithm for the Capacitated Plant Location Problem. *Proceedings of the Working Conference on Computational Issues in Combinatorial Optimization*, Capri, Italy, Marzo 1986.

[BaJöMi87] Barceló J., Jörnsten K., Migdadas S., The Resource Constrained Spanning Tree Problem: Alternative Modelling and Algorithmic Approaches. *Proceedings of the NATO Advanced Research Workshop*, Junio 1987.

[Barci85a] Barcia P., Constructive Dual Methods for Discrete Programming. Preliminary version. Universidade Nova de Lisboa, Mayo 1985.

[Barci85b] Barcia P., Métodos de Reforço do Dual em Programaçco Inteira. Tesis Doctoral. Instituto Superior Técnico de Engenharia de Sistemas, Lisboa, Diciembre 1985.

[BaGo79] Bazaara M.S., Goode J.J., A Survey of Various Tactics for Generating Lagrangian Multipliers in the Context of Lagrangian Duality. *European Journal of Operational Research* 3, 1979, pp. 322-338.

[BeSh77] Bell D.E.; Shapiro J.F., A Convergent Duality Theory ofr Integer Programming. *Operations Research* 25, no. 3. 1977, pp.419-434.

[BeRa71] Bellmore M.; Ratliff H.D., Set Covering and Involutory Bases. *Management Science* 18 (3), 1971, pp. 194-206.

[BeMi73] Bertsekas D.P.; Mitter S.K., A Descent Numerical Method for Optimization Problems with non Differentiable Cost Functionals. *SIAM J. Control* 11 (4), Noviembre 1973, pp. 637-65.

[CaFrMa75] Camerini P.M.; Fratta L.; Maffioli F., On Improving Relaxation Methods by Modified Gradient Techniques. *Mathematical Programming Study* 3, North-Holland, 1975, pp. 26-34.

[Cam83a] Campello R.E., Accelerates Euclidean Algorithm's Behavior on Solving High Density Equality-Constrained Set Covering Problems. WP 62-82-83, Carnegie-Mellon University, 1983.

[Cam83b] Campello R.E., A Prototype Algorithm Combining Branch and Bound and Cuts for the Set Partitioning Problems. WP 74-82-83, Carnegie-Mellon University, 1983.

[Cam83c] Campello R.E., A Hybrid Primal Cutting Plane/Implicit Enumeration Algorithm to the Equality Constrained Set Covering Problem. DP 05-83, Carnegie-Mellon University, 1983.

[CaMa83] Campello R.E. and Maculan N., On Deep Disjunctive Cutting Planes for Set Partitioning: A Computational Study. *Proc. of the Int. Cong. Mathematical Programming*, R. Cottle, B. Korte and M.Kemalson (Eds.). North Holland, 1983.

[Chv72] Chvàtal V., On Certain Polytopes Associated with Graphs. CRM-238, Université de Montreal, Octubre 1972.

[Chv79] Chvàtal V., A Greedy Heuristic for the Set Covering Problem. *Mathematics of Operations Research* 4 (3), 1979, pp. 233-235.

[CrJoPa83] Crowder H. Johnson E.L., Padberg M.W., Solving Large-Scale Zero-One Linear Programming Problems. *Operations Research* **31**, 1983, pp. 803-834.

[Dan59] Dantzig G.B., Notes on Solving Linear Programs in Integers, *Nav. Res. Log. Quart.* **6**, 1959, pp. 75-76.

[Del74] Délorme J., Contribution à la résolution du problème de recouvrement: méthodes de troncatures. Thèse de Docteur Ingénieur. Université de Paris VI, 1974.

[Dye80] Dyer M.E., Calculating Surrogate Constraints. *Mathematical Programming* **19**, North Holland, 1980, pp. 255-278.

[Edm65] Edmonds J., Maximum Matching and a Polyhedrom with 0,1 Vertices. *Journal of Research of the National Bureau of Standards* **69b**, 1965, pp. 125-130.

[ElEl85] Elinam A.A.; Elmaghraby S.E., On the Reduction Method for Integer Linear Programs, II. *Discrete Applied Mathematics* **12**, North Holland, 1985, pp. 241-260.

[Ema86] Emalghraby S.E., The Knapsack Problem with Generalized Upper Bounds. OR Report no. 209. North Carolina State University at Raleigh, 1986.

[EmWi70] Emalghraby S.E; Wig M.K., On the Treatmente of Stock Cutting Problems as Diophantine Programs. North Carolina State University and Corning Glass Research Center Report no. 61. Mayo 1970.

[Fis81] Fisher M.L., "The Lagrangean Relaxation Method for Solving Integer Programming Problems". *Management Science* **27** (1), Enero, 1981, pp. 1-18.

[FiKe86] Fisher M.L.; Kedia P., A Dual Algorithm for Large Scale Set Partitioning. KGSMP **894**, Purdue University, Mayo 1986.

[FrPl86] Freville A.; Plateau G., Heuristics and Reduction Methods for Multiple Constraint 0-1 Linear Programming Problems. *European Journal of Operational Research* **24**, North Holland, 1986, pp. 206-215.

[Gal58] Gallai T., Über Extreme Punkt-und Kantenmengen. *Ann. Univ. Sci. Budapest, Eötvös, Sect. Math.* **2**. 1958, pp.133-138.

[GaNe69] Garfinkel R.S.; Nemhauser G.L., The Set Partitioning Problem: Set Covering with Equality Constraints. *Operations Research* **17**, 1969, pp. 848-856.

[GaNe72] Garfinkel R.S.; Nemhauser G.L., *Integer Programming*, John Wiley, 1972.

[GaPi85] Gavish B.; Pirkul H., Efficient Algorithms for Solving Multiconstraint zero-one Knapsack Problems to Optimality. *Mathematical Programming* **31**, North Holland, 1985, pp. 78-105.

[GiGo61] Gilmore P.C; Gomory R.E., A Linear Programming Approach to the Cutting Stock Problem, Part I. *Operations Research* **9**. 1961, pp. 849-859.

[Geo69] Geoffrion A.M., An Improved Implicit Enumeration Approach for Integer Programming. *Operations Research* **17**. 1969, pp. 437-454.

[Geo74] Geoffrion A.M., Lagrangean Relaxation for Integer Programming. *Mathematical Programming Study* **2**, North-Holland, 1974, pp. 82-114.

[Glo75] Glover F., Surrogate Constraint Duality in Mathematical Programming. *Operations Research* **23** (3), 1975, pp. 434-45.

[Gof77] Goffin J.L., On Convergence Rates of Subgradient Optimization Methods. *Mathematical Programming* **13**, North-Holland, 1977, pp. 329-347.

[Gom58] Gomory R.E., Outline of an Algorithm for Integer Solutions to Linear Programs. *Bull. Am. Math. Soc.* **64**, 1958, pp. 275-278.

[GrPi70] Greenberg H.J., Pierskalla W.P., Surrogate Mathematical Programming. *Operations Research* **18**, 1970, pp. 924-939.

[Grö81] Grötschel M., Aproaches to Hard Combinatorial Optimization Problems. Report no. 79147-OR. Institut für Vkonometrie und Operations Research, Rheinische Friedrich - Wilhems - Universität. Bonn, Marzo 1981.

[HaHo85] Hall N.G.; Hochbaum D.S., The Multicovering Problem: The Use of Heuristics, cutting Planes and Subgradient Optimization for a Class of Integer Programs. WPS85-73. Administrative Science College. Ohio State University, 1985.

[HeKa71] Held M.; Karp R.M., The Travelling Salesman Porblem and Minimum Spaning Trees: Part II. *Mathematical Programming* **1**, 1971, pp. 6-25.

[HeWoCr74] Held M.; Wolfe P.; Crowder H., Validation of Subgradient Optimization. *Mathematical Programming* 6, North-Holland , 1974, pp. 62-88.

[KaRaBa87] Karwan M.H.; Ram B.; Babu A.J.G., Agregation of Constraints in Integer Programming. Manuscrito sin publicar, 1987.

[KeZi72] Kendall K.; Zionts S., Solving Integer Programming Problems by Aggregating Constraints. Paper presented to the Joint Meeting of The Operations Research Society of America, The Institute of Management Sciences, American Institute of Industrial Engeneers, Atlantic City, New Jersey. Noviembre 1972.

[Mac83] Maculan N., Programation Linéaire en Nombres Entiers. Publication no. 486. Départment d'Informatique et de Recherche Opérationelle, Université de Montréal, Agosto 1983.

[Mar74] Marsten R.E., An Algorithm for Large Set Partitioning Problems. *Management Science* 20, 1974, pp. 779-787.

[Mar75] Marsten R.E., The Use of Boxstep Method in Discrete Optimization. *Operations Research* 23 (3), Mayo-Junio, 1975, pp. 145-173.

[MaSh81] Marsten R.E.; Shepardson F., Exact Solution of Crew Scheduling Problems Using the Set Partitioning Model: Recent Successful Aplications. *Networks* **11**, 1981, pp. 165-177.

[MaHoBa75] Marsten R.E.; Hogan W.W.; Blankenship J.W., The Boxstep Method for Large Scale Optimization. *Operations Research* 23 (3), Mayo-Junio, 1975, pp. 389-405.

[MaSh81] Marsten R.; Shepardson F., Exact Solution of Crew Scheduling Problems Using the Set Partitioning Model: Recent Successful Aplications. *Networks* **11**, John Wiley, 1981, pp. 165-167.

[MaTo79] Martello S.; Toth P. The 0-1 Knapsack Problem. *Combinatorial Optimization,* Capítulo 9. N. Christofides (Ed.), John Wiley, 1979, pp. 237-279.

[NeTr74] Nemhauser G.L.; Trotter L.E., Properties of Vertex Packing and Independence System Polyhedra. *Mathematical Programming* **6**, 1974, pp. 48-61.

[NeWe79] Nemhauser G.L.; Weber G.M., Optimal Set Partitioning, Matchings and Lagrangian Duality. *Naval Research Logistic Quarterly* **26**(4), 1979, pp. 553-563.

[Pad 73] Padberg M.W., On the Facial Structure of Set Packing Polyhedra. *Mathematical Programming* **5**, 1973, pp. 199-215.

[Pad74] Padberg M.W., Perfect Zero-One Matrices. *Mathematical Programming* **6**, 1974, pp. 180-196.

[Pad75] Padberg M.W., On the Complexity of Set Packing Polyhedra. Working Paper no. 75-105, GBA, New York University, 1975.

[Pad79] Padberg M.W., Covering, Packing and Knapsack Problems. *Annals of Discrete Mathematics* **4**, 1979, pp. 265-287.

[Pol67] Polyak B.T., A General Method for Solving Extremum Problems. *Dokladi Akademii Nauk SSSR* **74** (1), 1967, pp. 33-36.

[Pol69] Polyak B.T., Minimization of Unsmooth Functionals. *Zhurnal Vychislitel'noi Matematiki i Matematischeskoi Fiziki* **9**, 1969, pp. 509-521.

[PuEd73] Pulleyblank W.; Edmonds J., Facets of 1-Matching Polyhedra. CORR 73-3, University of Waterloo, Marzo 1973.

[Sal75] Salkin H.M., *Integer Programming*, Addison-Wesley, Reading, Mass. 1975.

[SaKo73] Salkin H.M.; Konkall R.D., Set Covering by an All Integer Algorithm: Computational Experience. *ACM Journal* **20**, 1973, pp. 189-193.

[Sam74] Samuelsson H., A Symetric Ascent Method for Finitely Regularizable Linear Programs. WP 6-74-75, Carnegie-Mellon Unuversity, Agosto 1974.

[Sas85] Sassano A., On the Facial Structure of the Set Covering Polytope. Consiglio Nazionale delle Ricerche, Istituto di Analisi ed Informatica. Diciembre 1985.

[Sha79] Shapiro J.F., A Survey of Lagrangean Techniques for Discrete Optimization. *Annals of Discrete Mathematics* 5, North Holland, 1979, pp. 113-138.

[Shi79] Shih W., A Branch and Bound Method for the Multi Constraint Zero-One Knapsack Problem. *Journal of Operational Research Association Society* **30**, Pergamon Press Ltd., 1979, pp. 369-378.

[Tro74] Trotter L.E., A Class of Facet Producing Graphs for Vertex Packing Polyhedra. Technical Report no. 78, Yale University, Febrero 1974.

[Wey35] Weyl H., Elementare Theorie der Konvexen Polyeder. *Comm. Math. Helv.* 7. 1935, pp.290-306.

[YeKoKr84] Yemelichev V.A.; Kovalev M.M.; Kravtsov M.K., Polytopes, Graphs and Optimisation. Cambridge University Press, Cambridge 1984.

[Zio74] Zionts S., *Linear and Integer Programming*. Prentice-Hall, New Jersey 1974.